Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 21:43 14 November 2016 (PT)

Emerald Insight

Kybernetes

Adaptive SLA mechanism for service sharing in virtual environments

Yong Yin Jian Zhou Chaoyong Zhang Dejun Chen

Article information:

To cite this document:
Yong Yin Jian Zhou Chaoyong Zhang Dejun Chen , (2016),"Adaptive SLA mechanism for service sharing in virtual environments", Kybernetes, Vol. 45 Iss 7 pp. 1036 - 1051
Permanent link to this document:
http://dx.doi.org/10.1108/K-03-2014-0057

Downloaded on: 14 November 2016, At: 21:43 (PT)
References: this document contains references to 25 other documents.
To copy this document: permissions@emeraldinsight.com
The fulltext of this document has been downloaded 30 times since 2016*

Users who downloaded this article also downloaded:

(2016),"The use of distinction in the process of communication: Aspects of a system-theory of communication", Kybernetes, Vol. 45 Iss 7 pp. 1013-1023 http://dx.doi.org/10.1108/K-09-2015-0245

(2016),"The dynamic behaviors of a supply chain with stock-dependent demand considering competition and deteriorating items", Kybernetes, Vol. 45 Iss 7 pp. 1109-1128 http://dx.doi.org/10.1108/K-04-2015-0112

Access to this document was granted through an Emerald subscription provided by emerald-srm:563821 []

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/0368-492X.htm

Adaptive SLA mechanism for service sharing in virtual environments

Yong Yin
Key Laboratory of Fiber Optic Sensing Technology and Information Processing, Ministry of Education, Wuhan University of Technology, Wuhan, China
Jian Zhou
School of Information Engineering, Wuhan University of Technology, Wuhan, China
Chaoyong Zhang
The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, China, and
Dejun Chen
School of Information Engineering, Wuhan University of Technology, Wuhan, China

Abstract

Purpose – The purpose of this paper is to discuss an adaptive SLA mechanism for service sharing in virtual environment, which can organize and govern QoS items in terms of service execution time, reliability, and availability, and provides a common understanding about services, responsibilities, priorities, guarantees and warranties-related virtual cooperative issues.

Design/methodology/approach – The management framework for SLA is introduced, based on which the whole process including SLA contract, adaptive SLA negotiation strategy, SLA deployment and SLA assessment are discussed, and the prototype is implemented in the cloud manufacturing platform.

Findings – A proposed SLA framework for service sharing in virtual environments is given; electronic contracts are designed in the framework for encapsulating measurable aspects of service level agreements so as to provide common understanding about the service; and an improved SLA negotiation strategy with three phases is presented for the dynamicity of the virtual services.

Originality/value – The paper presents a very useful adaptive SLA mechanism for service sharing in virtual environments that can be utilized in concurrent or future advanced manufacturing modes.

Keywords Virtual environment, SLA, Electronic contract, Negotiation strategies

Paper type Research paper

1. Introduction

In recent years, a lot of innovative manufacturing modes have been appearing in the world-wide collaborative manufacturing and the global resource sharing environments. Typical examples of these sophisticated manufacturing modes are virtual alliance (VA), application service provider (ASP), networked manufacturing (NM), manufacturing

Kybernetes
Vol. 45 No. 7, 2016
pp. 1036-1051

0368-492X
DOI 10.1108/K-03-2014-0057

The authors would like to acknowledge the support of the funding from the National Science Fund Committee (NSFC) of China (Grant No. 51105289, 51275190) as well as the contributions from all members of the research group.

grid (MG), cloud manufacturing (CM) and so on. These manufacturing modes have two common features as follows:

1. they constitute a temporary cooperative network that exists on the basis of traditional internet and aims at the sharing of resources such as skills, knowledge, costs and benefits so as to achieve the prompt answering to the market opportunities for products and services; and
2. enterprises laying over the internet exist in a virtual form and remain globally distributed, dynamic and heterogeneous for the dynamic nature of the market opportunities, bringing in complementary competencies and co-ordinations available to each other in the virtual environments for manufacturing activities.

The above common characteristics for these advanced manufacturing modes imply a demand for SLA mechanism.

