IJCC Editorial Board

Editors-in-Chief
Hemant Jain, University of Wisconsin–Milwaukee, USA
Rong N. Chang, IBM Research, USA & China

Associate Editor-in-Chief
Qingyang Wang, Louisiana State University, USA

Editorial Board
Danilo Ardagna, Politecnico di Milano, Italy
Janaka Balasooriya, Arizona State University, USA
Roger Barga, Microsoft Research, USA
Viraj Bhat, Yahoo, USA
Rajdeep Bhowmik, Cisco Systems, Inc., USA
Jiannong Cao, Hong Kong Polytechnic University, Hong Kong
Buqiang Cao, Hunan University of Science and Technology, China
Keke Chen, Wright State University, USA
Haopeng Chen, Shanghai Jiao Tong University, China
Malolan Chetlur, IBM India, India
Alfredo Cuzzocrea, ICAR-CNR & University of Calabria, Italy
Ernesto Damiani, University of Milan, Italy
De Palma, University Joseph Fourier, France
Claude Godart, Nancy University and INRIA, France
Nils Gruschka, University of Applied Sciences, Germany
Paul Hofmann, Saffron Technology, USA
Ching-Hsien Hsu, Chung Hua University, Taiwan
Patrick Hung, University of Ontario Institute of Technology, Canada
Hai Jin, HUST, China
Li Kuang, Central South University, China
Bing Li, Wuhan University, China
Grace Lin, Institute for Information Industry, Taiwan
Xumin Liu, Rochester Institute of Technology, USA
Shiyong Lu, Wayne State University, USA
J.P. Martin-Flatin, EPFL, Switzerland
Vijay Naik, IBM T.J. Watson Research Center, USA
Surya Nepal, Commonwealth Scientific and Industrial Research Organisation, Australia
Norbert Ritter, University of Hamburg, Germany
Josef Schiefer, Vienna University of Technology, Austria
Jun Shen, University of Wollongong, Australia
Weidong Shi, University of Houston, USA
Liuba Shrira, Brandeis University, USA
Kwang Mong Sim, University of Kent, UK
Wei Tan, IBM T.J. Watson Research Center, USA
Kunal Verma, Accenture Technology Labs, USA
Raymond Wong, University of New South Wales & NICTA, Australia
Qi Yu, Rochester Institute of Technology, USA
Jia Zhang, Carnegie Mellon University – Silicon Valley, USA
Gong Zhang, Oracle Corporation, USA

http://hipore.com/ijcc
Call for Articles
International Journal of Cloud Services

Mission
Cloud Computing has become the de facto computing paradigm for Internet-scale service development, delivery, brokerage, and consumption in the era of Services Computing, fueling innovative business transformation and connected human society. More than 25 billion smart devices would be communicating dynamically over inter-connected clouds by 2020 as integral components of various industrial service ecosystems. The technical foundations of this trend include Service-Oriented Architecture (SOA), business & IT process automation, software-defined computing resources, elastic programming model & framework, and big data management and analytics. In terms of the delivered service capabilities, a cloud service could be, among other as-a-service types, an infrastructure service (managing compute, storage, and network resources), a platform service (provisioning generic or industry-specific programming API & runtime), a software application service (offering email-like ready-to-use application capabilities), a business process service (providing a managed process for, e.g., card payment), a mobile backend service (facilitating the integration between mobile apps and backend cloud storage and capabilities) or an Internet-of-things service (connecting smart machines with enablement capabilities for industrial clouds).

As the first Open Access research journal on Cloud Computing, the International Journal of Cloud Services (IJCS), formally International Journal of Cloud Computing (IJCC), aims to be a valuable resource providing leading technologies, development, ideas, and trends to an international readership of researchers and engineers in the field of Cloud Computing.

Topics
Topics of interest include, but are not limited to, the following:

- ROI Model for Infrastructure, Platform, Application, Business, Social, Mobile, and IoT Clouds
- Cloud Computing Architectures and Cloud Solution Design Patterns
- Self-service Cloud Portal, Business Dashboard, and Operations Management Dashboard
- Autonomic Process and Workflow Management in Clouds
- Cloud Service Registration, Composition, Federation, Bridging, and Bursting
- Cloud Orchestration, Scheduling, Autoprovisioning, and Autoscaling
- Cloud Enablement in Storage, Data, Messaging, Streaming, Search, Analytics, and Visualization
- Software-Defined Resource Virtualization, Composition, and Management for Cloud
- Security, Privacy, Compliance, SLA, and Risk Management for Public, Private, and Hybrid Clouds
- Cloud Quality Monitoring, Service Level Management, and Business Service Management
- Cloud Reliability, Availability, Serviceability, Performance, and Disaster Recovery Management
- Cloud Asset, Configuration, Software Patch, License, and Capacity Management
- Cloud DevOps, Image Lifecycle Management, and Migration
- Cloud Solution Benchmarking, Modeling, and Analytics
- High Performance Computing and Scientific Computing in Cloud
- IoT & Mobile Cloud Services, Edge & Fog Computing Services, Machine-to-Machine Clouds
- Cloud Programming Model, Paradigm, and Framework
- Cloud Metering, Rating, and Accounting
- Innovative Cloud Applications and Experiences
- Green Cloud Computing and Cloud Data Center Modularization
- Economic Model and Business Consulting for Cloud Computing
- Smart & Cognitive Cyberphysical Cloud Services

http://hipore.com/ijcc
Table of Contents

EDITOR-IN-CHIEF PREFACE
iv
Hemant Jain, University of Wisconsin–Milwaukee, USA
Rong N. Chang, IBM Research, USA & China

RESEARCH ARTICLES
1 Research Gaps and Trends in Cloud Computing: A Systematic Mapping Study
Liping Zhao, University of Manchester, UK
Liang-Jie Zhang, Kingdee International Software Group Company, China
Tina X. O. Liu, Credit Suisse, UK

12 Boosting Metrics: Measuring Cloud Services from the Holistic Perspective
Zheng Li, Australian National University and NICTA, Australia
Liam O’Brien, Geoscience Australia, Australia
He Zhang, Nanjing University, China

23 A SLA Violation Degree-Aware Cloud Service Evaluation Approach Based on Historical Records
Lianyong Qi, Nanjing University and Qufu Normal University, China
Wanchun Dou, Nanjing University, China
Xuyun Zhang, NICTA, Australia
Yuming Zhou, Nanjing University, China

34 Incident Notification Process as BPaaS for Electricity Supply Systems
Lai Xu, Bournemouth University, UK
Paul de Vrieze, Bournemouth University, UK
Nan Jiang, Bournemouth University, UK

45 Call for Papers: IEEE CLOUD/ICWS/SCC/MS/BigData/SERVICES 2015
Call for Articles: International Journal of Cloud Computing (IJCS)
Call for Articles: International Journal of Services Computing (IJSC)
Call for Articles: International Journal of Big Data (IJBD)
Editor-in-Chief Preface:
Cloud Service Evaluation and Quality Management

Hemant Jain
University of Wisconsin–Milwaukee, USA

Rong N. Chang
IBM Research, USA & China

Welcome to the fourth issue of 2014 of the International Journal of Cloud Computing (IJCC), the first open access on-line journal on cloud computing. As cloud computing becomes more and more popular the importance of the economics of cloud computing is becoming increasingly important. For the sustained development of cloud computing the provider needs to optimize their services and manage the environment in a way to make it economically sustainable. The cloud computing combined with developments like internet of things are significantly changing the life as we know it today. However, to deliver on these promises and to prevent cloud computing from becoming a passing fad significant economic, and business issues need to be addressed. IJCC is designed to be an important platform for disseminating high quality research on above issues in a timely manner and provide an ongoing platform for continuous discussion on research published in this journal. We aim to publish high quality research that addresses important technical challenges, economics of sustaining this environment, and business issues related to use of this technology including privacy and security concerns, legal protection, etc. We seek to publish original research articles, expanded version of papers presented at high quality conferences, key survey articles that summarizes the research done so far and identify important research issues, and some visionary articles from invited experts. We will make every effort to publish articles in a timely manner.

This issue contains one invited survey article on cloud research gaps and trends, and three extended articles in the areas of cloud service evaluation and quality management.

The first article is titled “Research Gaps and Trends in Cloud Computing: A Systematic Mapping Study”. This article undertakes a Mapping Study of 523 papers published in the proceedings of the IEEE CLOUD 2009 to CLOUD 2014 with the goal of providing a structure to them in terms of their research topic, contribution, and research type. Moreover, it uses thematic and frequency analysis to identify research gaps and trends. It concludes, in those six conferences, technology oriented papers extensively outnumbered business oriented papers; methods significantly outstripped other types of contribution whereas there is a desperate shortage of contribution from practical experience; and empirical research papers dominated the conferences whereas theoretical research was close to none.

The second article is titled “Boosting Metrics: Measuring Cloud Services from the Holistic Perspective”. This article addresses the difficulty in comparing two cloud services via the common cloud service evaluation practices, which use a set of benchmark-based separated, diverse, and possibly conflicting measurement criteria. Inspired by the boosting approaches to machine learning, it proposes an easy-to-consume comparative quality evaluation approach for cloud services based upon a new concept named Boosting Metrics, which facilitate measuring one complex cloud service feature involving multiple service properties such that simple conclusions can be drawn from the evaluation results. The proposed approach supplements the strategy of employing benchmark suites and holistically evaluates cloud services with concern of different workloads.

The third article is titled “A SLA Violation Degree-Aware Cloud Service Evaluation Approach Based on Historical Records”. This article addresses the issues of using a fixed quality value (or quality point) to represent a cloud service’s quality, though that value can be determined via historical quality measures. It proposes a cloud service quality evaluation approach which takes into account several practical concerns: long-term running characteristics of the service, unpredictability of the underlying network environment, and degree of SLA violation.
The fourth article is titled “Incident Notification Process as BPaaS for Electricity Supply Systems”. This article proposes a cloud-based scalable and flexible business process management (BPM) service development approach to supporting agile inter-enterprise business collaborations. The proposed approach leverages existing distributed local BPM systems under an extended PAD (process engine, activity, and data) model in which business process models can also be distributed and adopted for different purposes. The approach was successfully used for implementing a cross-organization incident notification process for Spanish electricity supply systems.

We would like to thank the authors for their effort in delivering these four quality articles and hope you enjoy reading them.

About the Editors-in-Chief

Dr. Hemant Jain is Roger L. Fitzsimonds Distinguished Scholar and Professor of Information Technology Management at University of Wisconsin–Milwaukee. Dr. Jain specializes in information system agility through web services, service oriented architecture and component based development. His current interests include development of systems to support real time enterprises which have situational awareness, can quickly sense-and-respond to opportunities and threats, and can track-and-trace important items. He is also working on issues related to providing quick access to relevant knowledge for cancer treatment and to providing medical services through a virtual world. Dr. Jain is an expert in architecture design, database management and data warehousing. He teaches courses in database management, IT infrastructure design and management, and process management using SAP. Dr. Jain was the Associate Editor-in-Chief of IEEE Transactions on Services Computing and is Associate Editor of Journal of AIS, the flagship journal of the Association of Information Systems.

Dr. Rong N. Chang is Member of IBM Academy of Technology at IBM Research. He received his Ph.D. degree in computer science & engineering from the University of Michigan at Ann Arbor in 1990 and his B.S. degree in computer engineering with honors from the National Chiao Tung University in Taiwan in 1982. Before joining IBM in 1993, he was with Bellcore researching on B-ISDN realization. He is a holder of the ITIL Foundation Certificate in IT Services Management. His accomplishments at IBM include five corporate-level Outstanding Technical Achievement Awards and seven division-level accomplishments. He is Associate Editor-in-Chief of the IEEE Transactions on Services Computing. He has chaired many conferences in Internet-enabled distributed services and applications. He is Chair of the 2015 IEEE World Congress on Services, including Cloud Computing, Web Services, Services Computing, Mobile Services. He is an ACM Distinguished Member.
RESEARCH GAPS AND TRENDS IN CLOUD COMPUTING: A SYSTEMATIC MAPPING STUDY

Liping Zhao¹, Liang-Jie Zhang², Tina X. O. Liu³
¹School of Computer Science, University of Manchester
liping.zhao@manchester.ac.uk
²Kingdee International Software Group Company Limited
zhanglj@kingdee.com
³Credit Suisse, London
tina.o.liu@gmail.com

Abstract
Context: Since 2009 a large number of papers have been presented at IEEE International Conference on Cloud Computing (CLOUD). Studying these papers and identifying relevant research areas from them are both difficult and time consuming. Objective: To address this problem, we aim to provide a structure to the entire collection of full-length papers published in the proceedings of the past CLOUD conferences (from CLOUD 2009 to CLOUD 2014). Method: We undertake a Systematic Mapping study of these papers by categorizing them. Results: 523 relevant papers are selected and assessed in terms of their research topic, contribution and research type. Thematic and frequency analysis is then conducted to identify research gaps and trends. Conclusions: By research topic, technology oriented papers extensively outnumbered business oriented papers. By contribution, methods significantly outstripped other types of contribution whereas there is a desperate shortage of contribution from practical experience. By research type, empirical research papers have dominated the past six CLOUD conferences whereas theoretical research is close to none.

Keywords: Systematic Mapping, Systematic Review, Cloud Computing Research, CLOUD Conferences

1. INTRODUCTION

Emerged around 2006, cloud computing has been quickly embraced by businesses and governments as a new way to deliver IT services over the Internet. Through cloud services, people can access applications from anywhere in the world on demand (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009). For businesses and organizations, cloud computing offers immediate visible economic benefits, such as cost saving for IT infrastructure and operation, flexible service delivery and reduced carbon emission. Cloud computing is particularly attractive to small and medium-sized businesses, as it enables them to benefit from considerable computing power without purchasing expensive hardware. Since 2011 and for the fifth consecutive year, cloud computing has been constantly identified as one of the top 10 strategic technology trends by Gartner¹.

According to a literature review (Yang & Tate, 2012), first academic papers on cloud computing were published in 2008. Early that year, IEEE Transactions on Services Computing (TSC) added the term “cloud computing” to its body of knowledge areas (Zhang, 2008) and thus officially launched cloud computing as a research discipline within the services computing community. In the same year, IEEE International Conference on Services Computing (SCC 2008) delivered a keynote panel entitled “Business Cloud: Bridging The Power of SOA and Cloud Computing” and a keynote address called “Cloud Computing”. In September 2008, IEEE International Conference on Web Services (ICWS 2008) delivered a keynote on “Web Services: Software-as-a-Service (SaaS), Communication, and Beyond” and a panel on “Cloud Computing and IT as a Service: Opportunities and Challenges”. These events inspired the researchers and practitioners in the areas of service computing and web services to help define and shape the emerging field of cloud computing.

Driven by the industrial force, the first IEEE International Conference on Cloud Computing (CLOUD) was held in 2009. Since then CLOUD conference has become a fixture at annual IEEE World Congress on Services and established itself as the flagship conference in the field. Year by year, the conference has grown from strength to strength and the number of papers presented at this conference has also been increased, from tens to more than one hundred each year. At the last count, there were a total of 636 papers published from CLOUD 2009 to CLOUD 2014.

With the growing number of papers, it becomes important to review and organize these papers, so that they can be easily accessed and digested. Papers that review and organize other papers are called “literature reviews” or “secondary studies”, whereas papers that directly report research results are called “primary studies”. Most papers in

¹ http://www.gartner.com/technology/home.jsp

doi: 10.29268/stcc.2014.2.4.1
the cloud computing field have been primary studies; however, with the maturity of the field, it is imperative to produce more good quality secondary studies. Such studies are essential to any discipline, as they aid in “analyzing the past to prepare for the future (Webster & Watson, 2002)”.

More specifically, they help uncover weak areas where more research is needed, identify saturated areas where researchers should avoid and discover emerging trends for future research directions.

In this paper, we present a secondary study of the papers published at the past six CLOUD conferences (from CLOUD 2009 to CLOUD 2014). Since there are 636 papers, we have chosen to use a specific review methodology called “Systematic Mapping” for our study. Section 2 introduces this methodology, while Sections 3 and 4 present our study using this methodology. Section 5 summarizes the key findings from this study and Section 6 concludes the paper.

2. SYSTEMATIC MAPPING

Systematic Mapping is an evidence-based review methodology for secondary studies (i.e. literature reviews). This methodology uses categorization and thematic analysis to structure and summarize primary studies. It is particularly suitable for providing high-level overviews for large collections of primary studies. This methodology has come from medical research, where it has been used to establish medical evidence in clinic trails, experiments or diseases. It was first introduced to the software engineering field in 2007 (Kitchenham & Charters, 2007), to help identify evidence “clusters” and “deserts” (i.e., gaps) in software engineering research papers (Bailey et al., 2007; Petersen, Feldt, Mujtaba, & Mattsson, 2008).

Systematic Review is another evidence-based methodology frequently used in medical research reviews (Kitchenham & Charters, 2007). Both Systematic Mapping and Systematic Review are complementary in that they focus on different levels of granularity, with the former surveying the literature landscape from above and in breadth, the latter scrutinizing a specific area in depth. This complementarity makes the methodologies good companions for literature review, where Systematic Mapping can be used to identify areas of interest in the literature landscape, whereas Systematic Review can then focus on studying a specific area (Kitchenham & Charters, 2007). This also implies that Systematic Review is not suitable for surveying larger areas, as its attention to detail would require too big an effort to review a large quantity of studies.

The both methodologies apply a systematic process to their studies, hence the name of these methods. The systematic process of SM studies consists of four essential steps:

1. Defining research questions.
2. Identifying relevant primary studies.
3. Categorizing relevant primary studies.
4. Analyzing and synthesizing results obtained and answering research questions.

This process is supported by several techniques, including: a search strategy to guide literature search, inclusion and exclusion criteria for determining and selecting the relevant primary studies, and a classification method for categorizing the relevant studies.

Secondary studies based on Systematic Mapping are also called “systematic mapping studies” (or “mapping studies” for short), whereas those based on Systematic Review are called “systematic reviews”.

Systematic Mapping and Systematic Review are still new to the cloud computing community and only a small number of secondary studies are currently available that have used these methods (Carvalho, Neto, Garcia, Assad, & Durao, 2013; daSilva et al., 2013; daSilva, Neto, Garcia, Assad, & Trinta, 2012; González-Martínez, Bote-Lorenzo, Gómez-Sánchez, & Cano-Parra, 2015; Neto & Duarte, 2013; Pallis, 2010; Rimal, Choi, & Lumb, 2009; Yang & Tate, 2012). Given the rising importance of cloud computing and the fast growing number of publications in the field, it is the high time for cloud computing researchers to conduct more such secondary studies to help analyze and synthesize the existing literature, and to produce high quality literature reviews.

According to the aforementioned systematic process, we present our mapping study in Sections 3 and 4.

3. A MAPPING STUDY FOR CLOUD COMPUTING

3.1 Research Questions

The main goal of a mapping study is to provide an overview of a research area. Such an overview should identify the quantity and type of publications, show the frequencies and trends of publications over time, and reveal research deserts and clusters. This goal should be reflected in the study’s research questions, to guide the review process. Consequently, as noted in (Petersen et al., 2008), research questions for mapping studies are similar, as they are driven by similar goals. Our research questions are similar to research questions used in two mapping studies (Bailey et al., 2007; Petersen et al., 2008):

RQ1: What topics have been most investigated in the papers published in the past six CLOUD proceedings and how have these changed over time?

RQ2: What types of contribution are most popular in these papers and how have these changed over time?

RQ3: What types of research are dominating these publications and how have these changed over time?
These questions have been used to guide and shape our mapping study and discussion.