Meanwhile, burgeoning standards and specifications spur service sharing in virtual environments with a service-oriented architecture (SOA) (Cristani and Cuel, 2004), and manufacturing activities have been transformed from "centralized in producing" to "centralized in service" with features of "service-based manufacturing" and "manufacturing-oriented service" and the idea of "service" has been an inevitable uptrend of their aims and principles for concurrent manufacturing enterprises. To the best of our knowledge and investigation, currently there are three dominant kinds of service access mechanisms, i.e., the web service (WS) (Thomas, 2005), the open grid services architecture (OGSA) and the web service resource framework (WSRF) (Czajkowski *et al.*, 2004). Among them the WSRF is an efficient, appealing and widely accepted approach for service access in such virtual environments (Tuecke *et al.*, 2003). It is based on the web services description language (WSDL) to describe the service interfaces, in which the universal discovery, description and integration service registry acts as service registration approach (Foster *et al.*, 2001) and the simple object access protocol (Don *et al.*, 2000) serves as a communication mechanism. It provides the opportunity for both the manufacturing service suppliers and consumers to dynamically establish and dismiss a business relationship on a case-by-case basis and on-demand. In such circumstances, WSRF is the first choice for the SLA consideration.

From the above introduction, it is impossible for enterprises involved in manufacturing service sharing activities to acquire prior understanding and knowledge on each other for the features of the boundary crossing, geographical dispersion, temporary, dynamic and virtual nature of these manufacturing modes. Thus definite and explicit electronic contracts are necessary to establish and maintain the fine and mutually beneficial service relationship in the virtual and service-oriented environments and mutual negotiation process become indispensable for service providers and consumers to reach an agreement. The electronic contracts and the negotiation process above can be organized and governed by service level agreement (SLA) which defines the scope of consumption and provision of service (Keller and Ludwig, 2003) and provides a common understanding about services, responsibilities, priorities, guarantees and warranties-related virtual cooperative issues (Czajkowski *et al.*, 2002). The paper focusses on the issues with SLA mechanism for service sharing in virtual environment, and its main contributions are summarized as follows:

1. a proposed SLA framework for service sharing in virtual manufacturing environments is given;

(2) electronic contracts are designed by WSDL encapsulating measurable aspects of SLAs, so as to provide common understanding about the service shared in the WSRF manufacturing architecture;

(3) an improved SLA negotiation strategy with three phases is presented for the dynamicity of the virtual services; and

(4) the prototype implementation of the SLA mechanism proposed in the paper is briefly introduced on a cloud manufacturing platform.

2. Backgrounds

SLA mechanism has been widely utilized in telecommunications or computing fields for their service provision. The *Tele-Management Forum's SLA Management Handbook* originally defined SLA for the telecommunications industry as follows:

> It is the Service Level Agreement (SLA) that defines the availability, reliability and performance quality of delivered telecommunication services and networks to ensure the right information gets to the right person in the right location at the right time, safely and securely. The rapid evolution of the telecommunications market is leading to the introduction of new services and new networking technologies in ever-shorter time scales. SLAs are tools that help support and encourage Customers to use these new technologies and services as they provide a commitment from SPs (Service Providers) for specified performance levels (Philip *et al.*, 2008).

The definition of SLA in a SOA environment is much recent, but it is becoming important as services on SOA architectures can cross organizational boundaries and thus extremely suitable for grid environment. When employed in SOA context, SLA is defined by some references as the following:

> A Service Level Agreement is a document that includes a description of the agreed service, service level parameters, guarantees, and actions and remedies for all cases of violations (Erik *et al.*, 2001).

The main idea of SLA is to provide an explicit definition of the formal agreements about service terms such as performance, availability, reliability and even billing. SLA for services in business such as IT services or for WSs have recently been widely studied and a great deal of work dealing with SLA-based quality of service (QoS) has been performed. An example for the application of SLA in IT services is the Information Technology Infrastructure Library with its main purpose to ensure that deployed applications can meet their defined service levels (Antonio *et al.*, 2013). Internet backbone service providers (or network service providers) also adopt SLA to explicitly state their own service level on websites (George *et al.*, 2014). An overview of SLA in IP networks is offered in Mobach *et al.* (2006). The work suggests three common approaches to satisfy SLAs in IP networks. Service abstraction in OGSA for QoS properties emphasizing on the application layer is given in Al-Ali *et al.* (2002). Strategies for incorporation of SLA-based resource brokering into existing grid systems have been discussed in Ouelhadj *et al.* (2005). A general bi-criterion heuristic scheduling based on dynamic programming, called the dynamic constraint algorithm is proposed by Abrishami *et al.* (2012). A WSs-based QoS-aware workflow management system with SLA utilized in context of GridCC project is described in Dan *et al.* (2004). A novel resource management model is presented in Comuzzi *et al.* (2009), on which resource interactions are mapped onto a well-defined set of platform-independent SLAs. A framework is put forward in Venugopal *et al.* (2006) to provide customers with differentiated levels of service through automated management and SLAs. Also two main specifications for describing SLA for WSs are discussed in Zhao *et al.*, which are the WS-Agreement from open grid forum and the web service level agreement (WSLA) language and framework from IBM.

SLA mechanism is no stranger to telecommunications or computing fields, but it is just at its beginning for service sharing in dynamic manufacturing activities. According to our investigation and study, we are aware that existing work with SLA in IT fields provide a high degree of customer satisfaction with SLA. This is because only a small set of static services are provided and the services are governed by stringent regulatory policies. However, dynamic manufacturing services with characteristics of boundary crossing, geographical dispersion and natural dynamicity that are being deployed today on the internet exhibit SLA requirements that differ radically from those listed above. As for existing researches on SLA under SOA architectures, they may solve the problem for static SLA requirements with negotiations, but they cannot solve the problems in dynamic environment and in frequent changing requirements of services. This situation motivates us to settle items with SLA mechanism for service sharing in virtual manufacturing environments. To begin with, we first discuss the SLA framework in the following section.

3. SLA framework

Mutual satisfaction of manufacturing service sharing for both service providers and consumers plays an important role in virtual environments, and SLA provides the approach for the QoS assurance with a specification of the verifiable quality characteristics of the service. To illustrate this function of SLA, we discuss the management framework for SLA in detail at first.

A scene of a manufacturing task requiring virtual services over the internet can be imaged as follows. To fulfill this manufacturing task, several steps must be performed. When the task request arrives in the virtual environments, the scheduler starts the resource service discovering and filtering activities on the service candidate nodes over the internet at the first step, which results in the choosing of manufacturing services that can meet the task request properly. Next the negotiation process is conducted and the electronic contract creation is concluded by the commitment of the involved service providers and consumers. Following the signed contract, the SLA items are deployed. The forth step is to monitor and verify the contract executive process. In the end, the SLA may be terminated successfully or penalties may be imposed on a party for his breaking of one or more SLA clauses. With reference to Keller and Ludwig (2003), the proposed management framework for SLA can be demonstrated in Figure 1.

It should be mentioned that although the service querying and publishing activities in the management framework for SLA are the prerequisite for the implementation of SLA, the methods and technologies of these activities belong to the service searching and selection, which has no relation with SLA processes. Thus service querying and publishing are beyond the scope of the paper. SLA contract, SLA negotiation, SLA deployment and SLA assessment will be discussed in detail in the following sections.

4. SLA management

4.1 SLA contract

SLA contracts act as a means to define the service contained in the SLA. They govern the details of which service can be exposed and to whom from a given virtual domain. They also identify exactly what guarantees are being offered and which kind of

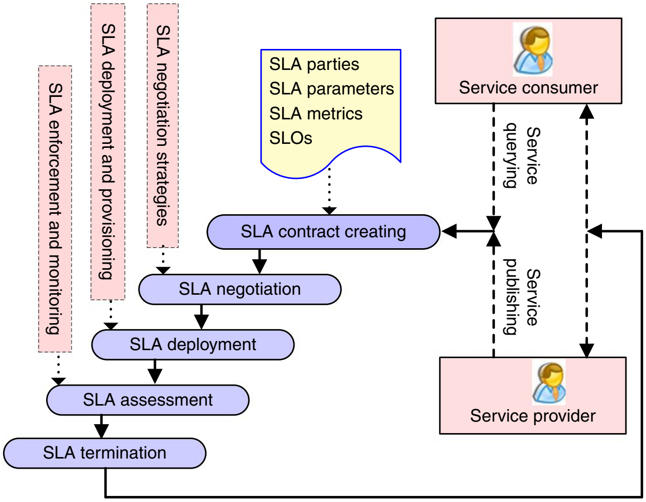


Figure 1.
The management framework for SLA

parameters, metrics and service level objectives (SLOs) in a formal way. Typically a SLA contract contains elements as follows:

- Description of the involved parties, their roles and the action interfaces they expose to the other parties of the contract.
- Description of SLA metrics representing methods for calculation of quality of service (QoS) values. The purpose of SLA metrics is to define how to measure or compute a service value. Besides a name, a type and a unit, it contains either a function or a measurement directive and a definition of the party taking charge of computing this value.
- Description of detailed specification of the service level parameters (SLA parameters). SLA parameters are specified by metrics defining how service parameters can be measured and specify values of measurable parameters. Examples of SLA parameters are service availability, throughput or response time.
- Description of SLOs determining contractually agreed objective values for specific SLA parameters. A SLO expresses a commitment to maintain a particular state of the service in a given period.