3.2 Identifying Relevant Primary Studies

Our search strategy is to identify relevant primary studies from the six proceedings of the CLOUD conferences, from CLOUD 2009 to CLOUD 2014. We have searched these proceedings one by one within IEEE Xplore® Digital Library and obtained 636 papers. To determine and select the relevant studies from these results, we have applied the following inclusion and exclusion criteria:

- **Inclusion Criteria:** All the full-length papers which report primary studies should be included.
- **Exclusion Criteria:** (1) All the “Work in Process” papers should be excluded, for the obvious reason. (2) All the secondary studies should be excluded.

After evaluating each of the 636 papers against these criteria, 523 papers have been selected as relevant and included in our study. Table 1 lists the search results and selected papers for the six CLOUD proceedings. We have downloaded all 523 papers and their bibliographic details into an EndNote library, from which our mapping study was conducted.

<table>
<thead>
<tr>
<th>Proceedings</th>
<th>All Papers</th>
<th>Selected Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOUD 2009</td>
<td>23</td>
<td>17</td>
</tr>
<tr>
<td>CLOUD 2010</td>
<td>77</td>
<td>65</td>
</tr>
<tr>
<td>CLOUD 2011</td>
<td>109</td>
<td>91</td>
</tr>
<tr>
<td>CLOUD 2012</td>
<td>135</td>
<td>116</td>
</tr>
<tr>
<td>CLOUD 2013</td>
<td>148</td>
<td>119</td>
</tr>
<tr>
<td>CLOUD 2014</td>
<td>143</td>
<td>115</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>636</strong></td>
<td><strong>523</strong></td>
</tr>
</tbody>
</table>

Table 1. Search Results (N=636) and Relevant Papers (N=523) for CLOUD Proceedings

3.3 A Classification Schema for Categorization

Categorization is essentially a process of placing things in categories whose members bear some similarity to each other. Categorization can be performed in two ways: top-down and bottom-up. In the top-down approach, a set of categories are defined in a classification schema, so categorization is essentially a process of placing things in appropriate categories under this schema. In the bottom-up approach, categories are unknown to start with and only created during the process of categorization. The bottom-up approach usually uses the “Affinity Diagram” technique (Brassard, 1989) to create categories. The process of affinity works like this: It looks for two things that seem to be related and places them into a new group; it then adds more related things to the group. When no more related things could be found, the process moves to create a new group. After all things have been allocated to groups, groups are given suitable names and become categories.

In our view, the bottom-up categorization approach suffers from a number of problems. First, it is random, as forming similarity groups is subjective and arbitrary. Second, there is no constraint on the number of the categories, so we may end up with a large number of small categories, which cannot reveal anything meaningful. Third, the categories formed are individuals, rather than a whole, which defeat the purpose of SM studies where thematic analysis of related studies is the key to achieve their goal.

The top-down categorization approach is not perfect either, as it tends to create a classification schema that is too general, as its categories are formed a priori, before papers have been studied.

In our study, we have applied both the top-down and bottom-up approaches to our categorization. We have defined a classification schema as a super-structure to constrain the bottom-up categorization, to allow low-level categories to emerge freely under this structure. These low-level categories are interrelated via this structure and are meaningful both individually and as a whole. In doing so, our hybrid categorization has benefited from the best of both approaches. In what follows, we describe a faceted classification schema.

Our classification schema consists of two levels. The top level contains three facets: **Topic**, **Contribution** and **Research Type**. These three facets correspond to our three research questions, suggesting that our categorization is guided by our research questions.

In the second level, each facet is divided into a number of **categories**. This allows us to identify clusters and deserts in a meaningful way.

To determine the categories for the **Topic Facet**, we have studied the following publications: the NIST cloud computing standards roadmap (Hogan, Liu, Sokol, & Tong, 2011), the cloud computing open architecture (Zhang & Zhou, 2009), architectural requirements for cloud computing systems (Rimal, Jukan, Katsaros, & Goel, 2011), cloud computing taxonomies (Armbrust et al., 2010; Carvalho et al., 2013; daSilva et al., 2013; daSilva et al., 2012; Neto & Duarte, 2013; Pallis, 2010; Rimal et al., 2009; Yang & Tate, 2012), and requirements classification in software engineering (Sommerville, 2004). From these publications, we have identified six broad cloud-computing topics, concerning **Functional Aspect**, **Non-Functional Aspect**, **Architectural Aspect**, **Resource Aspect**, **Adoption Aspect**, and **Domain Aspect**.

We believe these aspects are high-level, general and comprehensive enough to cover major cloud computing topics; they are also easily separable, because they represent different aspects about the development of cloud computing.
3.4 Categorizing Relevant Primary Studies

This step involves sorting the relevant primary studies (i.e., papers) into different categories within the classification schema. The categories within each facet are mutually exclusive and non-overlapping. This means the papers can only be classified into one and only one category within each facet. This requirement ensures that the total number of the relevant papers in each facet maintains the constant (N = 523).

To determine each paper’s category, we have employed the “keywording” technique (Petersen et al., 2008) to profile the paper. This involved reading each paper’s title, keywords, abstract, introduction, and conclusion to establish which category the paper should be placed. When we could not decide, we briefly read other sections of the paper or discussed with each other. We carried out this step iteratively to ensure that our classification is as accurate as possible.

This step is also called “data extraction” or “mapping”. The results of this step are presented as “a systematic map”, which is a visual representation of the distributions and statistics of different categories and their relationships. Such a map serves as an overview of the mapping study. Each SM study normally produces multiple systematic maps, showing different views of the mapping results.

Our mapping results are presented in the following ways: The categories and citations of the 523 papers are presented in six Word documents; each document contains the categories and citations of the papers for each proceedings. We will make these documents available at the journal website. The statistics and analysis of the mapping results are presented in multiple, complementary visual representations.
<table>
<thead>
<tr>
<th>Facet</th>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topic</strong></td>
<td>Functional Aspect</td>
<td>This topic deals with the functional aspect of Cloud Computing applications development, such as cloud services (IaaS, PaaS, SaaS, XaaS), cloud deployment (public, private, hybrid cloud), measurements, and application &amp; tool support.</td>
</tr>
<tr>
<td></td>
<td>Non-Functional Aspect</td>
<td>This topic addresses the non-functional or quality aspect of Cloud Computing systems development, including trust, security, privacy, data integrity &amp; protection, performance, reliability, availability, scalability, elasticity, fault-tolerance, and usability.</td>
</tr>
<tr>
<td></td>
<td>Architectural Aspect</td>
<td>This topic covers the architectural aspect of cloud systems, including cloud architectures &amp; design, programming language, virtual machine, virtualization, middleware, cloud compliancy &amp; standards, and cloud maturity model.</td>
</tr>
<tr>
<td></td>
<td>Resource Aspect</td>
<td>This topic concerns the resource aspect of cloud systems development, including managing, controlling and using cloud computing resources, such as resource utilization &amp; management, energy efficiency, job &amp; resource scheduling.</td>
</tr>
<tr>
<td></td>
<td>Adoption Aspect</td>
<td>This topic addresses the issues related to using, adopting Cloud Computing, and migrating to the cloud, including cloud adoption, cloud migration, cloud broker, contracts &amp; SLAs, cost, pricing, billing, accounting.</td>
</tr>
<tr>
<td></td>
<td>Domain Aspect</td>
<td>This topic addresses the development of domain-specific applications on Cloud Computing, including eScience, weather applications, mobile applications.</td>
</tr>
<tr>
<td><strong>Contribution Type</strong></td>
<td>Method</td>
<td>The type of contribution is a new or an extended method, approach, process, procedure, technique, strategy, language, or algorithm.</td>
</tr>
<tr>
<td></td>
<td>System</td>
<td>The contribution type is the implementation of a novel software system, platform, environment, software tool, or simulation.</td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>The contribution is a software architecture, conceptual model or framework, system design.</td>
</tr>
<tr>
<td></td>
<td>Formal Study</td>
<td>The contribution type is a theory, or a formal analysis or measurement of some aspect of Cloud Computing, such as performance, reliability and failure rates. The result may be a metric or formalism.</td>
</tr>
<tr>
<td></td>
<td>Experience</td>
<td>The contribution type is a description of personal experience and lessons learned. It is closely related to experience papers.</td>
</tr>
<tr>
<td><strong>Research Type</strong></td>
<td>Evaluation Research</td>
<td>This type of research investigates a practical problem and provides an implemented solution. It uses a sound research method (case study, field study, field experiment, survey, or mathematical proof/logic reasoning) to validate the knowledge claim. The paper presents sufficient discussion of related work and provides insights into lessons learned.</td>
</tr>
<tr>
<td></td>
<td>Validation Research</td>
<td>This type of research differs from evaluation research in that the techniques investigated are novel and have not yet been implemented in practice. The investigation uses a thorough, methodologically sound research setup. Possible research methods are experiments, simulation, prototyping, mathematical analysis, mathematical proof of properties, etc.</td>
</tr>
<tr>
<td></td>
<td>Solution Proposal</td>
<td>This type of research proposes a solution technique and argues for its relevance, without a full-blown validation. The technique must be novel, or at least a significant improvement of an existing technique. A proof-of-concept may be offered by means of a small example, a sound argument, or by some other means.</td>
</tr>
<tr>
<td></td>
<td>Experience Paper</td>
<td>This type of research reports personal experiences or case studies of one or more projects that have been completed and provides lessons learned. The experience reported must be original and relevant to practitioners.</td>
</tr>
<tr>
<td></td>
<td>Philosophical Paper</td>
<td>This type of research is philosophical in nature, and sketches a new way of looking at existing things by structuring the field in the form of a taxonomy or conceptual framework.</td>
</tr>
<tr>
<td></td>
<td>Opinion Paper</td>
<td>This type of research contains the author’s opinion about what is wrong or good about something, how we should do something, etc.</td>
</tr>
</tbody>
</table>
Systematic Maps: An overview of the SM study, showing the number of papers for each category and across different years, are provided in three systematic maps, one for each facet of classification. These maps are represented as Matrix Bubble Charts\(^2\) (shown in Figures 2 – 4).

Paper Distributions: The distributions of the papers across different categories within each facet and the ratios of these distributions are summarized in three Tables, as shown in Tables 3 – 5.

Trend Analysis: The Trend Diagrams were used to depict the trends of the papers over time (Figures 5 – 7).

In Section 4, we analyze our mapping results and answer our research questions.

4. RESULTS

The results are presented in order of the research questions posed. For each question, we conduct the frequency and thematic analysis of the publications to reveal research gaps and trends.

4.1 Research Topics

This section answers the first research question RQ1: *What topics have been most investigated in the papers published in the past six CLOUD proceedings and how have these changed over time?*

To answer the first part of the question, we analyzed the distributions of papers by topics, as shown in Table 3. According to the table, the most investigated topic is “Functional Aspect” (197 papers, 38%), followed by “Non-Functional Aspect” (107 papers, 20%). The least investigated topic is “Domain Aspect” (21 papers, 4%), followed by “Adoption Aspect” (53 papers, 10%).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Aspect</td>
<td>197</td>
<td>38%</td>
</tr>
<tr>
<td>Non-Functional Aspect</td>
<td>107</td>
<td>20%</td>
</tr>
<tr>
<td>Architectural Aspect</td>
<td>75</td>
<td>14%</td>
</tr>
<tr>
<td>Resource Aspect</td>
<td>69</td>
<td>13%</td>
</tr>
<tr>
<td>Adoption Aspect</td>
<td>53</td>
<td>10%</td>
</tr>
<tr>
<td>Domain Aspect</td>
<td>22</td>
<td>4%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>523</td>
<td>100%</td>
</tr>
</tbody>
</table>

Functional and non-functional aspects are related to Cloud Computing technologies and so are architectural (75 papers).  

\(^2\) Our Matrix Bubble Charts are based on a template from: https://goo.gl/8EW0zD
papers, 14%), and resource aspects (69 papers, 13%). On the other hand, cloud adoption and domain applications are related to business issues of Cloud Computing. This suggests that the most heavily published papers at CLOUD conferences are in technological categories, whereas the least published papers are in business categories. One reason for this lopsided emphasis on the technology aspects of Cloud Computing might be that Cloud Computing is still a relatively new paradigm and hence still facing many technological challenges (Hogan et al., 2011; Pallis, 2010; Wei & Brian Blake, 2010; Q. Zhang, Cheng, & Boutaba, 2010).

However, a serious shortage of papers on business issues indicates a lack of business perspective in current Cloud Computing (Yang & Tate, 2012). Given that Cloud Computing has been widely viewed as “utility computing” (Buyya et al., 2009; Foster, Zhao, Raicu, & Lu, 2008), more research effort needs to be stepped up to reduce this rather worrying gap.

To answer the second part of the question, we analyzed the trends of papers over time. Figure 5 shows that since 2009 the number of papers on “Functional Aspect” has been growing rapidly, consistently and farther apart from papers in other categories. Figure 5 also shows that this growth trend will continue. By contrast, the number of papers on cloud adoption and domain applications has been consistently low over the past six years. Overall, the gap between the technology categories and the business categories has become even wider over the course of the past six years.

4.2 Contribution Types

This section answers the second research question RQ2: What types of contribution are most popular in these papers and how have these changed over time?

To answer the first part of the question, we analyzed the distributions of papers by contribution types, as shown in Table 4. According to this table, the most popular contribution type is “Method” (234 papers, 45%), followed by “System” (106, 20%). The least popular contribution type is “Experience” (6 papers, 1%).

We noted that the contributions on systems and models are predominately made by researchers from the industry (this information was obtained from the author affiliation of the papers). This finding is extremely positive, as it demonstrates that the CLOUD conferences have a strong industry engagement.

However, the shortage of papers in the “Experience” category indicates a lack of participation by practitioners in Cloud Computing. This is a rather significant research gap in current CLOUD conferences.

To answer the second part of the question, we analyzed the trends of the papers by contribution type over time. Figure 6 shows that since 2011, papers in the “Method” and “System” categories have continuously outnumbered other types of paper, and by stark contrast, the number of papers...
in the “Experience” category has almost been negligible. No papers on “Formal Study” and “Experience” were published in 2009. Since 2011 there has been a healthy growth in the “Formal Study” category. The number of papers on “Model” was peaked in 2010 but has since been reduced gradually. An interesting observation is that the decrease in the “Model” papers correlates with the increase in the “System” papers and this might indicate that some of the proposed models have subsequently been implemented into systems.

4.3 Research Types

This section answers the third research question RQ3: What types of research are dominating these publications and how have these changed over time?

To answer the first part of the question, we analyzed the distributions of papers by research types, as shown in Table 5. According to the table, “Validation Research” dominates the publication population with 361 papers (69% of all publications), while at the other end of the scale there are only 10 “Philosophical” papers (2% of all publications) and 6 “Experience” papers (1% of all publications). There is no “Opinion Paper” – perhaps this type of research has not been accepted by the CLOUD conferences so far.

The lack of experience papers is directly linked to the gap in the “Experience” contribution type, as described in Section 4.2.

The shortage of philosophical papers indicates a lack of theory building work in Cloud Computing. Since “Validation Research”, “Evaluation Research” and “Solution Proposal” can be grouped into empirical research, our mapping results have identified a vast gap in theoretical research.

<table>
<thead>
<tr>
<th>Research</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation Research</td>
<td>77</td>
<td>15%</td>
</tr>
<tr>
<td>Validation Research</td>
<td>361</td>
<td>69%</td>
</tr>
<tr>
<td>Solution Proposal</td>
<td>69</td>
<td>13%</td>
</tr>
<tr>
<td>Experience Paper</td>
<td>6</td>
<td>1%</td>
</tr>
<tr>
<td>Philosophical Paper</td>
<td>10</td>
<td>2%</td>
</tr>
<tr>
<td>Opinion Paper</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Total</td>
<td>523</td>
<td>100%</td>
</tr>
</tbody>
</table>

To answer the second part of the question, we analyzed the trends of the papers by research type over time. Figure 7 shows that since 2009, the use of “Validation Research” has risen sharply and grown further apart from other types of research. This trend seems to continue. The use of “Solution Proposal” type of research was peaked in 2011 (with 28 papers) and has since been declined. No “Experience” and “Philosophical” papers were published in 2009 and the both types of paper remain to be extremely scarce throughout the review period.

4.4 Interesting Correlations

During our study we have observed some interesting correlations between contribution types and research types:

- Between “Method” and “Validation Research”: Method papers tend to use validation research.
- Between “Formal Study” and “Validation Research”: Formal study papers also tend to use validation research.
- Between “System” and “Evaluation Research”: System papers tend to use evaluation research.
- Between “Model” and “Solution Proposal”: Model papers tend to use solution proposal research.
- Between “Experience” and “Experience Paper”: Experience papers are indeed experience papers.

These relationships may be useful for researchers to design their research, as by choosing a particular type of contribution to make, researchers can predetermine the type of research validation that needs to be undertaken. These relationships may also be useful for paper reviewers, as by understanding the type of contribution a paper claims to make, reviewers can expect the type of research validation to be reported in the paper.

5. DISCUSSION

In this section, we summarize our key findings and discuss validity concerns to our study.
5.1 Key Findings
The key findings obtained from answering the three research questions are summarized as follows.

The purpose of the first research question (RQ1) is to understand the collection of publications in terms of their research topics. Our answer shows that 86% of the papers addresses technology aspects (Functional Aspect, Non-functional Aspect, Architectural Aspect, and Resource Aspect) whereas only 14% papers dealing with business aspects (Adoption Aspect and Domain Aspect). This enormous gap must be closed in order for Cloud Computing to have an impact on businesses.

The second research question (RQ2) is to understand the purposes of these selected papers through their contribution types. Our answer shows that among five types of contribution, the Method type draws 45% of the papers whereas at the other end of the scale the Experience type only attracts 1% of the papers. This serious shortage of the contribution from practical experience indicates a lack of participation from Cloud Computing practitioners at the CLOUD conferences. A potential danger of this is the isolation of cloud research from practice.

The third research question (RQ3) is to investigate the type of research conducted by these papers. Our answer shows that Validation Research alone pulls 69% of the papers and outstripped all other types of research combined. Overall, 97% of the papers use empirical research methods. This trend might be acceptable, as Cloud Computing research is application oriented.

It is interesting to note that our key findings are consistent with the findings of Yang and Tate in their mapping study (Yang & Tate, 2012). This is rather significant, as the two mapping studies are conducted for two different periods, with theirs from 2008 to 2011 and ours from 2009 to 2014. Furthermore, the two mapping studies have two different search scopes, their scope as journal articles and ours as CLOUD conference papers. This suggests that our study, although being focused on a specific community, also reflects a general trend in cloud computing research.

5.2 Validity Concerns
A common threat to the validity of systematic mapping studies is that the papers are not exactly what “it says on the tin”. According to (Mendes, 2005), 73% of the papers were designed incorrectly: for example, they promised an experiment but had no experiment. The same problem was also reported by other authors (Jorgensen & Shepperd, 2007), who found that the term “experiment” was not always used in line with the definition of controlled experiments. In our mapping study, we found that the term “framework” was used loosely to mean a range of different things, such as method, model, architecture, formal study, and application development framework. Similarly, the term “method” was not always used in line with the definition of methods. Consequently, as noted in (Petersen et al., 2008) pointed out, systematic mapping studies, when not evaluating the papers in detail, may make judgmental errors and hence classify the papers into wrong categories.

In our study, we have mitigated this threat by conducting a more detailed evaluation to the papers that we could not establish their categories through wording. In addition, we performed the categorization iteratively by double-checking or treble checking of the mapping results. Yet, as stated in (Petersen et al., 2008), this threat can be minimized in systematic reviews which will dive deeper into the papers. Our mapping study has identified an extremely large number of papers for the Method type of contribution and an exceptionally high number of Validation Research papers, and future refinement could be to conduct systematic reviews in these two specific areas to gain a detailed insight into these papers.

6. Conclusions
This paper has made two important contributions to cloud computing research. Firstly, it has presented a major mapping study of the 532 papers published in the six previous CLOUD conference proceedings (from CLOUD 2009 to CLOUD 2014). To the best of our knowledge, this is the first study that covers the entire collection of full-length papers published in these proceedings. The mapping results obtained from this study, including research gaps, trends and themes (topics, contribution types and research types) can help the CLOUD conference community to shape its future research directions and to conduct further secondary studies.

Secondly, this paper has introduced Systematic Mapping to cloud computing research as a methodology for secondary studies and illustrated the process of this methodology step by step in our mapping study. The faceted classification schema developed for this study and the detailed analysis of the mapping results by means of different visual techniques are useful examples for cloud computing researchers to conduct future mapping studies.

In addition, this paper has made a small contribution to the development of a cloud computing taxonomy, by proposing a novel classification of cloud computing topics, based on different aspects of concerns, rather than specific technical issues. Such a classification is more natural and extensible.

Building on this mapping study, our next step is to conduct in-depth systematic reviews of specific areas, for example, studies of different aspects of cloud computing, under the topic facet. We also encourage our fellow researchers to do the same and together we can contribute to a cumulative research culture.
7. REFERENCES


Authors

Liping Zhao is a Senior Lecturer (Associate Professor) from the School of Computer Science, University of Manchester. Dr Zhao is an associate editor for Expert Systems—The Journal of Knowledge Engineering, Wiley-Blackwell, a member of the editorial board of Requirements Engineering Journal, Springer, and a member of editorial board of International Journal of Big Data. She served as a guest editor for Expert Systems—The Journal of Knowledge Engineering, Wiley-Blackwell in 2013 and a guest editor for International Journal on Services Technology and Management, Inderscience in 2011. She co-founded the UK Academic Network on Service Sciences in 2007 and has since been at the forefront of service science research and education. She is a member of IEEE Computer Society’s Technical Committee on Services Computing and has served as Work-in-Progress Track Chair and Session Chair for several editions of IEEE International Conference on Services Computing (SCC). Her other recent work includes promoting software patterns research and practice and in this capacity she co-chaired four editions of International Workshop on Requirements Patterns, from 2012 to 2015, all co-located with annual IEEE International Conference on Requirements Engineering.
published more than 100 scientific papers and received three IBM Faculty Awards for her outstanding contributions to software patterns and service sciences.

**Liang-Jie (LJ) Zhang** is Senior Vice President, Chief Scientist, and Director of Research at Kingdee International Software Group Company Limited. He has served as the President of Shenzhen Big Data Alliance since 2013. He also served as a director of The Open Group from 2011 to 2012. Prior to joining Kingdee, he was a Research Staff Member and Program Manager of Application Architectures and Realization at IBM Thomas J. Watson Research Center as well as the Chief Architect of Industrial Standards at IBM Software Group. Dr. Zhang has published more than 160 technical papers in journals, book chapters, and conference proceedings. He has 50 granted patents. Dr. Zhang received his Ph.D. on Pattern Recognition and Intelligent Control from Tsinghua University. He chaired the IEEE Computer Society's Technical Committee on Services Computing from 2003 to 2011. He also chaired the Services Computing Professional Interest Community at IBM Research from 2004 to 2006. Dr. Zhang has served as the Editor-in-Chief of the International Journal of Web Services Research since 2003 and was the founding Editor-in-Chief of IEEE Transactions on Services Computing. He was elected as an IEEE Fellow in 2011, and in the same year won the Technical Achievement Award “for pioneering contributions to Application Design Techniques in Services Computing” from IEEE Computer Society. Dr. Zhang also chaired the 2013 IEEE International Congress on Big Data (BIGDATA Congress) and the 2009 IEEE International Conference on Cloud Computing (CLOUD 2009).

**Tina Liu** obtained her Master in Science degree in Mathematics and Computing, at Imperial College London in 2014. She is currently a technical analyst in the Trading Technologies department of Credit Suisse, London. Her research interests include data analytics, machine learning, statistical modelling, pattern recognition, and logic and reasoning.
**Boosting Metrics: Measuring Cloud Services from the Holistic Perspective**

Zheng Li, Liam O’Brien, and He Zhang

1 School of Computer Science, Australian National University and NICTA, Australia
2 ICT Innovation and Services, Geoscience Australia, Canberra, Australia
3 Software Institute, Nanjing University, Jiangsu, China

zheng.li@anu.edu.au
liamob99@hotmail.com
dr.hezhang@gmail.com

**Abstract**

Studies have shown that Cloud services evaluation would be crucial and beneficial for both service customers and providers, and metrics would play a vital role in any evaluation implementation. Considering the numerous and various aspects of Cloud services, a frequent suggestion is to perform evaluation from a holistic view. The currently normal strategy of holistic evaluation is to use a set of metrics along with a suite of benchmarks to conduct separated experiments. Given the separated, diverse, and even possibly conflicting measurement criteria, it could be still hard for customers with such evaluation reports to understand an evaluated Cloud service from a global perspective. Inspired by the boosting approaches to machine learning, we proposed the concept Boosting Metrics to represent all the potential approaches that are able to deliver summary measurement of Cloud services. Essentially, the idea of boosting metrics is to holistically measure Cloud services with concern of service properties, which supplements the strategy of employing benchmark suites that is to holistically evaluate Cloud services with concern of different workloads. This paper introduces two types of preliminary approaches, and unifies a set of sophisticated measurements into the notion of boosting metrics. In particular, we show that boosting metrics can be used as a summary Response for applying experimental design to Cloud services evaluation. Although the concept Boosting Metrics was refined based on our work in the Cloud Computing domain, we believe it can be easily adapted to the evaluation work of other computing paradigms.

**Keywords:** Cloud Computing; Cloud Services Evaluation; Measurement Criteria; Boosting Metrics; Experimental Design

---

1. **Introduction**

As Cloud Computing becomes one of the most promising computing paradigms in industry (Buyya et al., 2009), numerous vendors have started to supply public Cloud infrastructures and services with different terminologies, qualities, and cost models (Prodan and Ostermann, 2009). Since most providers do not reveal details about their infrastructures (Brooks, 2010), customers have little knowledge and control over the precise nature of public Cloud services even in the “locked down” environment (Sobel et al., 2008). As such, Cloud services evaluation would be crucial and beneficial for both service customers and providers (Li et al., 2010). For example, proper performance evaluation of candidate Cloud services can help customers perform cost-benefit analysis and decision making for service selection, while it can also help providers improve their service qualities against competitors.

When it comes to evaluating a Cloud service, suitable measurement criteria or metrics must be chosen. In fact, according to the rich research in the evaluation of traditional computer systems, the selection of metrics plays an essential role in evaluation implementations (Li et al., 2012b). Particularly, it is often useful and significant to evaluate Cloud services from a holistic view (Iosup et al., 2011; Rabl et al., 2010), and using single measurement index would be helpful and convenient for comparing alternatives and drawing conclusions (Islam et al., 2012). More importantly, a single index of an overall measurement can play a summary Response role in experimental design and analysis (Montgomery, 2009) for evaluating Cloud services.

Unfortunately, given the numerous and various aspects of Cloud services, the existing evaluation studies usually adopted multiple metrics to measure Cloud services feature by feature. To perform a holistic evaluation, the normal strategy is to employ a benchmark suite to cover and test various aspects of Cloud services from the perspective of workload. Through reviewing the relevant studies (Li et al., 2013c), we found that most evaluators intended to report individual benchmarking results with a lack of visibly integrated measurements. As a result, customers with such evaluation reports could have to further summarize various evaluation results by themselves. Although it is possible and sometimes flexible for customers to balance tradeoffs in employing a Cloud service, a single and overall index can significantly facilitate customers’ decision making by supplementing separated benchmarking results.
Therefore, it is valuable and necessary to investigate approaches to summary measurement of Cloud services. Moreover, considering different evaluation circumstances, we may expect different measurement approaches ranging from integrating homogeneous benchmarking results (Evangelinos and Hill, 2008) to catering for conflicting criteria (Zhang et al., 2012), e.g., the combination of performance and cost. Inspired by the boosting approaches to machine learning that combine weak rules into a single more accurate one (Schapire, 2002), we proposed the concept Boosting Metrics to represent all the potential approaches that are able to integrate a set of separated measures, as defined below.

A boosting metric is a secondary measurement criterion by manipulating the primary metrics that directly measure individual Cloud service aspects.

This paper has been significantly extended from our previous work (Li et al., 2013a) that only discussed preliminary approaches integrating a suite of benchmarking results. Firstly, we have essentially expanded the concept of Boosting Metric to cover the integration of not only homogeneous benchmarking results but also diverse and even conflicting measurement criteria. Secondly, in addition to suggesting preliminary approaches, we also identified the same backend essence of a set of sophisticated Cloud service measurements, and unified them into the notion of Boosting Metric. Thirdly, to perform convincing demonstration, we followed a formal Cloud evaluation methodology and updated the case study about how the boosting metrics can help facilitate experimental design and analysis. In summary, the contributions of this paper are threefold, as listed below.

- The proposed concept Boosting Metric essentially suggests holistic measurement of Cloud services from the output (service property) perspective, which supplements the input (workload) perspective of employing benchmark suites. Note that, although this idea was extracted from our work in the Cloud computing domain, we believe it can be easily adapted to the evaluation work of other computing paradigms.
- The summarized preliminary and sophisticated boosting metrics can be directly used in the future Cloud services evaluation work. Evaluators can also use them to inspire and develop new measurement approaches.
- The case study can be viewed as a template for the usage of boosting metrics in a rigorous procedure of Cloud services evaluation. Meanwhile, this case study also acts as a concrete application for validating the employed Cloud evaluation methodology.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the related work about holistic measurement of Cloud services. By employing a simple application scenario of using benchmark suites to evaluate Cloud services, Section 3 introduces two types of preliminary approaches to boosting metrics. Section 4 unifies four different and sophisticated Cloud service measurements into the notion of Boosting Metric. Section 5 employs a case study of evaluating Amazon EC2 to demonstrate how a boosting metric can be used in evaluation and how it can help analyze experimental results. Conclusions and some future work are discussed in Section 6.

2. RELATED WORK

Since the selection of measurement criteria or metrics plays an essential role in evaluation implementations (Obaidat and Boudriga, 2009), a set of suitable metrics must be chosen when evaluating Cloud services. Li et al. (2012b) have investigated the existing relevant studies and established a metric catalogue to facilitate metrics selection for Cloud services evaluation. In essence, the metric catalogue accommodates the de facto evaluation metrics in the Cloud computing domain, and the metrics have been categorized according to different Cloud service features. As such, evaluators can use a particular feature as the retrieval key to quickly locate candidate evaluation metrics in this catalogue. However, such a catalogue also shows that Cloud services are usually measured feature by feature. There seems a lack of approaches to summary measurement of multiple and even overall Cloud service features. For example, Cloud elasticity is related not only to the resource scaling time but also to the resource charging basis (Islam et al., 2012); consequently, it has become a challenge to explicitly quantify the amount of elasticity of a Cloud service (Li et al., 2012b, 2013c).

Therefore, to cover and test various service aspects from a holistic view, the current practitioners normally suggest employing benchmark suites for Cloud services evaluation (Iosup et al., 2011; Rabl et al., 2010). For example, the kernel benchmarks in NPB have been used to reveal different micro features of Amazon EC2 like computation, communication and storage respectively (Akioka and Muraoka, 2010); while six scale-out workloads are collected to simulate different macro application scenarios in today’s Cloud infrastructure (Ferdman et al., 2012). In particular, for verifying scientific computing in the Cloud, HPCC seems a popular benchmark suite to show high performance computing capabilities of Cloud services (Jackson et al., 2010; Ostermann et al., 2006). In addition to those predefined benchmark suites, the evaluator-selected application sets were also commonly adopted to evaluate Cloud services (Dejun et al., 2009; Jackson et al., 2010). Essentially, each application set here can be viewed as an individual benchmark suite.
However, even if benchmark suites were adopted to evaluate Cloud services, most evaluators intended to report individual benchmarking results without visibly integrated measurements (Li et al., 2013c). In fact, although it is often useful and convenient to compare alternatives by using a single index (Islam et al., 2012), the overall measurement of Cloud services would be significantly challenging because of the diverse and even conflicting criteria (Zhang et al., 2012).

3. Preliminary Boosting Metrics for Cloud Services Evaluation

As mentioned previously, we borrowed the “boosting” idea from the machine learning field to our Cloud services evaluation work. In machine learning, boosting refers to the method of producing a more accurate prediction rule by combining a set of rough and less accurate rules of thumb (Schapire, 2002). By analogy, in Cloud services evaluation, we treat “boosting” as integrating a set of local measures into a single global one, namely a boosting metric, to reflect the overall measurement of a Cloud service. A simple and typical application scenario is the employment of benchmark suites. When measuring different Cloud service aspects, a benchmark suite may adopt homogeneous primary metrics (e.g. NPB (NASA, 2012)) or inhomogeneous primary metrics (e.g. HPCC (Luszczek et al., 2006)). Correspondingly, here we show two types of preliminary boosting metrics, as described in the following two subsections respectively.

3.1 Mean as a Boosting Metric from a Spatial Perspective

In a benchmark suite, a set of different benchmarks are generally expected to be able to reflect different aspects of a Cloud service in an evaluation. If we imagine every single service aspect as an individual dimension, a Cloud service with n aspects can be represented as a Euclidean n-space. As such, when evaluating a Cloud service, the benchmarking results together would identify a particular point in the Euclidean n-space, which essentially uses a tuple to reflect the overall feature of the Cloud service with respect to the corresponding benchmark suite, as illustrated in Figure 1.

As previously mentioned, a boosting metric is supposed to represent overall service feature by using a single number instead of using a tuple. Thus, seeking boosting metrics here is to find single-number representations of the benchmarking point in the service-aspect space. Since the benchmarking point and the origin can determine a rectangular parallelepiped in the Euclidean n-space (cf. Figure 1), we can switch our focus from the coordinates of the point to the attributes of the corresponding rectangular parallelepiped, such as the perimeter, surface area, volume, etc. Furthermore, given the related work, we tried to rationalize several “classic” means (Cantrell, 2012) to suit the rectangular parallelepiped’s attributes, rather than reinventing “new” measures. The equations of the selected means are listed below, where Benchmarking_i denotes the benchmarking result by using the ith benchmark in a suite.

$$\text{Arithmetic Mean} = \frac{\sum_{i=1}^{n} \text{Benchmarking}_i}{n} \quad (1)$$

$$\text{Geometric Mean} = \sqrt[n]{\prod_{i=1}^{n} \text{Benchmarking}_i} \quad (2)$$

$$\text{Harmonic Mean} = \frac{n \times \prod_{j=1}^{n} \text{Benchmarking}_j}{\sum_{j=1}^{n} \frac{\text{Benchmarking}_j}{\text{Benchmarking}_i}} \quad (3)$$

Figure 1. The benchmarking results point in a Cloud service aspect space

Arithmetic Mean. Corresponding to that the perimeter of a rectangular parallelepiped is the sum of its side lengths, we may use the arithmetic mean as a potential boosting metric, as shown in Equation (1).

Geometric Mean. Corresponding to that the volume of a rectangular parallelepiped is the product of its side lengths, we may use the geometric mean as a candidate boosting metric, as shown in Equation (2).

Harmonic Mean. In particular, corresponding to the rate between the volume and surface area of a rectangular parallelepiped, we may use the harmonic mean as a candidate boosting metric, as shown in Equation (3).
Quadratic Mean. Corresponding to the distance between the benchmarking point and the origin, we may use the quadratic mean as a candidate boosting metric, as shown in Equation (4).

\[
\text{Quadratic Mean} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \text{Benchmarking}_i^2}
\]  

As can be seen, it is convenient to calculate these means of a set of benchmarking results to reflect the summary feature of a Cloud service. Interestingly, the Geometric Mean seems the most popular one in practice (Evangelinos and Hill, 2008; Jackson et al., 2010). Nevertheless, there is a default constraint when employing means as boosting metrics: different Cloud service aspects should be homogeneously measured by using different benchmarks in a suite. In other words, to calculate means (secondary metrics), different benchmarking results for different Cloud service aspects must adopt the same primary metric. If the constraint cannot be satisfied, we may employ a more generic solution – Radar Plot, as specified in the following subsection.

3.2 Radar Plot as a Boosting Metric

Radar plot is a simple but intuitive graphical tool that can simultaneously depict a group of different types of values relative to a central point. When a benchmark suite uses different primary metrics to measure different Cloud service aspects, we can use radar plot to represent the benchmarking results over a predefined baseline. In particular, we can also portray several groups of standardized benchmarking results in one radar plot without predefining any baseline (cf. Figure 2). Given the analysis of the existing metrics for Cloud services evaluation (Li et al., 2012b), here we elaborate two standardization methods of the existing metrics for Cloud services evaluation (Li et al., 2012b), here we elaborate two standardization methods respectively.

\[
\text{HB}_\text{Standardized}_i = \frac{\text{Benchmarking}_i}{\text{MAX(Benchmarking}_{1,n})}
\]

LB_Standardized
\[
\text{LB}_\text{Standardized}_i = \frac{1}{\text{MAX(Benchmarking}_{1,n})}
\]

Equation (5) is for the standardization of HB metrics, while Equation (6) for LB metrics. Here Standardized refers to the standardized ith benchmarking result Benchmarking. In fact, Equation (6) offers LB metrics a higher better representation through reciprocal standardization, so that all the standardized benchmarking results can be settled homogeneously higher better in a radar plot, and meanwhile constructs a bigger-area better polygon. As such, we can intuitively contrast the areas of different polygons to compare different groups of benchmarking results. Moreover, the area of a polygon can be regarded as a single numerical Response to facilitate experimental design and analysis. Suppose there are n benchmarking results standardized and marked in a radar plot, we can calculate the area of the corresponding polygon by summing up areas of the n adjacent triangles, as shown in Equation 7.

\[
\text{Area} = \sum_{i=1}^{n} \frac{\sin(\frac{2\pi}{n}) \times \text{Standardized}_i \times \text{Standardized}_{\text{adj}(i,n)}}{2}
\]

Here we employ a real case to demonstrate the radar plot as a boosting metric. For our convenience, the evaluation data reported in (Ostermann et al., 2006) are directly reused, as shown in Table 1. Given the various types of benchmarking results, such as HPL, STREAM, RandomAccess, Latency, and Bandwidth, within the HPCC benchmark suite (Luszczek et al., 2006), it is hard to compare the summary performance as a whole when evaluating different types of EC2 instances.