From the above discussion, a contract C can be defined by a triple set (P, Q, M), in which P represents the parties from the virtual domains involved in the SLA, Q expresses the assertions agreed over each party for QoS attributes related with the SLA parameters and metrics, while M is the set of methods (or operations) defined by the contract.

One example of P can be:

$$P = \{X \text{ company};\ \ Y \text{ factory};\ \ Z \text{ workshop};\ \text{etc.}\}$$

Assertions comprise mutual agreements or guarantees related with the service QoS defined in the contract. At any given time, Q takes on the set of values true or false depending on whether the items described in the assertion set is satisfied or not. The assertion set Q can be:

$$Q = \{\text{availability} > 99\%;\ \ \text{service time delay} < 100\text{ ms};\ \ \text{service price} < \$200;\ \text{etc.}\}$$

Methods defines the operations available on the contract, which will be used for service monitoring and enforcement. They permit the involved parties to query the values of the assertions in the contract. One example of P can be:

$$M = \{\text{measure};\ \ \text{verify};\ \ \text{query};\ \ \text{notify};\ \text{etc.}\}$$

SLA contract can be described by WSDL with WSRF architecture mentioned in Section 1, and it allows the SLA manager to associate the contract with a number of handlers, which hide the details of the service. One SLA contract example of the force analysis query service described by WSDL with WSRF architecture is given in the Appendix.

4.2 Adaptive negotiation strategy

In the virtual environment with distributed nodes providing assurance on service sharing for a manufacturing task, service QoS parameters must be negotiated and SLA must be created between the service provider and consumer. A manufacturing task generally requires the cooperation of the distributed nodes, so SLA must take into consideration multiple different services from different providers. A bilateral SLA is necessary to be negotiated and created between each of the individual service providers and consumer. The SLA creation process is a transaction composed of multiple bilateral SLAs. For the completion of the manufacturing task, either all individual bilateral SLAs must be created, or none is created resulting in the task failure. Before reaching an agreement of the QoS parameters, the negotiation process may be carried out between each of the individual service provider and consumer. In order to fulfill this purpose, a coordinator is necessary in the service scheduler. The coordinator is often provided by the core middle layer in the form of APIs. When changes occur in one node, the change information will be propagated by the coordinator to all the other nodes participating in the service for the manufacturing task. Upon the receipt of the change information from one node, other nodes taking part in the manufacturing task come into the "negotiation" state, and they have to make decisions whether to accept this change or not, and then send their decisions to the coordinator. If decisions from all nodes partaking in the manufacturing task are approving, the coordinator changes into a "commit" state, and the "commit" order is sent to all the nodes involved in the manufacturing task. Upon receiving the "commit" order, all the nodes must pledge to the commit and ensure the completion of the manufacturing task facing the change request. On the contrary, if the decision from one node entering into the manufacturing task is negative, the coordinator grows into an "abort" state and the "abort" order is sent to all nodes resulting in the cancel of the transaction, as shown in Figure 2.

In Figure 2, a two-phase commit protocol is represented as a finite state machine (FSM) (Antoine *et al.*, 2007). In FSM the result of the negotiation process must lead to a "commit" state or a "abort" one. It seems that the FSM can be qualified for the adaptation strategy of SLA before its creation. However, there does exist unexpected cases. The problem is that in case of the system malfunction such as the coordinator failure, it is impossible for all the nodes to know whether the transaction is committed or aborted. For example, when the coordinator breaks down after having sent a "commit" order to some nodes but not all, the remaining nodes cannot know whether the transaction should be aborted or committed.

In order to solve this problem, a three-phase strategy shown in Figure 3 is provided (Antoine *et al.*, 2007) with an intermediary state before the commitment. The intermediary state represents a state of "preparation" for the commit of SLA creation.

In Figure 3, when a service node receives an SLA negotiation, its state changes from "start" to "negotiation." Once the bilateral negotiation process between each of the involved service providers and consumers is conducted, a piece of request information for SLA creation is sent to all the involved nodes. Then each node responds with an agreement or a further negotiation request for a counter offer, or responds with a rejection of the negotiation. In case of a counter offer, the node remains in the "negotiation" state. It can also reject the negotiation and transfer to the "abort" state, thus leads to the abandonment of the manufacturing task. If all nodes agree, the coordinator sends a notification to all involved nodes to change from "negotiation" to "preparation" state. In the course of the state changing, the node stays "preparation" until all the other nodes are prepared; otherwise it goes to the "abort" state. In this way SLA is guaranteed for creation.