<table>
<thead>
<tr>
<th>Name</th>
<th>m1.large</th>
<th>m1.xlarge</th>
<th>c1.medium</th>
<th>c1.xlarge</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPL (GFLOPS)</td>
<td>7.15</td>
<td>11.38</td>
<td>3.91</td>
<td>51.58</td>
</tr>
<tr>
<td>STREAM (GBps)</td>
<td>2.38</td>
<td>3.47</td>
<td>3.84</td>
<td>15.65</td>
</tr>
<tr>
<td>RandomAccess (MUPS)</td>
<td>54.35</td>
<td>168.64</td>
<td>46.73</td>
<td>249.66</td>
</tr>
<tr>
<td>Latency (µs)</td>
<td>20.48</td>
<td>17.87</td>
<td>13.92</td>
<td>14.19</td>
</tr>
<tr>
<td>Bandwidth (GBps)</td>
<td>0.7</td>
<td>0.92</td>
<td>2.07</td>
<td>1.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>m1.large</th>
<th>m1.xlarge</th>
<th>c1.medium</th>
<th>c1.xlarge</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPL</td>
<td>0.1386</td>
<td>0.2206</td>
<td>0.0758</td>
<td>1</td>
</tr>
<tr>
<td>STREAM</td>
<td>0.1521</td>
<td>0.2217</td>
<td>0.2454</td>
<td>1</td>
</tr>
<tr>
<td>RandomAccess</td>
<td>0.2177</td>
<td>0.6755</td>
<td>0.1872</td>
<td>1</td>
</tr>
<tr>
<td>Latency</td>
<td>0.6797</td>
<td>0.779</td>
<td>1</td>
<td>0.981</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>0.3382</td>
<td>0.4444</td>
<td>1</td>
<td>0.7198</td>
</tr>
</tbody>
</table>

Thus, we first standardize those benchmarking results respectively, as listed in Table 2. Note that the generated numbers in Table 2 do not come with any benchmarking unit. Then, the standardized benchmarking results are represented in a radar plot, as illustrated in Figure 2. Through this radar plot, we can intuitively and conveniently identify that: c1.xlarge has absolutely better overall performance than m1.large and m1.xlarge; c1.xlarge is also better than c1.medium in general, while slightly poorer in...
terms of Bandwidth and Latency. In particular, the areas of different benchmarking polygons in the radar plot are further calculated to quantitatively reflect the summary performance of the four types of EC2 instances, as bracketed beside the legend entries. Essentially, the numerical areas may play a Response role in the design of experiments for Cloud services evaluation. A complete experimental design and analysis sample by using boosting metrics is elaborated in the next section.

![Radar Plot of the standardized HPCC benchmarking results](image)

**Figure 2. The Radar Plot of the standardized HPCC benchmarking results**

4. **SOPHISTICATED BOOSTING METRICS FOR CLOUD SERVICES EVALUATION**

Although not common, the idea of boosting metrics has been intuitively employed in some Cloud services evaluation work, together with a little preliminary discussion about the merits of employing boosting metrics. For example, the geometric mean of eight NAS Parallel Benchmarks (NPB) results (BT, CG, FT, IS, LU, MG, SP, UA) was used to measure the computational performance of Amazon EC2 on a wide set of model applications and kernels (Evangelinos and Hill, 2008). Interestingly, some sophisticated measurements of Cloud services can also be unified into the notion of Boosting Metric. In other words, we may identify the same essence behind some completely different approaches to Cloud service measurement. Four samples are demonstrated in the following subsections.

4.1 **SUSTAINED SYSTEM PERFORMANCE (SSP)**

Considering the diversity of users’ requirements in a supercomputer center, Jackson et al. (2010) employed seven typical scientific applications as a benchmark suite to evaluate the Amazon Cloud services for high performance computing. These applications span a range of science domain, parallelization schemes, concurrences, and machine-based characteristics (e.g., communication, computation, memory, and storage). To better represent the effectiveness of Cloud services for delivered performance on applications rather than peak FLOP rates, the authors proposed an aggregate measure of computing system performance, namely Sustained System Performance (SSP) metric, as shown in Equation 8.

\[
SSP = N \left( \prod_{i=1}^{M} P_i \right)^{1/M} 
\]  

In detail, evaluators may select \( M \) applications as a set of benchmarks; \( P_i \) indicates the application \( i \)'s performance expressed in units of GFlops per second per core; the size of the evaluated computing system is considered as the number \( N \) of its computational cores. As such, the calculation of SSP is to multiply the geometric mean of individual applications’ performance per CPU core by the number of computational cores, which can be viewed as an extended-Geometric Mean-based boosting metric.

4.2 **PENALTY MODEL**

As mentioned previously, it has been identified that evaluating elasticity of a Cloud service is not trivial (Kossmann and Kraska, 2010), because Cloud elasticity is related to both the resource scaling time and the resource cost (Islam et al., 2012). By integrating relevant basic Quality of Service (QoS) metrics to monitor the requested Cloud resources, Islam et al. (2012) suggested a Penalty Model to measure the imperfections in elasticity of Cloud services for a given workload, as shown in Equation 9.

\[
P = \frac{P_o(t_o, t_e) + P_d(t_e, t_o)}{t_e - t_o} 
\]  

Where \( P_o(t_o, t_e) \) refers to the penalty for over-provisioning, which captures the cost of provisioned but unutilized resources for a period starting at \( t_o \) and ending at \( t_e \); while \( P_d(t_e, t_o) \) measures opportunity cost from the performance degradation that arises with under-provisioning during the same period. Note that, the authors assume that each Cloud resource type can be allocated in units, and users can learn the resource allocation level with relevant QoS measurement results. Eventually, the Penalty Model acts as a boosting metric, and delivers a single penalty score \( P \) in monetary units over the time interval \([t_o, t_e] \).

4.3 **CUSTOMER SATISFACTION**

Given the uncertainty in the runtime of Cloud services, the existing Service Level Agreements (SLAs) often lack providing comprehensive information about the overall performance of a service regarding specific tasks (Lenk et al., 2011). Similar to the aforementioned Penalty Model, a utility theory-inspired model was developed for measuring Customer Satisfaction in the Cloud, while Customer
Satisfaction was treated as an explicit metric to support utility-based SLAs in order to balance the performance of applications and the cost of running them (Chen et al., 2011). In this case, the authors considered different customer satisfactions as different combinations of service price and request processing time, as shown in Equation 10.

\[
U(p,t) = U_0 - ap - \beta t
\]

in which, the satisfaction or utility \(U\) of using a service is defined as a function of the service price \(p\) and the response time \(t\); \(U_0\) is the maximum utility that the service delivers to the customer, and it is proportional to the size of the service request; while \(a\) and \(\beta\) are coefficients, and \(a/\beta\) is known as marginal rate of substitution in economics, denoting the rate at which the customer is willing to give up response time in exchange for service price without any satisfaction change.

### 4.4 Analytic Hierarchy Process (AHP) Based Decision

The overall measurement of Cloud services could have to cater for a number of conflicting criteria, e.g., performance and cost, and the problem would be further aggravated by the fact that different applications have heterogeneous QoS requirements. By treating the Cloud service measurement as a decision-making problem, Zhang et al. (2012) suggested the multi-criteria decision-making technique Analytic Hierarchy Process (AHP) to handle various and mixed qualitative and quantitative criteria. AHP is based on pair-wise comparisons of the criteria. For each pair of criteria, the administrator is required to provide a subjective opinion of their relative importance. By converting the subjective opinions to numerical values, AHP finally generates numerical priorities for all the decision alternatives. In other words, AHP can supply straightforward indexes as measures of candidate Cloud services to customers for their service selection. Due to the limit of space, we do not elaborate the detailed mechanism of AHP in this paper.

### 5. A Case Study of Using Boosting Metrics in Experimental Design and Analysis

<table>
<thead>
<tr>
<th>Specification</th>
<th>m1.xlarge</th>
<th>M2.xlarge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Amount</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>ECU Amount</td>
<td>8</td>
<td>6.5</td>
</tr>
<tr>
<td>Network I/O</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory Size</td>
<td>15 GB</td>
<td>17.1 GB</td>
</tr>
<tr>
<td>Platform</td>
<td>64 bit</td>
<td>64 bit</td>
</tr>
<tr>
<td>Storage Size</td>
<td>1690 GB</td>
<td>420 GB</td>
</tr>
<tr>
<td>Windows Usage Cost</td>
<td>$0.92 per Hour</td>
<td>$0.57 per Hour</td>
</tr>
</tbody>
</table>

Cloud Services to be Evaluated. As mentioned previously, the evaluation requirement in this case can be viewed as merely a rough understanding of the parallel

---

**Figure 3. CEEM for Cloud services evaluation (Li et al., 2013b)**

Here we use a case study of evaluating Amazon EC2 to demonstrate how a boosting metric can be used in evaluation and how it can help analyze experimental results. To achieve convincing demonstration, we followed the ten-step Cloud Evaluation Experiment Methodology (CEEM, cf. Figure 3) (Li et al., 2013b) to perform the evaluation.

### 5.1 Requirement Recognition and Service Feature Identification

**Problem and Motivation.** We proposed to use a set of Amazon EC2 instances to perform a small-scale parallel computing project. According to the estimation of our project and the predefined EC2 instance types (Amazon, 2013), we initially selected m1.xlarge and m2.xlarge as two candidate types of parallel computing nodes. The specifications and prices of these two EC2 instance types are listed in Table 3. When making decision to choose the most suitable alternative from these two options, we found that it was hard to directly distinguish the better one based on their specifications. Unlike the clear differences between the other types of EC2 instances, each of these two options has its own distinctions. For example, m1.xlarge seems overall better than m2.xlarge, while m2.xlarge has faster single CPU core, larger memory, and lower price. As such, we decided to evaluate these two types of EC2 instances to roughly compare their potential performance in our project. It is then suitable to consider boosting metrics for summary measurement in this case.

**Table 3. Specifications and Prices of Two EC2 Instance Types**
computing capability of those two instance types. To save

time, we decided to perform evaluation for each option only

on a single EC2 instance rather than on a real parallel cluster

environment. In detail, we applied one m1.xlarge instance

and one m2.xlarge instance respectively from Amazon’s

US-EAST-1 region, and both instances came with the same

quick launch Amazon Machine Image (AMI) – 64 bit

Windows Server 2008 Base.

5.2 Benchmark & Metrics Listing and Selection

As a well-known and well-accepted parallel computing

benchmark suite, NPB has been widely used for Scientific

Computing evaluation in the public Cloud (Akioka and

Muraoka, 2010; Carlyle et al., 2010; Evangelinos and Hill,

2008; He et al., 2010; Walker, 2008). Therefore, we also

employed NPB as the benchmark suite to evaluate the

summary performance of EC2 instances for our project. In

particular, since the software system in our project was

implemented using JAVA, we selected the latest JAVA


Although different benchmarks in NPB are used to reflect

different features of a computing system like computation,

communication and storage, all the NPB benchmarking

results adopt the same format with homogeneous metrics,

such as benchmark runtime (time in seconds) and

benchmark FLOP rate (floating point Mops total).

Following the popular choice (cf. Section 4), we also chose

Geometric Mean as the boosting metric over the primary

metrics benchmark runtime and benchmark FLOP rate in

this case study.

5.3 Experimental Factors Listing and Selection

Before evaluating a system, experimental factors

identification is a tedious but necessary task (Le Boudec,

2010). Factors here refer to the elements in the system or the

workload that may influence the evaluation result. In fact,

our previous work has established a factor framework for

Cloud services evaluation, and the latest framework version

capsules the state-of-the-practice of performance evaluation

factors that people currently take into account in the Cloud

Computing domain (Li et al., 2012a). Since this evaluation

work would also measure performance of EC2 instances, we

conventionally identified experimental factors within the

proposed framework. In detail, we explored experimental

factors related to Cloud resource and benchmark’s workload

respectively: Instance Type (m1.xlarge vs. m2.xlarge),

Thread Number (2 vs. 4), and Workload Size (Class W vs.

Class A).

5.4 Experimental Design

When it comes to experimental design, there are three

basic principles: Randomization, Replication, and Blocking

(Montgomery, 2009). In this case, we only focus on the

Randomization and Replication. Although an entire NPB

suite run can be treated as a block, here we try to simplify

the demonstration without elaborating sophisticated design

approaches. The detailed designing process is then

composed of three steps, as specified below.

Determining Individual Experimental Trials. In this

evaluation work, an experimental trial indicates a specific

benchmark run on an EC2 instance. In practice, we used one

batch command to drive a single NPB benchmark run

during the experiments. Thus, a series of batch commands

were listed to represent different individual experimental

trials.

Determining Amount of Experimental Trials. As

mentioned previously, we decided to investigate two levels

of Workload Size (Class A and W) and two levels of Thread

Number (2 and 4) for two Instance Types (m1.xlarge and

m2.xlarge). To facilitate the investigation, we also planned

benchmarking with single thread as a reference baseline. On

the other hand, the JAVA-version NPB suite comprises

seven benchmarks. According to our pilot test of running

those seven benchmarks on a local machine, we decided to

replicate all the different trials five times. Therefore, there

are 2×3×7×5=210 experimental trials in total on each

instance.

Determining Sequence of Experimental Trials. To

achieve a randomized trial sequence, we assigned two

random numbers to each trial-associated batch command in

EXCEL. The 210 batch commands can be ordered by one

random number and another in turn, to run experiments on

the m1.xlarge and m2.xlarge instances respectively.

Through such a randomization, we made individual trials as

independent as possible between each other to reduce the

experimental sequence-related bias.

5.5 Experimental Implementation

Due to the limit of space, the specific experimental

results from individual NPB benchmarks are not reported in

this paper. In summary, by averaging results of identical

experimental trials, and dividing the trials into different

workload-size, thread-number, and instance-type groups, we

obtained a set of Runtime and FLOP Rate geometric means

with respect to the NPB suite under different conditions, as

listed in Table 4.

<table>
<thead>
<tr>
<th>Cloud Resource</th>
<th>Workload</th>
<th>Boosting Metric (Geometric Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NPB Runtime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(second)</td>
</tr>
<tr>
<td>EC2 m1.xlarge</td>
<td>1 Thread</td>
<td>Class W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td></td>
<td>2 Thread</td>
<td>Class W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td></td>
<td>4 Thread</td>
<td>Class W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class A</td>
</tr>
</tbody>
</table>
To intuitively show the instances’ performance changing when varying conditions, we also used four line charts to represent the boosting metric’s measurements, as illustrated in Figure 4. It is not surprising that, benefiting from the faster single core, the m2.xlarge instance defeats the m1.xlarge instance for running NPB suite before over-saturating its CPU cores, while the m1.xlarge instance performs better with four-thread trials. Nevertheless, it is still hard to tell whether Instance Type is a significant factor or not in general. Therefore, we employed formal experimental-analysis techniques to unfold further investigation, as explained in the following subsection.

Figure 4. Illustration of geometric means of NPB3.0-JAV benchmarking results with different circumstances.

5.6 EXPERIMENTAL ANALYSIS

Since only two levels of an experimental factor were particularly concerned in this case (cf. Subsection 5.3), we naturally adopted the optimal design and analysis technique, namely Full-factorial $2^k$ Design (Montgomery, 2009), to analyze the experimental results. Given the three factors considered, a pseudo $2^3$ design matrix was generated as shown in Table 5. The response columns in the matrix were filled with pseudo-trial results that correspond to eight geometric means in Table 4. For conciseness, we further assigned aliases to those experimental factors and responses, as listed below.

- Factor A: Instance Type (m1.xlarge vs. m2.xlarge).
- Factor B: Thread Number (2 vs. 4).
- Factor C: Workload Size (Class W vs. Class A).
- Response R1: NPB Runtime (seconds).
- Response R2: NPB FLOP Rate (Mops).

Recall that the analysis is to investigate if Instance Type (A) (or other factors) significantly influences the benchmarking results. By setting the significance level $\alpha$ as 0.05 (Jackson, 2011), we can draw Pareto plots (Antony, 2003) to detect the factor and interaction effects that are important to the parallel computing (NPB suite in this case), as shown in Figure 5. To save space, we do not elaborate the backend statistics here. In brief, given a particular significance level, Pareto plot displays a red reference line besides the effect values. Any effect that extends past the reference line is potentially important (Antony, 2003).
Since the effect of factor Workload Size (C) is beyond the reference line in Figure 5a, it is apparent that Workload Size (C) dominates the runtime of NPB suite. On the contrary, the factor Instance Type (A) has little influence on the benchmark runtime in this case. As for the FLOP Rate analysis in Figure 5b, we show that none of the factor or interaction effects significantly influences the transaction speed. However, relatively speaking, Thread Number (B) is the most important to the NPB FLOP Rate, while Instance Type (A) is still the least important factor. Therefore, for our proposed parallel computing project, we are now suggested to pay more attention to the workload size to distinguish between those two EC2 instance types.

From Figure 4, it is clear that increasing thread numbers will not increase computing performance if an instance’s CPU cores are already saturated, especially with larger workload. However, increasing workload size seems to be able to continually increase the performance difference between two instances at 4 or more threads, which was further confirmed by running a supplementary experiment with NPB’s 4X larger workload Class B. To facilitate analysis, we calculated different performance improvements of switching from m2.xlarge to m1.xlarge at 4 threads by using Equation (11), as listed in Table 6. Note that we use the minimum performance value between the two instances as the denominator to avoid the Ratio Game bias (Jain, 1991).

\[ I = \frac{|Performance_{m2} - Performance_{m1}|}{\text{MIN}(Performance_{m1}, Performance_{m2})} \times 100\% \]  

Given the price increase of switching from m2.xlarge to m1.xlarge (0.95–0.57)/0.57×100%=61.4%), the instance-hour for running m2.xlarge is 1.614 times higher than running m1.xlarge within the same budget. In other words, m2.xlarge always has a cost advantage over m1.xlarge until the performance improvement reaches 61.4%, although the total runtime may be longer. According to the previous analysis, we finally decided to choose m2.xlarge as the cost-wise option for our small-scale parallel computing project.

### Table 5. A Full-factorial (2^3) Design Matrix for This Case Study

<table>
<thead>
<tr>
<th>Trial</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>R1 (second)</th>
<th>R2 (Mops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>m1</td>
<td>2</td>
<td>W</td>
<td>3.727</td>
<td>299.813</td>
</tr>
<tr>
<td>2</td>
<td>m1</td>
<td>4</td>
<td>A</td>
<td>18.138</td>
<td>513.873</td>
</tr>
<tr>
<td>3</td>
<td>m2</td>
<td>2</td>
<td>W</td>
<td>3.401</td>
<td>351.003</td>
</tr>
<tr>
<td>4</td>
<td>m1</td>
<td>2</td>
<td>A</td>
<td>31.176</td>
<td>298.949</td>
</tr>
<tr>
<td>5</td>
<td>m2</td>
<td>2</td>
<td>A</td>
<td>24.537</td>
<td>379.765</td>
</tr>
<tr>
<td>6</td>
<td>m2</td>
<td>4</td>
<td>A</td>
<td>25.32</td>
<td>368.289</td>
</tr>
<tr>
<td>7</td>
<td>m1</td>
<td>4</td>
<td>W</td>
<td>2.73</td>
<td>412.717</td>
</tr>
<tr>
<td>8</td>
<td>m2</td>
<td>4</td>
<td>W</td>
<td>2.987</td>
<td>373.948</td>
</tr>
</tbody>
</table>

### Table 6. Performance Improvement of Switching from M2.xlarge to M1.xlarge at 4 Threads

<table>
<thead>
<tr>
<th>Workload</th>
<th>Performance Improvement</th>
<th>NPB FLOP Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class W</td>
<td>9.4%</td>
<td>10.4%</td>
</tr>
<tr>
<td>Class A</td>
<td>39.6%</td>
<td>39.5%</td>
</tr>
<tr>
<td>Class B</td>
<td>48.4%</td>
<td>48.4%</td>
</tr>
</tbody>
</table>

### 6. Conclusions and Future Work

Metrics play a vital role in any evaluation implementation. To evaluate Cloud services from a holistic view, a widely adopted strategy is to use a set of metrics along with a suite of benchmarks to perform separated experiments. With those separated measurements, customers would have to further summarize various evaluation results by themselves. In particular cases, an evaluation requirement may comprise diverse and even conflicting measurement criteria. As a result, it could become hard for customers to balance tradeoffs in employing Cloud services. Considering that delivering a single score is also a usual benchmarking strategy to facilitate drawing simple conclusions from evaluation results (Islam et al., 2012), we suggest using boosting metrics to depict summary measurements of Cloud services. In other words, boosting metrics are supposed to measure one complex Cloud service feature involving multiple service properties, or even
measure the overall service features, so as to help customers realize the overall quality of a Cloud service. For example, we may calculate different means or draw radar plot to integrate a suite of benchmarking results into a summary measurement index. Moreover, we can use the idea of boosting metrics to unify and organize a set of different and sophisticated approaches to Cloud service measurement, such as SSP, penalty model, customer satisfaction, and AHP-based decision. Benefiting from the unification, practitioners could conveniently understand, locate, or develop proper boosting metrics for Cloud services evaluation. More importantly, the usage of boosting metrics can further facilitate applying experimental design and analysis to the evaluation work, as demonstrated in Section 5.

Our future work will be unfolded along two directions. Firstly, we plan to gradually collect, propose, and report new boosting metrics to supplement primary measures of individual Cloud service features. Secondly, we will concentrate on the Elasticity of Cloud services, and help improve the current approach (Islam et al., 2012) to Elasticity evaluation by employing more suitable boosting metrics.

7. ACKNOWLEDGMENT

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. NICTA is also funded and supported by the Australian Capital Territory, the New South Wales, Queensland and Victorian Governments, the Australian National University, the University of New South Wales, the University of Melbourne, the University of Queensland, the University of Sydney, Griffith University, Queensland University of Technology, Monash University and other university partners.

8. REFERENCES


Authors

Zheng Li received his Degree of M.E. by Research from the University of New South Wales (UNSW). He is now a PhD student at the School of Computer Science at the Australian National University (ANU), and a graduate researcher with the Software Systems Research Group (SSRG) at National ICT Australia (NICTA). He is the author of more than 15 journal and conference publications. His research interests include empirical software engineering, software cost/effort estimation, machine learning, Web service composition, and Cloud computing.

Liam O’Brien has over 23 years experience in research and development in software engineering. He is a Software and Applications Architect with Geoscience Australia and was previously Chief Software Architect with CSIRO and a Principal Researcher at NICTA’s e-Government Initiative. He is also a Member-at-Large of the Service Science Society Australia which he co-founded in 2010. He has previously worked as a researcher with Lero (Ireland), Carnegie Mellon University’s Software Engineering Institute (USA), CSIRO (Australia) and the University of Limerick (Ireland). His main areas of research include enterprise architecture, software architecture, SOA, service science, software reuse, software modernisation, and cloud computing. He holds a BSc and PhD from the University of Limerick, Ireland. He is a member of the IEEE and IEEE Computer Society.

He Zhang is a professor at State Key Laboratory of Novel Software Technology at Software Institute at Nanjing University. Before joining Nanjing University, he was a Research Scientist with National ICT Australia (NICTA), and a Conjoint Lecturer at University of New South Wales (UNSW). He received his Ph.D from UNSW. He joined academia after 7 years in industry, developing software systems in the areas of aerospace and complex data management. He has published 70+ peer-reviewed research papers in international journals, conferences, and workshops. His current research areas include software process modeling and simulation, process assessment and improvement, software engineering for embedded systems, empirical software engineering, and service-oriented computing.
A SLA VIOLATION DEGREE-AWARE CLOUD SERVICE EVALUATION APPROACH BASED ON HISTORICAL RECORDS

Lianyong Qi\textsuperscript{1,2,3,*}, Wanchun Dou\textsuperscript{1,2}, Xuyun Zhang\textsuperscript{4}, Yuming Zhou\textsuperscript{1,2}

1 Department of Computer Science and Technology, Nanjing University, China
2 State Key Laboratory for Novel Software Technology, Nanjing University, China
3 School of Information Science and Engineering, Qufu Normal University, China
4 Machine Learning Research Group, NICTA, Melbourne, Australia
{lianyongqi@gmail.com}

Abstract
Cloud computing has provided a flexible and cost-effective resource provision manner to the public. From the perspective of cloud users, it is necessary to evaluate the cloud service’s quality based on the service’s historical execution records. However, the traditional evaluation approaches often assume that cloud service’s historical quality is a fixed quality value (i.e., quality point), while neglects the long-term running characteristic of cloud service and the unpredictability of network environment, as well as the resulted fluctuant service quality (i.e., quality curve); this may decrease the completeness of cloud service evaluation. Besides, the traditional SLA (Service Level Agreement) contract manner only considers whether the agreed SLA is violated, while neglects the SLA violation degree, which may be unfair to cloud service providers. In views of the above two challenges, a SLA violation degree-aware cloud service evaluation approach \textit{Vio\textunderscore degree\textunderscore evo} is put forward in this paper. \textit{Vio\textunderscore degree\textunderscore evo} not only considers two kinds of historical records (i.e., quality point and quality curve), but also includes SLA violation degree in cloud service evaluation, so as to make the evaluation more reasonable. Finally, through a set of experiments, we validate the feasibility of our proposal.

Keywords: cloud service, evaluation, historical record, SLA violation degree, quality curve

1. INTRODUCTION

Different from the traditional computing patterns, cloud computing technology has provided the public a completely new computing resources provision manner. With the increasing number of cloud services published by cloud providers (e.g., \textit{IBM}, \textit{Amazon}, \textit{Microsoft}), users can easily deploy their various business applications, in a flexible and cost-effective way\textsuperscript{[1]}. Today, many cloud providers have published their respective cloud services with same or similar functionality. For example, \textit{Amazon S3}, \textit{Google Drive} and \textit{Windows Azure} all provide flexible data storage service. In this situation, from the perspective of cloud users, it is necessary to evaluate the quality of functional-equivalent cloud services so that a quality-optimal candidate could be selected. However, due to the fake quality propagation and unstable network environment, the service quality advertised by cloud service providers is not always trustworthy\textsuperscript{[2]}. Therefore, it becomes a necessity to evaluate the quality of a cloud service based on the service’s historical execution records. Many researchers have studied this hot academic problem and made their contributions. However, there are still some challenges in the present research work.

(1) The traditional cloud service evaluation approaches often assume that cloud service’s historical quality is a fixed quality value (i.e., quality point). This assumption does not always hold in actual cloud service execution, because a cloud service may continuously serve a cloud user for a long period (e.g., one year)\textsuperscript{[3]}. While during this long period, cloud service’s running performance may fluctuate with time, because the network environment cannot always stay stable or be predicted accurately. In this situation, cloud service’s historical quality is not a fixed value (i.e., quality point), but fluctuates with time (i.e., quality curve). Unfortunately, present research work seldom considers the different forms (i.e., quality point and quality curve) of cloud service quality as well as their integration problem.

(2) Due to the unstable cloud service quality, the SLA (Service Level Agreement) contracted between cloud user and cloud provider is vulnerable to network fluctuation. For example, the response-time SLA of Chinese train-ticket-order service (www.12306.cn, whose some computing resources are rented from cloud platform \textit{Aliyun})\textsuperscript{[4]} can be easily violated by a sharp increment of ticket order quantity, especially when the Chinese festival is approaching. In this situation, cloud service (e.g., \textit{Aliyun}) may violate the contracted SLA and should compensate the cloud user. However, present SLA contract manner often considers the SLA violation result (i.e., whether SLA is violated) only, while neglects the SLA violation degree, which may be unfair to the cloud providers.
In view of the above two challenges, a cloud service evaluation approach named \( \text{Vio\_degree\_eva} \) (SLA violation degree-aware cloud service evaluation) is put forward in this paper. \( \text{Vio\_degree\_eva} \) not only considers two kinds of historical records (i.e., quality point and quality curve) of cloud services, but also introduces SLA violation degree into SLA contract so as to make the contract fairer to cloud providers.

The remainder of this paper is organized as follows. In Section 2, we discuss the different forms of cloud service quality and introduce a novel concept of flexible double-range SLA, based on which the motivation of this paper is demonstrated. A SLA violation degree-aware cloud service evaluation approach, i.e., \( \text{Vio\_degree\_eva} \) is put forward in Section 3, which considers different forms of service quality and SLA violation degree simultaneously. In Section 4, a set of experiments are deployed to validate the feasibility of our proposal, in terms of effectiveness and efficiency. Our \( \text{Vio\_degree\_eva} \) approach is evaluated in Section 5; and finally, in Section 6, we summarize the paper and point out our future research directions.

2. FORMAL DEFINITIONS AND MOTIVATION

In this section, we first classify the quality forms of cloud services into two categories. Afterwards, a novel concept of double-range SLA is introduced. Finally, with cloud service’s different quality forms and different SLA forms, we demonstrate the motivation of our paper.

2.1 QUALITY FORMS OF CLOUD SERVICES.

Concretely, the quality forms of cloud services could be classified into the following two categories: quality point (Def.1) and quality curve (Def.2). To ease the following discussions, \( CS \) denotes a cloud service; \( HR \) denotes a historical record of \( CS \); \( [0, T] \) denotes the agreed service-period of \( CS \). Besides, for simplicity, we only consider a negative quality dimension \( q \) here (positive quality dimension can be converted into negative one by multiplying -1).

**Definition 1. Quality point.** For historical record \( HR \) of cloud service \( CS \), its quality over dimension \( q \) (denoted by \( HR(q) \)) is a quality point, iff equation in (1) holds.

\[
HR(q) = a \quad (1)
\]

Here, \( a \) is a fixed value calculated after comprehensive consideration of \( CS \)'s running quality during \([0, T]\). For example, after statistics, \( HR(\text{failure\_rate}) = 5\% \) during the service-period \([0 \text{ hour}, 100 \text{ hours}]\). Here, we utilize \( \text{General}_{\text{SLA}} \) to denote the traditional single-range SLA (e.g., \( \text{failure\_rate} \in [0, 10\%] \)) over quality dimension \( q \). Then a conclusion could be drawn that SLA is satisfied if \( HR(q) \in \text{General}_{\text{SLA}} \) holds; otherwise, SLA is violated.

**Definition 2. Quality curve.** For historical record \( HR \) of cloud service \( CS \), its quality over dimension \( q \) (denoted by \( HR(q) \)) is a quality curve, iff equation in (2) holds.

\[
HR(q) = f(t) \quad (0 \leq t \leq T) \quad (2)
\]

Here, \( HR(q) \) is not a fixed value, but changes with service time \( t \). Next, an example is provided in Figure 1 to illustrate the quality curve of latency of cloud service \( CS \), during its service-period \([0 \text{ hour}, 100 \text{ hours}]\). As Figure 1 shows, the latency of cloud service \( CS \) is not a fixed value, but fluctuant with running time \( t \).

In this situation, we cannot determine the concrete latency value of \( CS \). Besides, the fluctuant service quality brings another challenge, i.e., the traditional SLA contract specified by a single range (e.g., \( \text{latency} \in [0 \text{ms}, 200 \text{ms}] \)) is not suitable to constrain the fluctuant cloud service quality very well, because the single-range SLA could be easily violated by a sudden fluctuation of cloud service quality. For example, considering the example in Figure 1, the single-range latency SLA, i.e., \([0 \text{ms}, 200 \text{ms}]\) is violated by a sudden but normal fluctuation at peak point \( P \).

In this situation, we cannot simply conclude that the latency SLA of cloud service \( CS \) is violated, due to the following two reasons. First, as Figure 1 shows, in most of the service-period \([0 \text{ hour}, 100 \text{ hours}]\), cloud service \( CS \) performs very well in latency and does not violate the SLA constraint (e.g., \( \text{latency} \in [0 \text{ms}, 200 \text{ms}] \)). Second, cloud service’s running quality is fluctuant in nature and cannot be predicted accurately before its execution. Therefore, the traditional single-range SLA cannot accommodate the fluctuant quality of cloud service very well. In view of the above considerations, in this paper, we try to improve the traditional single-range SLA and introduce a novel concept of double-range SLA to accommodate the fluctuant service quality curve.

2.2 DOUBLE-RANGE SLA

**Definition 3. Double-range SLA.** A double-range SLA of cloud service, i.e., Double\_SLA could be formalized as a two-tuple in (3). Here, \( \text{General}_{\text{SLA}} \) denotes the traditional single-range SLA that depicts cloud user’s general quality expectation, while \( \text{Peak}_{\text{SLA}} \) is another quality constraint that limits the fluctuant quality of cloud services at the peak point (e.g., point \( P \) in Figure 1).

\[
\text{Double\_SLA} = (\text{General}_{\text{SLA}}, \text{Peak}_{\text{SLA}}) \quad (3)
\]
The example in Figure 2 illustrates the meanings of double-range SLA. In Figure 2, latency ∈ [0ms, 200ms] is the GeneralSLA promised by cloud service CS (To ease the subsequent discussions, we denote GeneralSLA with its upper bound only, i.e., 200ms), while latency ∈ [0ms, 300ms] is the PeakSLA that cannot be violated by the peak point of CS's running quality (Likewise, we denote PeakSLA with its upper bound only, i.e., 300ms).

Next, by comparing cloud service's fluctuant service quality (i.e., HR(q) = f(t) in (2)) and double-range SLA (i.e., Double_SLAN = (GeneralSLAN, PeakSLAN) in (3)), we can measure a cloud user's satisfaction degree with cloud service CS's historical quality over dimension q. Concretely, as Figure 2 shows, when t ∈ [0, t1] ∨ [t2, T], the farther f(t) is away from GeneralSLA, the more satisfaction a user gets; while when t ∈ [t1, t2], the farther f(t) is away from GeneralSLA, the more dissatisfaction a user gets. Therefore, we can calculate user satisfaction degree with dimension q's service quality in historical record HR of cloud service CS, i.e., Sat_degree (CS, HR, q), with the area covered by HR(q) = f(t), GeneralSLA, t = 0 and t = T. More formally, Sat_degree (CS, HR, q) could be calculated by (4), where HR(q) = f(t) (t ∈ [0, T]) holds.

\[
\text{Sat\_degree (CS, HR, q) = } \int_{0}^{T} (\text{GeneralSLA} - f(t)) \, dt \quad (4)
\]

Based on the above analyses, we can determine whether a double-range SLA could be satisfied by the fluctuant quality of a cloud service, with the following definition (here, only a negative quality dimension q, e.g., latency is considered for illustration purpose).

**Definition 4.** Fluctuant service quality satisfies double-range SLA. For cloud service CS, its fluctuant quality over dimension q, i.e., \( HR(q) = f(t) \ (t \in [0, T]) \) satisfies the promised double-range SLA, i.e., Double_SLAN = (GeneralSLAN, PeakSLAN), iff the constraints in (5) and (6) are met simultaneously. Otherwise, the promised double-range SLA is violated.

\[
\begin{align*}
\text{Max} \ f(t) \leq \text{PeakSLA}, \ (t \in [0, T]) & \quad (5) \\
\int_{0}^{T} (\text{GeneralSLA} - f(t)) \, dt \geq 0 & \quad (6)
\end{align*}
\]

Next, we explain the physical meanings of formulas (5) and (6), by the example in Figure 2. Concretely, the constraint in (5) ensures that the maximum of fluctuant service quality does not exceed the promised peak threshold PeakSLAN; while the constraint in (6) ensures that the cloud user's general satisfaction degree with cloud service CS's fluctuant quality is always positive (or zero), even if CS's fluctuant quality may violate the promised GeneralSLA sometimes (e.g., when \( t \in [t_1, t_2] \) in Figure 2).

### 2.3 Motivation

In subsection 2.1-2.2, we have introduced two forms of historical record (i.e., quality point, quality curve) and two forms of SLA (i.e., single-range SLA, double-range SLA). Next, with the above concepts, we illustrate the motivation of this paper in Figure 3. As Figure 3 shows, a cloud user wants to evaluate the quality of cloud service CS, based on CS's L historical records, i.e., \( HR_1, HR_2, \ldots, HR_L \). Here, each record \( HR_i (1 \leq i \leq L) \) consists of m (negative) quality dimensions, i.e., \( q_1, q_2, \ldots, q_m \), and records their quality values \( HR_q (q_j) (1 \leq j \leq m) \) in the form of quality point or quality curve. Besides, \( HR_i \) also records the SLA corresponding to each \( HR_q (q_j) \) (concretely, a single-range SLA, i.e., GeneralSLAN is available for each quality point, while a double-range SLA, i.e., Double_SLAN = (GeneralSLAN, PeakSLAN) is available for each quality curve).

In this situation, it is a challenge to evaluate the quality of cloud service CS, based on CS's L historical records (with different quality forms and different SLA forms). Besides, it is another challenge to consider the SLA violation degree in service evaluation. In view of these challenges, a novel cloud service evaluation approach named Vio_degree_eva is put forward in the next section.

![Cloud Service Evaluation Approach](image-url)
3. A SLA VIOLATION DEGREE-AWARE CLOUD SERVICE EVALUATION APPROACH

In this section, a novel cloud service evaluation approach, i.e., \textit{Vio\_degree\_eva} is introduced, which is based on cloud service \textit{CS}'s \textit{L} historical records with different quality forms and different SLA forms. The main idea of our proposed \textit{Vio\_degree\_eva} is: \textbf{first}, we evaluate cloud user's satisfaction degree with quality points; \textbf{second}, we evaluate cloud user's satisfaction degree with quality curves; \textbf{third}, satisfaction degree integration. Concretely, the three steps of our proposed \textit{Vio\_degree\_eva} approach are listed in Figure 4. To ease the following discussions, in Step1 and Step2, we only consider a negative quality dimension \textit{q} and a historical record \textit{HR} of cloud service \textit{CS}; while in Step3, all the \textit{m} quality dimensions \textit{q1}...\textit{qm} and \textit{L} historical records \textit{HR1}...\textit{HR_L} of \textit{CS} and are considered.

\begin{itemize}
  \item \textbf{Step1}: Quality point evaluation. According to cloud service \textit{CS}'s fixed quality \textit{HR(q)} = \textit{a} and contracted single-range SLA \textit{GeneralSLA}, calculate cloud user's satisfaction degree with quality point, i.e., \textit{Sat\_degree} (\textit{CS}, \textit{HR}, \textit{q}).
  \item \textbf{Step2}: Quality curve evaluation. According to cloud service \textit{CS}'s fluctuant quality \textit{HR(q)} = \textit{f(t)} (\textit{t} \in [0, \textit{T}]) and contracted double-range SLA \textit{Double\_SLA}, calculate cloud user's satisfaction degree with quality curve, i.e., \textit{Sat\_degree} (\textit{CS}, \textit{HR}, \textit{q}).
  \item \textbf{Step3}: Satisfaction degree integration. By integrating the user satisfaction degrees \textit{Sat\_degree} (\textit{CS}, \textit{HR}, \textit{q}_j) (1 \leq \textit{i} \leq \textit{L}, 1 \leq \textit{j} \leq \textit{m}) obtained in Step1 and Step2, we evaluate the quality of cloud service \textit{CS}.
\end{itemize}

![Figure 4. Three Steps of Cloud Service Evaluation Approach Vio\_degree\_eva](image)

3.1 Step1: Quality point evaluation.

In this step, we calculate cloud user's satisfaction degree with dimension \textit{q}'s fixed quality in the historical record \textit{HR} of cloud service \textit{CS}, i.e., \textit{Sat\_degree} (\textit{CS}, \textit{HR}, \textit{q}), based on \textit{CS}'s fixed quality \textit{HR(q)} = \textit{a} (i.e., quality point) and contracted single-range SLA constraint \textit{GeneralSLA}. Next, we introduce the concrete calculation process by considering the example in Figure 5.

\begin{itemize}
  \item In Figure 5, the relationship between cloud service \textit{CS}'s quality point and corresponding SLA constraint \textit{GeneralSLA} (here, only a negative quality dimension \textit{q} is discussed for simplicity) is divided into three categories:
    \begin{itemize}
      \item (a) service quality violates \textit{GeneralSLA} (see Figure 5(a))
      \item (b) service quality satisfies \textit{GeneralSLA} (see Figure 5(b))
      \item (c) service quality just equals \textit{GeneralSLA} (see Figure 5(c))
    \end{itemize}
\end{itemize}

![Figure 5. Three Relationships of Quality Point and GeneralSLA](image)

In Figure 5(a), cloud service \textit{CS}'s quality corresponding to historical record \textit{HR} violates the contracted SLA; therefore, cloud user's satisfaction degree is low. Here, we utilize satisfaction degree range \([0, 1]\) to depict this kind of SLA violation; and the farther a quality point is away from \textit{GeneralSLA} upper bound, the smaller satisfaction degree a cloud user will get. While in Figure 5(b), cloud service \textit{CS}'s quality corresponding to historical record \textit{HR} satisfies the contracted SLA; therefore, cloud user's satisfaction degree is high. Here, we utilize satisfaction degree range \([0, 1]\) to depict this kind of SLA satisfaction; and the farther a quality point is away from \textit{GeneralSLA} upper bound, the larger satisfaction degree a cloud user will get. Finally in Figure 5(c), cloud service \textit{CS}'s quality corresponding to historical record \textit{HR} just equals the contracted SLA; in this situation, we set the user satisfaction degree 0.