4.3 *SLA deployment*

After a SLA is created among service providers and consumers, SLA will be deployed over the internet to involved parties according to the SLOs defined in the contract,

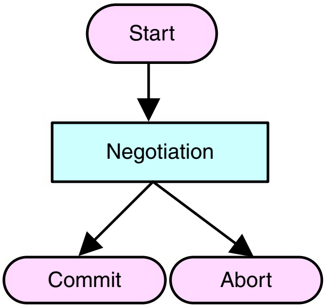


Figure 2.
Two-phase commit protocol

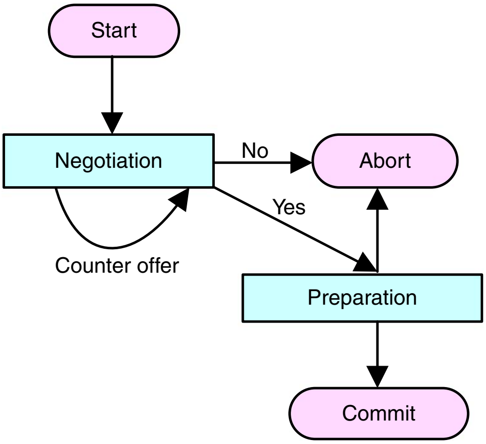


Figure 3.
Commit protocol with intermediary state

as shown in Figure 4. Thus the validity of the SLA will be checked and then be distributed either in full or in appropriate parts to the involved components.

Each signatory party uses its own independent deployment function that interprets the WSLA and takes appropriate action. Two steps are necessary for the SLA deployment. In the first step, SLA deployment function of a signatory party generates configuration information in the service deployment information (SDI) format with WSDL and sends it to its supporting parties. SDI is a fragment of WSDL document. Rather than containing a complete WSDL, SDI only includes relevant information for a particular party. The service activities are defined with the M set in SLA contract, as discussed in Section 4.1. For example, a measurement operation only contains SDI about how to retrieve and aggregate the metrics and parameters and how to interact with other parties (e.g. either to get these metrics or to make them available). In the second step, the supporting parties utilize service deployment functions to configure the services in order to perform their role in the process of SLA monitoring and enforcement, that will be discussed in the next section.

4.4 SLA assessment

Upon the completion of the SLA deployment, assessment activities including SLA monitoring and enforcement are required to retrieve SLA metrics and parameters, evaluate SLOs and take appropriate management actions. Thus signatory parties can predict and monitor the level of commitment of the service provider based on the contract for a better manage of their own requirements instead of relying solely on the provider to meet their obligations.

The aim of SLA monitoring is to dynamically monitor the state of the SLA-bound service during execution. By monitoring the execution process of the service, the SLA manager periodically gathers information to manage the operation status of the service. The actual performance metrics of a given service is tested and verified by the SLA manager. Execution results are compared with the required SLA parameter clauses. Appropriate control actions are adopted trying to prevent violations of the QoS parameters within the SLA. Facing a SLA failure, predefined management actions are

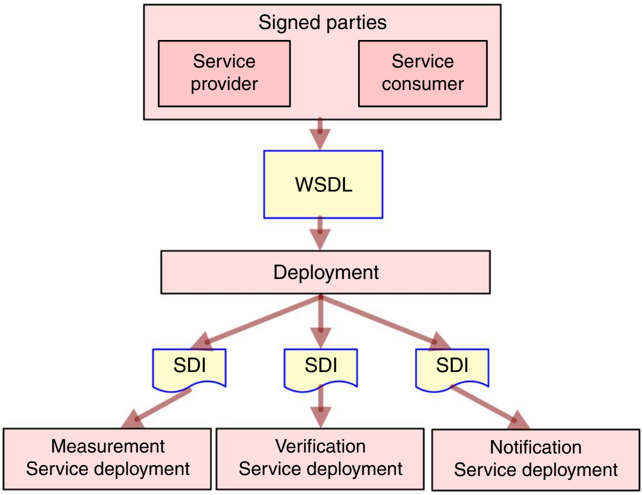


Figure 4.
SLA deployment

invoked. For example, decisions regarding the reconfiguration or termination of the SLA contract may be taken, and even penalties are inescapable. A simple monitoring and verifying model proposed by WSLA is illustrated in Figure 5 (Asit *et al.*, 2003).