Besides, we argue that cloud user's satisfaction degree with service quality also obeys the “Marginal Effect”\(^{5}\) in Social Psychology domain. Let's consider the example in Figure 6. In Figure 6(a), quality points A and B violate the contracted SLA, and they are both far away from the \textit{GeneralSLA} upper bound (however, A is close to B); in this situation, cloud user's satisfaction degrees with A and B should be close too (actually, both near -1). Similarly, In Figure 6(b), quality points C and D satisfy the contracted SLA, and they are both far away from the \textit{GeneralSLA} upper bound (however, C is close to D); in this situation, cloud user's satisfaction degrees with C and D should be close too (actually, both near 1). Therefore, “Marginal Effect” should be considered in user satisfaction degree model.
3.2 Step2: Quality curve evaluation.

In this step, we calculate cloud user’s satisfaction degree with dimension \( q \)’s fluctuant quality in the historical record \( HR \) of cloud service \( CS \), i.e., \( Sat\_degree\ (CS, HR, q) \), based on \( CS \)’s fluctuant quality \( HR(q) = f(t) \ (t \in [0, T]) \) (i.e., quality curve) and contracted double-range SLA constraint Double\_SLA = (General\_SLA, Peak\_SLA). Next, we introduce the concrete calculation process by considering the example in Figure 2.

As introduced in Figure 2, we can calculate user satisfaction degree with dimension \( q \)’s fluctuant service quality in the historical record \( HR \) of cloud service \( CS \), i.e., \( Sat\_degree\ (CS, HR, q) \), with the area covered by \( HR(q) = f(t) \), General\_SLA, \( t = 0 \) and \( t = T \). Concretely, user satisfaction degree \( Sat\_degree\ (CS, HR, q) \) could be calculated by formula (4); however, according to (4), \( Sat\_degree\ (CS, HR, q) \) \( \in (-\infty, +\infty) \) holds, which cannot reflect the meaning of user satisfaction degree very well. Therefore, in this step, we improve formula (4) to make the user satisfaction degree belong to range \([-1, 1]\). Next, we introduce the detailed improvement process.

Concretely, formula (8) is recruited to replace formula (4) to calculate user satisfaction degree \( Sat\_degree\ (CS, HR, q) \). Here, if \( \text{Max}(f(t)) > \text{Peak\_SLA} \ (t \in [0, T]) \), i.e., Peak\_SLA is violated, then \( Sat\_degree\ (CS, HR, q) = -1 \) and no further calculation is needed. Otherwise, Peak\_SLA is satisfied; in this situation, we further calculate \( Sat\_degree\ (CS, HR, q) \) by the second and third equations in formula (8). Concretely, if \( \int_{0}^{t}(\text{General\_SLA}\cdot f(t))\ dt < 0 \) holds, then user satisfaction degree \( Sat\_degree\ (CS, HR, q) \) is equal to the ratio between \( \int_{0}^{t}(\text{General\_SLA}\cdot f(t))\ dt \) and \( \int_{0}^{T}f(t)\ dt \), which belongs to range \((-1, 0)\); else if \( \int_{0}^{T}(\text{General\_SLA}\cdot f(t))\ dt \geq 0 \) holds, then \( Sat\_degree\ (CS, HR, q) \) is equal to the ratio between \( \int_{0}^{t}(\text{General\_SLA}\cdot f(t))\ dt \) and \( \int_{0}^{T}\text{General\_SLA}\ dt \), which belongs to range \([0, 1]\). Therefore, through formula (8), we can calculate user satisfaction degree with fluctuant quality curve, i.e., \( Sat\_degree\ (CS, HR, q) (\in [-1, 1]) \).

\[
Sat\_degree\ (CS, HR, q) = \begin{cases} 
-1, & \text{if } \text{Max}(f(t)) > \text{Peak\_SLA} \\
\frac{\int_{0}^{t}(\text{General\_SLA}\cdot f(t))\ dt}{\int_{0}^{T}f(t)\ dt}, & \text{if } \int_{0}^{T}(\text{General\_SLA}\cdot f(t))\ dt < 0 \\
\frac{\int_{0}^{T}(\text{General\_SLA}\cdot f(t))\ dt}{\int_{0}^{T}\text{General\_SLA}\ dt}, & \text{if } \int_{0}^{T}(\text{General\_SLA}\cdot f(t))\ dt \geq 0 
\end{cases}
\]
3.3 Step3: Satisfaction degree integration.

In Step1–Step2, we have obtained user satisfaction degree with a dimension $q$’s quality (quality point or quality curve) corresponding to historical record $HR$ of cloud service $CS$, i.e., $Sat\ degree\ (CS,\ HR,\ q)$. However, a cloud service often has multiple historical records and each record contains multiple quality values over different quality dimensions. Here, we assume that cloud service $CS$ owns $L$ historical records $HR_1,…HR_L$ and each historical record $HR_i$ ($1 \leq i \leq L$) contains quality information over $m$ dimensions $q_1,…q_m$. Next, in this step, we will introduce how to evaluate cloud service $CS$, by integrating the multiple satisfaction degree $Sat\ degree\ (CS,\ HR,\ q)\ (1 \leq i \leq L,\ 1 \leq j \leq m)$ derived in Step1 and Step2.

The integration process is based on two kinds of weight, i.e., weight $w_i$ of historical record $HR_i$ ($1 \leq i \leq L$, $\sum w_i=1$) and weight $\omega_j$ of quality dimension $q_j$ ($1 \leq j \leq m$, $\sum \omega_j=1$) (here, we assume that $w_i$ and $\omega_j$ are known already, because the discussions of weight design are out of the scope of this paper). Concretely, the quality of cloud service $CS$ could be evaluated by formula (9), where $Quality(CS)$ denotes the evaluated quality of $CS$.

$$Quality(CS) = \frac{1}{L} \sum_{i=1}^{L} w_i \left( \frac{1}{m} \sum_{j=1}^{m} \omega_j \cdot Sat\_degree\ (CS,\ HR_i,\ q_j) \right)$$ \hspace{1cm} (9)

According to (9), the evaluated quality of cloud service $CS$, i.e., $Quality(CS) \in [-1, 1]$ holds. Here, we evaluate the quality of cloud service $CS$ by considering both $CS$’s historical quality and $CS$’s agreed SLA, because it does not make any sense to consider $CS$’s historical quality only without comparing its corresponding SLA constraints. Therefore, we argue that formula (9) is more reasonable for cloud service evaluation.

With the above three steps of $Vio\_degree\_eva$, we can evaluate the quality of a cloud service, based on its multiple forms of historical records (i.e., quality point and quality curve) and multiple SLA forms (i.e., single-range SLA and double-rang SLA). More formally, the pseudo code of our proposal is specified as below.

4. Experiments

In this section, a set of experiments are designed and deployed to validate the feasibility of our proposed cloud service evaluation approach $Vio\_degree\_eva$.

4.1 The Dataset and Experiment Deployment

In our proposed $Vio\_degree\_eva$ approach, both quality point (corresponding to Single-rang SLA, i.e., $General_{SLA}$) and quality curve (corresponding to Double-rang SLA, i.e., $General_{SLA}$, $Peak_{SLA}$) are considered. However, to the best of our knowledge, there is no present dataset that contains these two kinds of quality data. Fortunately, we can construct the experiment dataset with advanced software tools.

### Algorithm: $Vio\_degree\_eva\ (CS,\ HR,\ Q)$

**Input:** $CS$: a cloud service 
$HR=(HR_1,…HR_L)$: a set of historical records of $CS$ 
$Q=(q_1,…,q_m)$: a set of quality dimensions of $CS$

**Output:** $Quality(CS)$: evaluated quality of cloud service $CS$

```
1: Quality(CS) = 0
2: for i = 1 to L do
3:     for j = 1 to m do
4:         if HR(q_j) is a quality point then
5:             calculate $Sat\_degree\ (CS,\ HR_i,\ q_j)$ by (7)
6:         else if HR(q_j) is a quality curve then
7:             calculate $Sat\_degree\ (CS,\ HR_i,\ q_j)$ by (8)
8:         end if
9:     end for
10: end for
11: Quality(CS) = Quality(CS) + $Sat\_degree\ (CS,\ HR_i,\ q_j)$

13: Quality(CS) = Quality(CS) / $(L \cdot m)$
14: return Quality(CS)
```

Concretely, the quality curve set of cloud services, i.e., $Dataset\_quality\_curve$, could be generated with the help of performance-monitoring software CloudWatch [6]. CloudWatch is developed by Amazon and could provide monitoring service over a series of quality dimensions of cloud services. Corresponding to $Dataset\_quality\_curve$, the double-range SLA, i.e., $(General_{SLA}, Peak_{SLA})$ could be generated randomly by considering the range of fluctuant quality (as only negative quality dimensions are considered, $General_{SLA}$ should be smaller than $Peak_{SLA}$ in the constructed double-range SLA). Besides, the quality point set of cloud services, i.e., $Dataset\_quality\_point$, could be generated from WS-DREAM [7] published by Dr. Zibin Zheng in 2011. WS-DREAM consists of 4532 services from public sources on the web, and 142 distributed computers from Planet-Lab are employed for evaluating the real service quality performance in 64 time intervals. Corresponding to $Dataset\_quality\_point$, the single-range SLA, i.e., $General_{SLA}$ could be generated randomly by considering the range of quality point data. Besides, we assume weight $w_i = 1/L$ ($1 \leq i \leq L$) and $\omega_j = 1/m$ ($1 \leq j \leq m$). For a cloud user, it is difficult to accurately evaluate his/her satisfaction degree with a quality curve by (8), as the service quality fluctuates continuously; therefore, sampling technique is recruited here for quality curve, so as to achieve approximate evaluation results. Some symbols recruited in the experiments as well as their specifications could be found in Table 1.
The experiments were conducted on a HP pc with I7-6700 CPU (4 cores, 4GHz processors) and 8.0 GB RAM. The machine is running under Windows XP and JAVA 1.5. Each experiment was carried out 10 times and the average results were adopted.

### 4.2 Experiment Results

In this subsection, we compare our evaluation approach \( \text{Vio\_degree\_eva} \) with another two evaluation approaches, i.e., \( \text{SD-HCF} \) \(^8\) (only quality point is considered) and \( \text{FL-FL} \)\(^9\)(only quality curve is considered). Concretely, five evaluation profiles were tested. The parameter values recruited in experiments are listed in Table 1.

#### Profile 1: Utilization rate of historical quality records

Due to the sparse cloud service quality feedback, we should make full use of all the collected historical quality records. However, for different cloud service evaluation approaches (i.e., \( \text{SD-HCF}, \text{FL-FL}, \text{Vio\_degree\_eva} \)), their utilization rates of historical quality records are not the same. In this profile, we design a set of experiments to compare their utilization rates. Concretely, let \( L=200 \) and \( m=15 \), then there are totally \( 200*15=3000 \) historical quality records. We assume \( \text{RATIO}_{\text{point\_curve}} = 1/2 \), which means that there are 1000 quality points and 2000 quality curves. Next, we evaluate the 3000 historical quality records by three approaches respectively and calculate the corresponding user satisfaction degree distribution.

The concrete experiment results are shown in Figure 8, where two observations are available. (1) All the 3000 historical quality records are utilized in our proposed \( \text{Vio\_degree\_eva} \), while only 1000 quality points are considered in \( \text{SD-HCF} \) and only 2000 quality curves are considered in \( \text{FL-FL} \). Namely, both \( \text{SD-HCF} \) and \( \text{FL-FL} \) waste some precious historical execution information of cloud services, so their utilization rates are smaller than \( \text{Vio\_degree\_eva} \). (2) In \( \text{SD-HCF} \), the user satisfaction degrees with most (actually 54%) quality points are equal to 0, which means that a SLA violation occurs and the violation degree is not considered in \( \text{SD-HCF} \). Likewise, in \( \text{FL-FL} \), the user satisfaction degrees with most (actually 62%) quality curves are equal to 0, which means that a SLA violation occurs and the violation degree is omitted. Besides, in \( \text{FL-FL} \), there is no quality curve whose satisfaction degree is in range \((0, 0.5)\) (i.e., two red cylinders are absent from Figure 8), which seems unreasonable to some extent. While in our proposed \( \text{Vio\_degree\_eva} \), the user satisfaction degree is relatively “average” (i.e., utilization rate of historical quality records is high), and negative satisfaction degree is presented to indicate both SLA violation result and SLA violation degree.

#### Profile 2: Time cost with respect to \( L \)

In this profile, we compare the time cost of three approaches with respect to the number of historical records of cloud service \( CS \), i.e., \( L \). Concretely, the number of quality dimensions, i.e., \( m = 15 \) holds; \( \text{RATIO}_{\text{point\_curve}} = 1 \) holds in \( \text{Vio\_degree\_eva} \); for each quality curve, the sampling point number \( K = 10 \) holds in both \( \text{FL-FL} \) and \( \text{Vio\_degree\_eva} \); \( L \) is varied from 200 to 1000.

The experiment results are shown in Figure 9. It can be seen from Figure 9 that the time costs of three approaches all grow approximately linearly with the rise of \( L \). Furthermore, \( \text{SD-HCF} \) outperforms the other two approaches in time cost because it only considers the simple quality points without sampling cost; in contrast, \( \text{FL-FL} \) does not perform very well in time cost, since sampling is
needed for each quality curve considered in FL-FL. Also, as Figure 9 shows, our proposed Vio_degree_eva realizes a medium-sized time cost between SD-HCF and FL-FL, as both quality points and quality curves are considered in Vio_degree_eva.

Profile 3: Time cost with respect to \(m\)

In this profile, we compare the time cost of three approaches with respect to the number of quality dimensions of cloud service \(CS\), i.e., \(m\). Concretely, the number of historical records, i.e., \(L = 1000\) holds; \(RATIO_{point/curve} = 1\) holds in Vio_degree_eva; for each quality curve, the sampling point number \(K = 10\) holds in both FL-FL and Vio_degree_eva; \(m\) is varied from 3 to 15.

The experiment results are shown in Figure 10. It can be seen from Figure 10 that the time costs of three approaches all grow approximately linearly with the rise of \(m\), because all the \(m\) quality dimensions need to be considered in the three approaches. Furthermore, similar with Figure 9, SD-HCF outperforms the other two approaches in time cost because it only considers the quality points without sampling cost; while FL-FL and Vio_degree_eva achieve similar but poor performance in time cost, as sampling is needed for each quality curve considered in these two approaches.

Profile 4: Time cost with respect to \(K\)

In this profile, we compare the time cost of three approaches with respect to the number of sampling points, i.e., \(K\). Concretely, the number of historical records, i.e., \(L = 1000\) holds; \(RATIO_{point/curve} = 1\) holds in Vio_degree_eva; the sampling point number \(K\) is varied from 10 to 50 in both FL-FL and Vio_degree_eva.

The experiment results are shown in Figure 11. It can be seen from Figure 11 that the time cost of SD-HCF is small and stays relatively stable, this is because only quality points (without sampling, so we can regard \(K = 1\) holds constantly) are considered in SD-HCF. While the time costs of FL-FL and Vio_degree_eva both increase approximately linearly with the growth of \(K\), this is because all the \(K\) sampling points should be considered in these two approaches. However, as Figure 11 shows, our proposed Vio_degree_eva performs better than FL-FL, because only partial historical quality records (i.e., quality curve) in Vio_degree_eva need sampling.

Profile 5: User satisfaction degree with respect to \(K\)

In both FL-FL and Vio_degree_eva, sampling technique is recruited to simplify the calculation of user satisfaction degree with quality curve (see formula (8)). Concretely, each continuous quality curve is divided into \(K\) discrete quality points. Next, we test the correlation between \(K\) and derived user satisfaction degree (by FL-FL and Vio_degree_eva). Concretely, we consider a quality curve \(HR(q)\) corresponding to quality dimension \(q\) and historical record \(HR\) of cloud service \(CS\). Then \(HR(q)\) is divided into \(K\) (\(K = 10, 20, 30, 40, 50\)) discrete quality points, through which the user satisfaction degree with \(HR(q)\) could be calculated by (8). We compare the user satisfaction degree and the comparison results are shown in Figure 12.

Figure 10. Comparison of Time Cost w.r.t. \(m\)

Figure 11. Comparison of Time Cost w.r.t. \(K\)

Figure 12. User Satisfaction Degree w.r.t. \(K\)
In Figure 12, three observations are available. (1) For both FL-FL and Vio_degree_eva, user satisfaction degrees fluctuate slightly with the rise of $K$ in most cases (i.e., when $K$ is varied from 20 to 50). And user satisfaction degree stays approximately stable when $K$ is large (i.e., $K = 40, 50$), this is because when $K$ is large enough, the $K$ sampled quality points can approach the original continuous quality curve $HR(q)$ infinitely. (2) When $K$ is small (i.e., $K = 10$), a distortion of satisfaction degree is observed; this is because when $K$ is small, the $K$ sampled quality points cannot represent the original continuous quality curve $HR(q)$ very well. (3) User satisfaction degree calculated by FL-FL is larger than that by Vio_degree_eva, this is because an additional smoothing operation (i.e., $Sat_degree = (Sat_degree + 1) / 2$) is adopted in FL-FL.

5. Evaluation

In this section, we first analyze the time complexity of Vio_degree_eva introduced in Section 3, to evaluate the efficiency of our proposal. Afterwards, a comparison with related work is presented, which is followed by discussions regarding the paper limitations and our future work.

5.1 Time Complexity Analysis

Suppose that cloud service CS owns $L$ historical records, each record corresponds to $m$ quality dimensions, and the sampling point number is $K$ for each quality curve. Then there are $L^m$ quality points or quality curves totally. Next, we discuss the time complexity of Vio_degree_eva by considering the following two extreme situations.

(1) There are $L^m$ quality points.

If $HR(q)$ is a quality point, then according to Step1, user satisfaction degree $Sat_degree (CS, HR, q)$ could be calculated by formula (7), whose time complexity is $O(1)$. As there are totally $L^m$ quality points, the time complexity of satisfaction degree integration in Step3 is $O(L^m)$. With the above analyses, time complexity of Vio_degree_eva is $O(L^m)$. With the above analyses, time complexity of Vio_degree_eva is $O(L^m)$.

(2) There are $L^m$ quality curves.

If $HR(q)$ is a quality curve, then according to Step2, user satisfaction degree $Sat_degree (CS, HR, q)$ could be calculated by formula (8), whose time complexity is $O(K)$. As there are totally $L^m$ quality curves, the time complexity of satisfaction degree integration in Step3 is $O(L^m)$. With the above analyses, time complexity of Vio_degree_eva is $O(L^m + L^mK)$.

With the above analyses, a conclusion could be drawn that the time complexity of Vio_degree_eva is $O(L^m + K)$.

5.2 Related Works and Comparison Analyses

The dynamic and unstable execution environment of cloud services makes it a great challenge to accurately evaluate the service quality. A monitoring mechanism is provided in [10] to dynamically collect the running service quality level and compare it with the contracted SLA between users and services, so as to determine whether SLA constraints are violated. However, the monitoring/verification work cannot be enacted and executed by cloud users or cloud providers, because cloud users lose direct control of their business execution in cloud, while cloud providers may be not trustworthy from the perspective of cloud users. In view of this, a third-party cloud service broker (CSB) is proposed in [11] to complete the tasks of service quality monitoring and SLA verification. Similarly, a SLA verification framework is put forward in [12], which leverages a third-party auditor (TPA) to realize the effective detection of SLA violations.

While the above works mainly focus on the SLA monitoring or verification, few of them consider the cloud service evaluation based on the monitored historical quality data. In view of this, works [1][13][14] recruit the historical records of services for service composition, service evaluation and service trust calculation, respectively. Furthermore, in [15], weight is assigned to each historical record for better service evaluation and selection decisions. However, the above works only consider the historical records in the form of quality points, while neglect the fluctuant characteristic of cloud service quality during the long service-period.

In [3], the authors have observed the long-term business relationship between cloud users and cloud service providers, as well as the fluctuant service quality during the long service-period. However, the paper only focuses on the cloud service composition (different from our paper), and does not discuss the flexible SLA as well as the SLA violation degree. In our previous work [9], a flexible SLA concept is put forward to accommodate the fluctuant cloud service quality; and furthermore, a cloud service could be evaluated by comparing its fluctuant service quality and flexible SLA. However, further discussion about SLA violation degree is absent from [9]; besides, [9] only discusses the quality curve of cloud service, without considering the quality point as well as their integration problem.

In view of the above shortcomings, a SLA violation degree-aware cloud service evaluation approach Vio_degree_eva is put forward in this paper. Vio_degree_eva not only considers two kinds of historical records (i.e., quality point and quality curve), but also introduces SLA violation degree in cloud service evaluation. Finally, through a set of experiments, we validate the feasibility of our proposal, in terms of evaluation effectiveness and efficiency.
5.3 Further Discussions

In our proposed \( Vio\_degree\_eva \) approach, we assume that the cloud service’s historical quality records are known already. While in fact, due to privacy concern and poor incentive, it is really a challenging task to collect the historical quality records of a cloud service. Besides, most historical quality data is published by cloud providers, which is not always trustable from the cloud user perspective. Therefore, in the future, we will consider these shortcomings and study a trusted and privacy-aware cloud service evaluation approach.

6. CONCLUSIONS

It is a challenge to accurately evaluate the quality of cloud services, due to the long service-period and unstable quality performance. In view of this challenge, we classify the historical quality records of cloud services into two categories, i.e., quality point and quality curve; furthermore, two forms of SLA (i.e., single-range SLA corresponding to quality point and double-range SLA corresponding to quality curve) are brought forth; finally, we calculate the SLA violation degree and develop a SLA violation degree-aware cloud service evaluation approach \( Vio\_degree\_eva \). Through a set of experiments, we validate the feasibility of \( Vio\_degree\_eva \) in terms of effectiveness and efficiency. In the future, we will discuss the privacy issues in cloud service evaluation, and study a trusted and privacy-aware evaluation approach.

7. ACKNOWLEDGMENT

We would like to thank Dr. Zibin Zheng for the dataset WS-DREAM recruited partially in our experiments. (Partial work of this paper has been accepted by ICWS 2014).

8. REFERENCES


Authors

Lianyong Qi received his PhD degree in Department of Computer Science and Technology from Nanjing University, China, in 2011. In 2010, he visited the Department of Information and Communication Technology, Swinburne University of Technology. Now, he is a lecturer of the School of Information Science and Engineering, Qufu Normal University, China. Also, he is doing his postdoctoral research in the Postdoctoral Research Centre for Software Engineering, Nanjing University, China. He is now chairing a NSFC project and has published nearly 20 research papers in international journals (e.g., TCC, CCPE, JCSS, Computing) and international conferences (e.g., ICWS, HPCC, APSCC). His research interests include big data, cloud computing and services computing.

Wanchun Dou received his PhD degree in Mechanical and Electronic Engineering from Nanjing University of Science and Technology, China, in 2001. From Apr. 2001 to Dec. 2002, he did his postdoctoral research in the Department of Computer Science and Technology, Nanjing University, China. Now, he is a full professor of the State Key Laboratory for Novel Software Technology, Nanjing University, China. From Apr. 2005 to Jun. 2005 and from Nov. 2008 to Feb. 2009, he respectively visited the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, as a
visiting scholar. Up to now, he has chaired three NSFC projects and published more than 60 research papers in international journals (e.g., TPDS, TETC, FGCS, JCSS, JWSR, CCPE, IJHPCA) and international conferences (e.g., ICWS, HPCC, ICPADS). His research interests include workflow, cloud computing and service computing.

Xuyun Zhang is a researcher at the Machine Learning Group, Australia's Information Communications Technology Research Centre of Excellence (NICTA), working with mentors Prof. Chris Leckie and Prof. Ramamohanarao (Rao) Kotagiri from The University of Melbourne. He studied his PhD degree in the Faculty of Engineering and Information Technology (FEIT) at the University of Technology Sydney (UTS), supervised by A/Prof. Jinjun Chen. He also worked at the Australian national scientific body Commonwealth Scientific and Industrial Research Organization (CSIRO) as a co-supervised PhD student, with Principal Research Scientist Surya Nepal. His research interest includes Cloud Computing, Big Data, scalable Machine learning and Data Mining, Large Graph Processing, Privacy and Security, Web Service Composition, QoS.

Yuming Zhou received his PhD degree from Southeast University, China, in 2003. Now, he is a full professor of the State Key Laboratory for Novel Software Technology, Nanjing University, China. From Feb. 2005 to Feb. 2008, he visited Hong Kong Polytechnic University, as a Research Associate and Research Fellow. He has published some research papers in international journals (e.g., TSE, TOSEM, JSS, JCSS, SPE, IST, JCST) and conferences (e.g., ICSM, SCAM, APSEC, SEKE, SAC, Metrics). His research interests include software metrics, software evolution, and program understanding and analysis.
INCIDENT NOTIFICATION PROCESS AS BPaaS FOR ELECTRICITY SUPPLY SYSTEMS
Lai Xu, Paul de Vrieze, and Nan Jiang
School of Design, Engineering and Computing
Bournemouth University, Bournemouth, UK
{lxx, pdvrieze, njiang}@bournemouth.ac.uk

Abstract

Business Process Management (BPM) systems have been deployed in many large organizations to improve their business effectiveness and efficiency. Cloud based BPM systems have provided SMEs using BPM in a pay-per-use manner. Previous work has focused on looking at cloud based BPM from the perspectives of distribution of data, activity or/and business engine and related issues, such as scalability of system, security of data, distribution of data and activities. To achieve business agility business process collaboration needs to seamlessly connect local BPM systems and cloud based BPM systems. In this paper we look at BPM in the cloud from a new user perspective, how process models can be handled in the cloud for the fast pace of change of business collaborations. The paper proposes a distribution solution in which at the design time, the shared process model can be discovered from a process repository, and adapted to local needs; at run-time a process is distributed. A real world case is used to explain our design and implementation. Collaborative process for incident notifications is built to work across different organizations.

Keywords: Business Process as a Service, Incident Management, Business Process Mashup, Cloud Computing, Service-oriented Computing.

1. INTRODUCTION

Industrial incidents are inevitable. Those related to our everyday life, such as power outages like the 2012 blackout in India (BBC News, 2012), will usually have strong adverse impact on the society and economy. These impacts are commonly measured as direct and indirect economic cost (La Commare & Eto 2004), (Lineweber & McNulty, 2001) where the latter is often large. For example, apart from enormous economic recovery costs, the nuclear accident at Fukushima has resulted in radical community reaction due to the lack of effective communications from main stakeholders (Secretariat, 2012).

The process of handling incidents, and specifically notifying stakeholders thereof, is a typical situational application that is not expected to occur often. Incidents generally involve multiple parties and to reduce the impact of the incidents, joined effort is needed. The situational applications handle business needs within a short period for a small group of users while the traditional BPM systems are supporting mission critical and core business of the organization (Balasubramanizm et al. 2008), (Pahlke, Beck & Wolf, 2010), (de Vrieze, Xu & Xie, 2010). In this paper, we therefore look at cloud based business process management for situational applications which are beyond traditional BPM systems.

The Internet as a conduit for information exchange has fundamentally changed how business is conducted. Cloud computing has enabled users to use computing resources in a pay-per-use manner and perceiving these resources as unlimited. At the cloud computing age, business process management systems have also adapted themselves into cloud environment. Business Processes as a Service (BPaaS) as a relative new concept, could be any type of business process that is delivered based on a cloud provisioning model. BPaaS can be seen as a new trend for Business Process Management (BPM). BPaaS can therefore classified differently according to the cloud based setting, i.e. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Business process service can be further classified as a SaaS based Business Process Service, a PaaS based Business Process Service, and an IaaS based Business Process Service.

SaaS based Business Process Services provide the process oriented capabilities as a readymade solution that is more appropriate to smaller organizations. In this approach the customers use a cloud based BPM solution as a pre-configured system where all business processes are predefined or only slightly modified based upon simple

doi: 10.29268/stcc.2014.2.4.4
configuration variables. SaaS based BPM allows for customers to forgo the need of specialist administration and up-front investments in soft and hardware.

When existing BPM systems migrate to cloud based platform services, business process engines run on cloud side, providing a sort of PaaS capabilities for all cloud users. Individual users specify their business logic and the specification, which are uploaded to the business process engine in the cloud for execution. The PaaS cloud provider provides execution, reliability and storage capabilities. The services of business process management for users are provided in a pay-per-use manner. Organizations that choose PaaS BPM do not have to buy expensive hardware and BPM software. No maintenance of the BPM platform is required. Multiple organizations share same business process management platform, each tenancy is responsible for its business logics or business process models though.

When existing BPM systems migrate to the cloud based infrastructure services, for the organization the only change is the hardware where the BPM systems runs. The organization still owns the business process management system, business process logics, process activities, and data. However, the organization does not need to do maintenance and updating of hardware, which is taken care by infrastructure cloud provider.

Due to the fear of losing or exposing sensitive data, cloud based BPM is at this time not widely adopted, and often only used on small scales. In this paper, we explore how cloud based BPM can be useful for enterprise needs. Especially we look at how a cloud based BPM can be applied for industrial incident management. The incident notification process is normally a collaborative process because industrial incident management most of time involves multiple parties. This kind of ad-hoc notification process is not supported by existing BPM systems.

The remainder of this paper is organized as follows, Section 2 discusses related work. Section 3 presents an incident notification case in the Spanish electricity system. Section 4 analyses the requirements and design principles of the incident notification system. Cloud based business process applications and new distribution patterns are discussed in Section 5. Design and implementation are discussed in Section 6. Service improvement and benefits analysis of our solution is presented in Section 7. We conclude our work and point out our future research in Section 8.

### 2. RELATED WORK

Related work is addressed from two aspects: the aspect of process-oriented solutions for incident management and the aspect of cloud based BPM.

In our previous research, we have designed and implemented business process mashup for incident notification in a cloud environment. Paper (Jiang et al. 2012) looks at data as a service in a cloud environment. Our research includes the design and implementation of a business process mashup engine (de Vrieze et al. 2011) and lightweight business process modeling language (Xu et al. 2010) (Xie, Xu, & de Vrieze, 2010). Special attention is paid to the end-user aspect which is orthogonal to extending our designs for cloud based BPM.

In recent years a good amount of work on providing BPM capabilities in the cloud has appeared. It specially attracts SME (Small and Medium Enterprise), as they now can use scalable BPM services in a pay-per-use manner without incurring large maintenance costs (Buyya, Yeo, & Venugopal, 2008). Major IT vendors have begun to provide cloud based BPM services, such as Salesforce’s Sales cloud, Market cloud (Salesforce, 2012), IBM’s Blue Works (IBM,2013), Vitria’s M3O (VitriaCloud,2013), and a solution based upon Amazon EC2 cloud (Amazon, 2013). These services are at different cloud levels.

There is also academic research on cloud based BPM. Han et al. (2010) present a design architecture for cloud based BPM with user-end distribution of non-compute-intensive activities and sensitive data. An analysis of optimal distribution of activities and data for synthetically utilizing is given from both user side and cloud side resource perspectives. Santos et al. (2012) and Anstett et al. (2009) discuss how to transfer a business process into the business processes for both user and cloud sides.

The previously mentioned papers (Han et al. 2010), (Santos et al. 2012) present a PAD (Process engine, Activity execution and Data) storage model that has been used to classified four types of BPM: traditional standalone BPM, user side BPM with cloud side distribution, cloud based BPM...
BPM with user side distribution and SaaS based BPM. In traditional BPM process engine, activity execution and data are local to organization of the user (see Figure 1 (a)).

For a SaaS based BPM, process engine, activities, and data exist in the cloud (Figure 1 (d)). For the user side BPM with cloud side distribution, it means that a process engine, activities and data (P, A, and D) are still running at the user side, and only distribute some compute-intensive activities and those parts of the data that are compute bound to it are distributed to the cloud to take advantage of stronger capabilities and better performance (see Figure 2 (b)). For cloud based BPM with user-end distribution, the process engine is in the cloud and the process models are enacted there, but activities can be run at the user side for sensitive, or local, data (see Figure 1 (c)).

3. Supporting Case

In order to demonstrate how the proposed solution can be used to support incident management we provide a case study addressing power outages (i.e. ‘blackouts’) in the Spanish national electricity system.

3.1 Spanish Electricity System and Key Stakeholders

A national electricity system is generally formed with of high-voltage electric power transmission network and grid connecting power stations and substations to transport electricity from where it is generated to where it is needed. In an industrial perspective, there are number of stakeholders in the system (see Figure 2): generators, distributors (also called network owners and operators) and suppliers, market operators, and system operators. The system operator is responsible for operating the nation’s power transmission system and electricity grid. Red Electrica de Espana in Spain is the system operator and OMEL is the market operator (REE, 2012). Note that a utility company like Gas Natural Fenosa is both a generator and distributor in the system.

![Figure 2. Stakeholders in Spanish Electricity System](image)

The electricity transmission starts from power stations where energy is generated from various sources by generators. The production is later transformed to a high-voltage and transported to the system operator through the transmission network. After that, it is transmitted from power substations owned by distributors, through an output line substation, to a transforming center and is finally transformed to the needed voltage level for different consumption needs at the supplier level. Since electricity cannot be stored in large quantities, the whole process must work continuously without any interruptions. This becomes even more difficult when power is generated intermittently from renewable sources. Moreover, the system also needs to be balanced to make sure that the demand is met by supply at the same time.

Incidents and Issues in the Electricity System. The whole electricity system is complex so incidents can occur anywhere and anytime in the process of electricity transportation. These incidents, ranging from signal errors, cabling problems to serious substation overloads, will affect energy supply, lead to power cut or even generate huge impact to the community and economy. Incidents have to be coordinated at system level which involves the system operator and all energy distributors as the latter may use each other’s network towers and substations to service their clients in affected regions. On the end-user side, larger industrial users are connected directly to the high voltage network. Because of the interruptible contracts (Baldick, Kolos, & Tompaidis, 2006) they must be notified of disruptions. However, domestic users normally interact with (electricity or power) marketers. The distributors, system operators, and distributors do not have the contact details of the domestic users. In the current systems domestic users are not notified of disruptions.

The problem lies in here: energy distributors, operators, and marketers face different markets and different responsibilities. They operate separate businesses and contain different customer and operational data. For the daily operation, there is no need to share certain data. However, when incidents/interruptions occur, there is need to notify not only industrial users, also domestic users.

4. Requirements and Principles

Incidents can be singular acts, but can also consist of a number of related (sub-) incidents that each contribute to an incident. The notification of incidents or interruptions may thus need to involve different partners and their information systems. The function of the incident notification is important for consumer satisfaction of electricity supply, but does not belong to daily operations for any involved partners. It is used only for a few weeks. It is also very hard to predict when the incidents may happen. In any case incidents should not occur often for a single distributor view. However, on a country level, minor incidents could occur rather frequent, i.e. at least once per month.
To improve domestic customer satisfactions of the power supply services, this research is driven by the energy distributors. The energy distributors are looking for a solution which does not offend (electricity or power) marketers who possess the contact details of domestic customers, but may not have proper IT knowledge to quickly build and maintain an application which is not a daily operation for the marketers (there is a large variation in size of the marketers).

The proposed solution needs to collect data from different partners. There is no individual partner to take responsibility for it. The first priority is to identify what is the best way to create, deploy, and maintain the application. The owner(s) of the application should be a permanent stakeholder such as the market operator rather than those on fixed term contracts like marketers as their contracts may not get renewed. The system needs to be designed to allow for implementation with minimal maintenance required. At the same time local modifications must be possible to allow for marketer driven enhancements and innovation.

To meet the objective of combining data from various systems the proposed solution must work with many different systems from different partners which could use varying technologies, standards, data formats etc. Reliability of the proposed solution is an important factor that must be ensured. The proposed solution needs to be ready to create notifications as soon as a problem is detected. The solution should be rarely used but it must be tested regularly to ensure its reliability especially after changes are made to ensure they have not affected its operation.

Costs of the solution and the notification methods used are also a key factor and these will be considered when specifying the solution. Due to the nature of the problem (power loss) at least some of the notification options should work without general availability electricity. In some cases there maybe advance notice of the outage, e.g. work will begin on substations between 12am and 2am and power will be unavailable throughout, and notification systems that require electricity can be used before the power outage. During power outages, especially when advanced notice was not possible, options that do not need electricity should be considered to notify affected customers. However options that require power can still be considered as affected customers may visit places with power and see the notifications there. As the duration of the problem increases the amount of notifications used should ideally also increase.

The requirements can be listed as follows:

- The system must integrate data from different owners and from different systems.
- A list of affected streets addresses should be obtained.
- A list of the affected customers for one company should be obtained.
- The list of the affected customers should be checked.
- All data processing must be done in line with data protection laws.
- The information about the streets affected, the problem and forecast must be published on the marketers' website in order to allow customers without mobile, to have access to the information.
- All customers must be informed about the problem, not only the most important.
- All customers in the area affected must be informed as soon and accurately as possible about the incident with appropriate information on causes and timeframes.
- The information must be sent by mobile text.
- In case the problem cannot be solved in the estimated time the system must send the up-to-date information with the new forecast.

The principles and rationale behind our design are cloud principle, web principle, service-orientation principle, and reusability principle

Cloud computing provides business users the opportunity of offering different services in different forms for different requirements, i.e. SaaS for end users, PaaS for application developers and IaaS for provisioning. Internet-based data management and computing facilities are the most popular functionalities delivered as cloud services. It also provides a scalable multi-user virtual community delivering cloud services, which use the Internet for controlled sharing and collaboration. Providing cloud services also move the maintenance pressure from individual partners.

A Web-based system is easy to extend; in principle every partners should be able to contribute effortlessly to the system either as a provider or consumer of information. The usage of the proposed solution must be as simple, smooth, and unrestricted.