5. Implementation

In above sections, SLA framework and its mechanism for service sharing in virtual environments have been discussed. In this section, we describe the prototype implemented on a cloud manufacturing platform that was supported by the project of "the National 863 Key Project on Cloud Manufacturing (2011AA040501)" from the Ministry of Science and Technology of China (Tao *et al.*, 2011). The architecture of the cloud manufacturing platform is described in Figure 6.

The access of the platform is web-based, namely, the communication among each virtual domain uses the HTTP protocol so that existing SSL and PKI certificates can be employed for security consideration. When cooperative activities are performed via the cloud portal over the cloud manufacturing platform, formal agreements about shared service terms such as performance, availability, reliability and payment, etc. are absolutely necessary. In the platform, SLA management is located on the cloud service core layer, which is written in Java language with all functions provided with Java API, and a Java-enabled web server is employed as its front-end. The WSRF-based manufacturing service server is located on the computer with a IP of 192.168.0.20 as the service provider, while the service consumer on the client of 192.168.0.12 want to access force analysis service for his product. The force analysis service request page is given in Figure 7, and the contract produced by the SLA management model is illustrated in Figure 8.

In order to match SLOs from the service provider and consumer, negotiation processes for the contract with QoS criteria follow according to the specifications in their SLA documents. The negotiation is in the charge of the negotiation service (NS) module of the SLA manager on the CM platform, as shown in Figure 9.

It is worth mentioning that the consumer's budget and deadline for acquiring resources are employed by the prototype to make a decision on accepting or rejecting

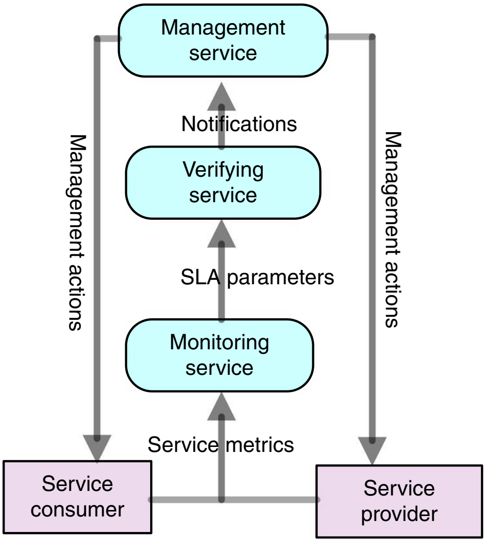


Figure 5.
A simple monitoring and verifying model

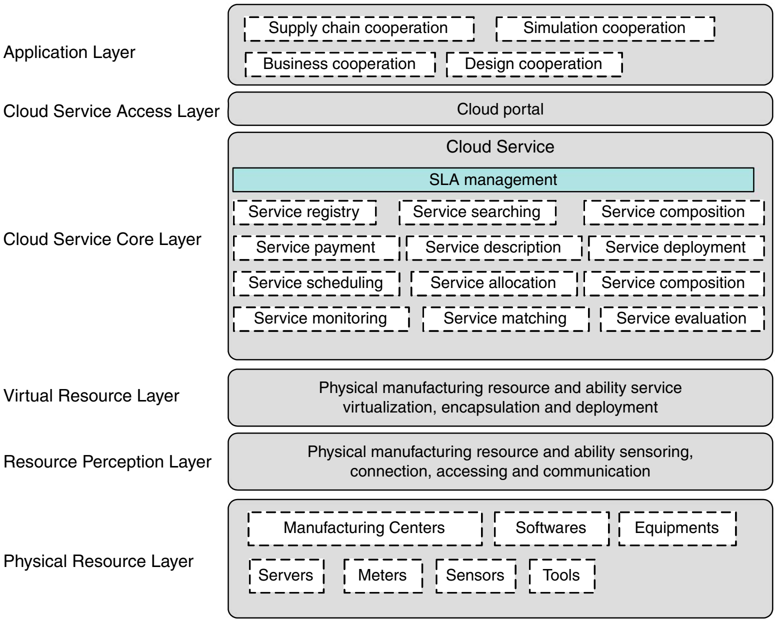
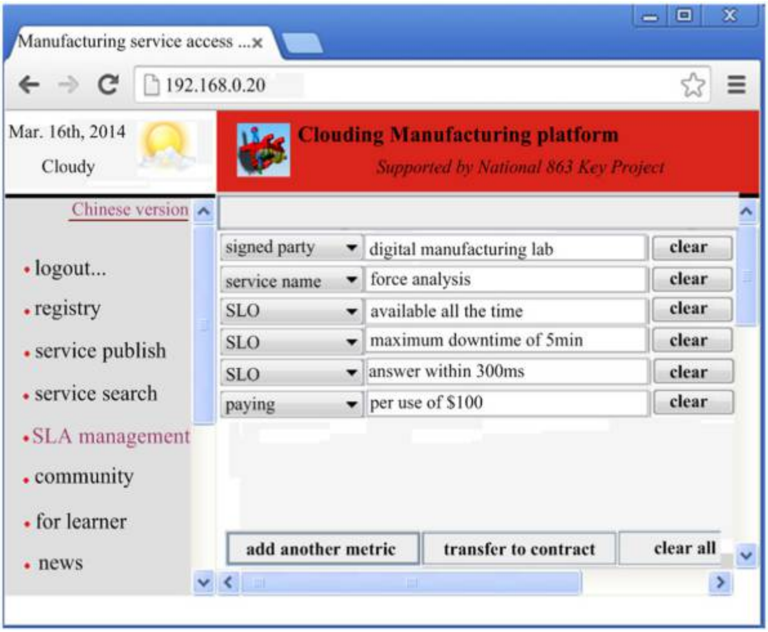


Figure 6.
Architecture of the cloud manufacturing platform



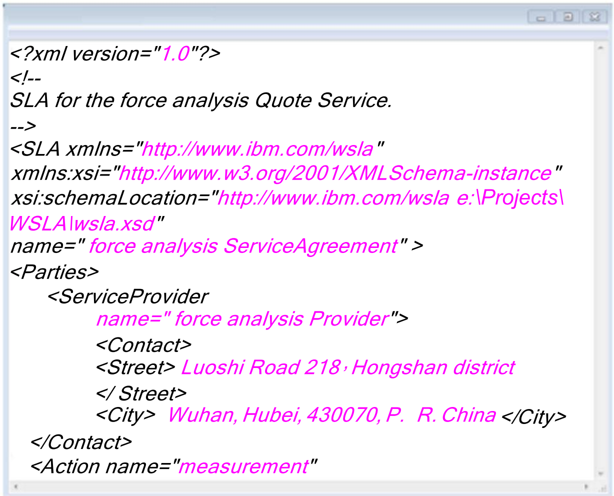
Mar. 16th, 2014
Cloudy

Clouding Manufacturing platform
Supported by National 863 Key Project

signed party	digital manufacturing lab
service name	force analysis
SLO	available all the time
SLO	maximum downtime of 5min
SLO	answer within 300ms
paying	per use of $100

Figure 7.
Service request page

the service. A time-dependent strategy is utilized in the SLA management as an input and then the negotiation processes are initiated by the NS for the service offer. Once the consumer receives the offer delivered by the NS sent from the provider, it negotiates with the provider for a counter offer. The SLA contract is signed ultimately on the basis



```
<?xml version="1.0"?>
<!--
SLA for the force analysis Quote Service.
-->
<SLA xmlns="http://www.ibm.com/wsla"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/wsla e:\Projects\
WSLA\wsla.xsd"
name=" force analysis ServiceAgreement" >
<Parties>
     <ServiceProvider
           name=" force analysis Provider">
           <Contact>
           <Street> Luoshi Road 218，Hongshan district
           </ Street>
           <City>  Wuhan, Hubei, 430070, P.  R. China </City>
   </Contact>
   <Action name="measurement"
```