The solution should support the collaborative process from several computing power to run a single task, transparently and coherently. The system also allows the transparent distribution of components over the network so that executing processes running in middle ware can be scaled across numerous physical servers which span across organizations over a cloud environment.

The cloud based BPM solution allows the integration of heterogeneous proprietary and legacy solutions through standards based common interfaces where they exist. Interoperability on the Web is platform and vendor neutral allowing all providers and requesters of information to participate on level playing field.
Service-orientation provides a broad design paradigm that provides separation of concerns on internet scales. It uses services as the basic building blocks of functionality. Service-orientation provides a way to think about the design of a solution in terms of services, service-based development and the outcomes of the services. The cloud based BPM solution aims at enhancing efficiency, agility, flexibility and productivity by positioning services as the primary functional elements.

The reusability principle involves the concept of reusability. In the process modeling field workflow patterns are regarded as reusable elements. A subset of workflow patterns has been adopted as basis of our process patterns and process templates (Xu et al. 2010) (Xie, Xu, & de Vrieze, 2010). Process patterns, process templates, process fragments, reference models, best-practice models, and further example models are provided in association with the process modeling language; these are all reusable.

Each involved partner can select a suitable process model which can be modified according to the situations. The process model can be downloaded from the process model repository and the improved process model can also be uploaded to the process model repository. The repository provides additional features such as tagging, liking and sharing to encourage discovery and reuse.

5. **CLOUD BASED SOLUTIONS FOR SITUATIONAL BUSINESS PROCESS APPLICATIONS**

When the main components of BPM systems are identified as PAD: Process engine, Activities, and Data in papers (Han et al. 2010), (Santos et al. 2012), (Anstett et al. 2009) different bases cloud BPM solutions can be seen in Figure 1. Business process models have been seen as an important organization asset and they are related with data and activities. Many times they could be independent with process engine. They provide business logic to process engine when, where, how and whom should process certain activities. In traditional BPM system, business process models do not change often after the BPM system deployed. However, building business process models rapidly is a key factor for implementing situational process applications. Therefore, we add business process model as an important component of BPM services after PAD and the ownership of each components. The cloud based distribution patterns of business process services is extended accordingly in Figure 3.

![Figure 3. Cloud Based Business Process Management Services Distribution Patterns](image)

Four main kinds of cloud based business process services are classified as general cloud-based business process services, IaaS based business process services, PaaS based business process services, and SaaS based business process services. The color of the box represents the ownership of each component. The components in a blue box mean that the components belong to a cloud provider; the components in a light yellow box mean that the components belong to each user/organization. The box with dish line indicates the optional.
As traditional BPM, the organization possesses the data, activities, process models (or business logics), process engine, and hardware and infrastructure which runs its BPM systems.

General cloud-based business process services just like the traditional BPM. The organization takes control of the BPM but also takes advantage of cloud computing and storage power for some compute-intensive activities and large data. Therefore the part of data and activities are in cloud.

In IaaS based BPM users migrates their systems from the local computing resources into cloud computing resources without any changes of the process models. Although data, activities, process models and process engine are running in the cloud, the individual organization holds the ownership of them. The cloud provider provides BPM platform, and hardware and infrastructure for multiple organizations in the Paas BPM pattern. The each organization has itself data, activities, and process models running at the provider side. Some sensitive data and activities may still keep at the user side. The PaaS based BPM allows users/individual organizations to build change or extend their process models for their different needs.

A SaaS BPM, in which a BPM system is offered as a software service. Instead of having to buy hardware and software, the system is used in a pay-per-use manner. This cloud based solution can easily deal with scalability of the system, so that additional resources can be instantiated relatively easily in peak load situations. When the rush is over, the additional resource can be released. However, the fear of losing or exposing sensitive data in the cloud, the current cloud based BPM is adopted by some small scale of business department, e.g. adopting Salesforce for a sales department. Some of data and activities can belong to the users and running at the cloud. Another data and activities run at the cloud side. SaaS based BPM includes a fixed/semi-process model with limited flexibility.

In our solution for the incident notification in Figure 4, business process oriented mashup engines are distributed in the cloud and different partners. The process mashup engine is deployed by all distributors and marketers. It ensures that all involved partners can flexibly deal with incidents as they appear. The engine can also be used for different situational applications. Therefore, we include process mashup engine at both cloud and user sides. A process model repository is provided in the cloud together with a process editor at the client that joins both local and shared information. This cloud based editing (even in the local editor) uses social web features to promote collaboration and sharing of best practice models. Users can select a suitable process model from the process model repository to adjust for their own usage. The process editor can be used to view, test, and learn how the process model works, and can even be populated with historical data (if available).

6. Design and Implementation

Our lightweight process modelling language in (Xu et al. 2010), (Xie, Xu, & de Vrieze, 2010), is designed for business users who have a professional background knowledge and understanding of the tasks and processes they are trying to support, but who are not professional software developers (or modelers). These users cannot be expected to have received significant training in programming, modeling nor system design either. Because of the popularity of BPMN (Business Process Modeling Notation), we adopted BPMN symbols. Figure 5 shows the collaboration process model of the incident notifications. The problem case starts when there is a disruption in the electrical transmission grid which is the responsibility of Red Electrica de Espana (REE). This problem can affect one or several distribution substations. Almost instantly, both REE and the distributor substations are informed automatically of the failure by their operational systems called SCADA. The process can follow two paths.
There are two different ways in which the incident notification system can be triggered. In the first way, when the affected distributors are alerted about the failure, the distributors should obtain an approximate forecast about the time that the problem should be repaired. There are two operational systems BDI and SGC that in this case provide the information needed for acquiring the list of affected streets by blending those sources with the list of affected substations. At this stage, the distributor is able to publish the information on its web site. Only when this path is completed can the other processes could be completed.

Regarding to the other way, REE is alerted by its SCADA system about the disruption. Then, REE informs all the distributors, which can include the distributor-substation owner. Subsequently, distributors inform their respective marketers. At this stage, they have to wait until the affected-street-to-marketers list is published on the distributor web site.

Once the information related to a current incident has been released, marketers should extract the information of the all customers affected by combining the information extracted from the distributor system with their own customer data. Finally, the information will be released on their own marketers’ website. Furthermore, the customer should receive a SMS or/and facebook/twitter message with the information about the problem and the repair forecast.

In our approach, we use a business process oriented mashup engine. The business process oriented mashup engines are deployed for all distributors and marketers. It insures that all involved partners can flexibly deal with appeared incidents and it can also be used for other situational applications.
commercially sensitive data, the data from the side of the marketers is.

Therefore, it is necessary to the part of the processing that uses customer data at the marketer side, and therefore to have the marketers to run a business process engine. For the incident notification purpose, the marketers could download a common process model from the process model repository on the cloud. Based on the downloaded process model, the marketers can modify the process model to adopt different notification channels, for example, beside to send SMS messages, Facebook messages are also sent. The marketers can use a local process editor to allocate the data (affected customers' mobile phone number, Facebook ID, or Twitter ID) and run the business process at local process-oriented mashup engine.

For the distributors, the incidents or interruptions can be caused or observed in different parts of the organization. Therefore, after download a sample process model, the model can be modified according to the situation. The process can keep monitoring the process of the repair and information is consistently published on the Web using the local process engine. The implementation process model is presented in Figure 7.

The cloud provider mainly concentrates to maintain the process model repository and to provide some common web services. It also provides a process editor and a process engine for users testing the process model. The users are certainly able to upload their data for running their processes in case of the local process engine is out of order. A collaborative process which runs in the cloud can be supported for special cases, e.g. monitoring the collaborative process.

7. SERVICE IMPROVEMENT

As a prime example of this kind of incident, on 23 July in 2007 the Collblanch substation, owned by REE, suffered a fire. Around 320,000 people were affected by this incident, having energy cuts of between 6 to 60 hours. A lack of information for the population resulted in disturbances on Barcelona’s streets. The population’s dissatisfaction was felt by the Government and both companies implied REE and ENDESA. The fine was exemplary. Around 40 million Euros was demanded by the judge, and 21 million Euros by the Generalitat (Catalonia government). In addition, the National Energy Commission punished both companies. However, it is difficult to know the total amount of money spent on the incident and the impact concerning the customers' switch to other companies (Sanchez, 2007).

The current industrial recommendations for the SLA of call centre: Average handling time (AHT) should be between 4.5 and 6 mins. Agents wrap time: 10 seconds, means how long they can rest after a call. Speed of answer:
80% answered within 20 seconds and $90\%$ answered within 10 seconds. A regional call centre has a staff of 1,200 (in total, including managers and shifts) and they take 70,000 calls a normal month. It indicates 2300 calls a day. For 12 hours call in a normal working time, the average call per minute is 3.24.

For an incident like we mentioned above, if we do not change the average handling time, the call rate changes from 0.054 calls per seconds into 8 calls per seconds or more. According Erlang C traffic model (Zeng, 2003), it means that the call centre needs over 2,300 staff when an incident occurs. If the call rate change into 8 calls per seconds and the average handling time is 2 mins longer, over 4,300 staff are needed to handle the calls.

In Erlang C traffic model, $m$ specifies the number of staff in the call centre, $u$ represents traffic intensity: $u = \lambda T$ where $\lambda$ is the average call arrival rate and $T$ indicates the average call duration.

$$E_c(m, u) = \frac{um}{m!} + (1-p) \sum_{k=0}^{m-1} \frac{uk}{k!}$$

Table 1 presents our simulation results. There are three data groups. The first data group targets service level objective 60%, answer time is 240 seconds 4 mins, talk time is 300 seconds (5 mins), for processing 4 call rate per seconds (240 call per 10 mins) needs 1240 staff members; for processing 10 calls rate per seconds (600 calls per mins) needs 3100 staff members.

The second data group, if the talking time changes into 400 seconds (about 6 mins), for processing 4 call rate per seconds (240 call per 10 mins) needs 1640 staff members; for processing 10 call rate per seconds (600 call per 10 mins) needs 3690 staff. The third data group, we increase the taking time and reduce service level objective.

<table>
<thead>
<tr>
<th>Number of staff</th>
<th>Call rate</th>
<th>Talk Time[s]</th>
<th>Wrap time [s]</th>
<th>Target answer time [s]</th>
<th>Service Objective [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1240</td>
<td>4</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1333</td>
<td>4.3</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1488</td>
<td>4.8</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1550</td>
<td>5</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1612</td>
<td>5.2</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1736</td>
<td>5.6</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1860</td>
<td>6</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1922</td>
<td>6.2</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2077</td>
<td>6.7</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2170</td>
<td>7</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2325</td>
<td>7.5</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2480</td>
<td>8</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2790</td>
<td>9</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>3100</td>
<td>10</td>
<td>300</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>1640</td>
<td>4</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2050</td>
<td>5</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2460</td>
<td>6</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2870</td>
<td>7</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>3280</td>
<td>8</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>3690</td>
<td>9</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2040</td>
<td>4</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>2550</td>
<td>5</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>3060</td>
<td>6</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>3570</td>
<td>7</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>4080</td>
<td>8</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
<tr>
<td>4590</td>
<td>9</td>
<td>400</td>
<td>10</td>
<td>240</td>
<td>60</td>
</tr>
</tbody>
</table>

Source: Fictitious data, for illustration purposes only

Table 1. The results of Erlang C model simulations

So for an incident like we mentioned above, 320,000 people were affected. 600 calls per 10 mins is easily achieved. Therefore, the benefits of our solutions are obvious. First, it provides a centrally-organized notification.
process. Log files can be used as evidence for the future identifying of responsible partners in the case that customers are not notified. The distributors do not have access to contact details of domestic customers, but they are held responsible for informing customers by the government. This notification process depends very much on the efforts of the marketers. There is no effective mechanism for monitoring the marketers' notification efforts.

Besides solving the notification problem, as a second benefit, incident or interruption notification process models can be shared among distributors, REE and marketers. An improved process model can be uploaded by each partners and managed by the market operator OMEL. Third, the (Web) services related to the incident notification process can also be shared to improve performance. Multiple process mashup engines run at different sides, which also allow them to function as backups for each other. In general, a process-oriented solution will be able to significantly reduce the communication time of domestic user awareness of the incident and related information.

8. CONCLUSIONS
Cloud-based BPM systems bring many benefits to business users. They provide a good opportunity for organizations which seek saleable and flexible solutions. Cloud-based BPM systems are not likely to replace all local BPM systems. To achieve business agility business process collaboration needs to seamlessly connect local BPM systems and cloud-based BPM systems. In this paper, we extend the PAD (process engine, activity, and data) model (Han et al., 2010). A new user perspective is taken, i.e. not only process engine, activity and data can be distributed; in our new mode, business process models can also be distributed and adopted for different purposes. This extension is a big step for PaaS BPM. We have use incident notification process to verify our design.

Effective incident management can be considered as a more cost-effective solution to reduce the negative impact on the community and economy after an industrial incident occurs. From the end-user perspective, this relies on instant situation awareness and response. The core concept of the architecture is that it uses verified incident notification process models and a business mashup engine to help organizations respond to emerging situations triggered by incidents more intuitively. The architecture also supports on-demand and flexible situational applications which are used to address timely and immediate customer needs. The case study has demonstrated how the framework facilitates in solving a real world problem effectively. Whilst the implementation was based on the Spanish electricity supply system, it can also be applied to other national electricity supply systems especially for the European Union countries in compliance with Act 17/2007. Moreover, the situational application shown in the case study focuses on electricity incident management which can also be extended to other incident management areas where immediate collaborations based on data sharing and access are needed.

9. ACKNOWLEDGMENT
This work is made possible by the support of the Natural Science Foundation of China (NSFC) under Grant No.61150110484, ESSENTIAL: Enterprise Service deSign based on Existing software Architectural knowLedge. References (ESSENTIAL)

10. REFERENCES
LaCommare, K.H., Eto, J.H. (2004). Understanding the cost of power interruptions to u.s. electricity consumers
Lineweber, D., McNulty, S. (2001). The cost of power disturbances to industrial and digital economy companies
Secretariat of the Investigation Committee on the accidents at the Fukushima Nuclear Power Station. (2012). Investigation committee on the accident at the fukushima nuclear power stations of tokyo electric power company


VitriaCloud. (2013) Vitriacloud m3o in the cloud

Amazon. (2013) Amazon elastic compute cloud (amazon ec2)


de Espaa, R.E. (2012) company profile


Authors

Dr. Lai Xu is a senior lecturer in faculty of Science and Technology (SciTech) of Bournemouth University. Previously she was a Senior Researcher at SAP research, Switzerland, a Senior Research Scientist at CSIRO ICT Centre, Australia, a post-doctoral researcher at the Institute of Information and Computing Sciences of the Utrecht University and the Department of Computer Science of the Free University Amsterdam. She received her Ph.D. in Computerized Information Systems from Tilburg University in 2004. Her research interests include BPaaS, process mashups, business process collaboration, and business process modelling.

Paul de Vrieze is a Senior lecturer faculty of Science and Technology (SciTech) of Bournemouth University. Previously he was a Senior Researcher at SAP research, Switzerland, and worked as a Postdoctoral Research Fellow at CSIRO ICT Centre, Australia. He received his Ph.D. in Information Systems from Radboud University Nijmegen, The Netherlands in 2006. His main research interests are in enterprise information system integration, user modeling, systems modeling, and semantics integration.

Dr. Nan Jiang is a lecturer of faculty of Science and Technology (SciTech) at of Bournemouth University. He received his PhD in Web Usability from Department of Computer Science, Queen Mary, University of London. He is mainly specialised in HCI area but he is also interested in adpting emerging technologies such as cloud computing, enterprise mashups and crowdsourcing from user perspective.
Call for Articles
International Journal of Services Computing

Mission
The International Journal of Services Computing (IJSC) aims to be a reputable resource providing leading technologies, development, ideas, and trends to an international readership of researchers and engineers in the field of Services Computing. To ensure quality, IJSC only considers extended versions of papers published at reputable international conferences such as IEEE ICWS.

From the technology foundation perspective, Services Computing covers the science and technology needed for bridging the gap between Business Services and IT Services, theory and development and deployment. All topics regarding Web-based services lifecycle study and management align with the theme of IJSC. Specially, we focus on: 1) Web-based services, featuring Web services modeling, development, publishing, discovery, composition, testing, adaptation, and delivery, and Web services technologies as well as standards; 2) services innovation lifecycle that includes enterprise modeling, business consulting, solution creation, services orchestration, services optimization, services management, services marketing, business process integration and management; 3) cloud services featuring modeling, developing, publishing, monitoring, managing, delivering XaaS (everything as a service) in the context of various types of cloud environments; and 4) mobile services featuring development, publication, discovery, orchestration, invocation, testing, delivery, and certification of mobile applications and services.

Topics
The International Journal of Services Computing (IJSC) covers state-of-the-art technologies and best practices of Services Computing, as well as emerging standards and research topics which would define the future of Services Computing. Topics of interest include, but are not limited to, the following:

- Services Engineering
- XaaS (everything as a service)
- Cloud Computing for Internet-based services
- Big Data services
- Internet of Things (IoT) services
- Pervasive and Mobile services
- Social Networks and Services
- Wearable services
- Web 2.0 and Web X.0 in Web services
- Service-Oriented Architecture (SOA)
- RESTful Web Services
- Service modeling and publishing
- Service discovery, composition, and recommendation
- Service operations, management, and governance
- Services validation and testing
- Service privacy, security, and trust
- Service deployment and evolution
- Semantic Web services
- Scientific workflows
- Business Process Integration and management
- Service applications and implementations
- Business intelligence, analytics and economics for Services
Call for Articles
International Journal of Big Data

Mission
Big Data has become a valuable resource and mechanism for the practitioners and researchers to explore the value of data sets in all kinds of business scenarios and scientific investigations. New computing platforms such as cloud computing, mobile Internet, social network are driving the innovations of big data. From government initiative perspective, Obama Administration in United States launched "Big Data" initiative that announces $200 Million in new R&D investments on March 29, 2012. European Union also announced "Big Data at your service" on July 25, 2012. From industry perspective, IBM, SAP, Oracle, Google, Microsoft, Yahoo, and other leading software and internet service companies have also launched their own innovation initiatives around big data.

The International Journal of Big Data (IJBD) aims to provide the first Open Access publication channel for all authors working in the field of all aspects of Big Data. Big Data is a dynamic discipline. One of the objectives of IJBD is to promote research accomplishments and new directions. Therefore, IJBD welcomes special issues in any emerging areas of big data.

Topics
IJBD includes topics related to the advancements in the state of the art standards and practices of Big Data, as well as emerging research topics which are going to define the future of Big Data. Topics of interest to include, but are not limited to, the following:

**Big Data Models and Algorithms** (Foundational Models for Big Data, Algorithms and Programming Techniques for Big Data Processing, Big Data Analytics and Metrics, Representation Formats for Multimedia Big Data)

**Big Data Architectures** (Cloud Computing Techniques for Big Data, Big Data as a Service, Big Data Open Platforms, Big Data in Mobile and Pervasive Computing)

**Big Data Management** (Big Data Persistence and Preservation, Big Data Quality and Provenance Control, Management Issues of Social Network enabled Big Data)

**Big Data Protection, Integrity and Privacy** (Models and Languages for Big Data Protection, Privacy Preserving Big Data Analytics Big Data Encryption)

**Security Applications of Big Data** (Anomaly Detection in Very Large Scale Systems, Collaborative Threat Detection using Big Data Analytics)

**Big Data Search and Mining** (Algorithms and Systems for Big Data Search, Distributed, and Peer-to-peer Search, Machine learning based on Big Data, Visualization Analytics for Big Data)