Figure 8.
Contract produced by SLA

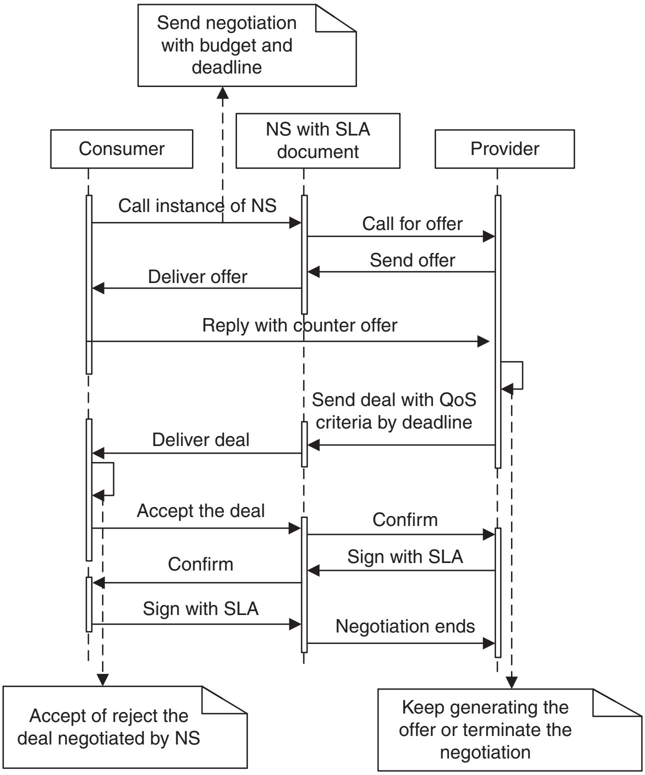


Figure 9.
SLA negotiation processes

of the agreement of the deal with expected SLA metrics if the negotiation is successful. If SLO violations occur, SLA adaptation may be implemented, or predefined penalties may be addressed.

As the service availability and load may change dynamically, SLA adaptation may be involved. To avoid the failure of the manufacturing task and the loss of task executing results, migration and renegotiate access to a new service node is allowed in the prototype. To perform the migration, rule controlled management is designed by referring to James (2006). A series of control variables and associated control actions reflecting the guarantees within the SLA are introduced. The rules are triggered when the premise is satisfied, which in turn triggers the control action leading to the resource migration.

Rule controlled migration:

```
    procedure migration_decision (Tremaining, Tschedule,……)
        if Tremaining > Tschedule then
            control_action := migrate
    else
            control_action := null
    if ……then
    ……
    …………
    return(control_action)
end procedure
```

The migration will be triggered and handled by the SLA manager once the failure of the manufacturing task happens. Thus the manufacturing task is migrated onto another node, and the SLA monitoring and verification is restarted in a similar way as discussed above.

6. Summaries and future work

Internet-based manufacturing services require cooperation and resource sharing among multiple organizations for concurrent sophisticated manufacturing modes such as VA, ASP, NM, MG, CM and so on. Due to the features of the boundary crossing, geographical dispersion, temporary, dynamic and virtual nature of these manufacturing modes, enterprises involved in manufacturing service sharing on temporary cooperative network have no opportunities to acquire prior understanding and knowledge on each other. Certain mechanisms are indispensable to aid in management of cooperation and resource sharing among multiple organizations. Particularly SLAs are employed by service providers as a means of specifying service level attributes that are provided to their partners and customers. So in virtual environment, it is necessary to develop tools and techniques over the platforms with those advanced manufacturing modes to monitor whether providers are meeting their service level obligations and to enable providers to manage their infrastructure to those agreements. The paper develops an adaptive SLA mechanism for service sharing in virtual environments. Based on the SLA framework, the whole SLA management processes including SLA contract creation, SLA negotiation, SLA monitoring and verification are introduced, and a novel SLA adaptive strategy employing an intermediate state is presented. The prototype of the adaptive SLA framework and mechanism is implemented on a cloud manufacturing platform. In this prototype, the SLA management is realized with HTTP protocol on the basis of SOA architectures. The prototype performs well to settle items with SLA mechanism for service sharing in virtual environments. However, much work still requires further efforts: the work presented in the paper is just a prototype, its performance still requires test and improvement; although we are aware of the mechanism of SLA in our platform can perform well, the interfaces for the extension of future standards still lack and still need consideration.

References

Abrishami, S., Naghibzadeh, M. and Epema, D.H.J. (2012), "Cost-driven scheduling of grid workflows using partial critical paths", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23 No. 8, pp. 1400-1414.

Al-Ali, R.J., Rana, O.F., Walker, D., Jha, W.S. and Sohail, S. (2002), "G-QoSM: grid service discovery using QoS properties", *Computing and Informatics*, Vol. 21 No. 4, pp. 363-382.

Antoine, P., Philipp, W., Oliver, W. and Wolfgang, Z. (2007), "Dynamic SLA-negotiation based on WS-agreement", CoreGRID Technical Report No. TR-0082, June 24.

Antonio, C., Giuseppe, D., Salvatore, D. and Antonio, P. (2013), "An SLA-based broker for cloud infrastructures", *Journal of Grid Computing*, Vol. 11 No. 1, pp. 1-25.

Asit, D., Heiko, L. and Giovanni, P. (2003), "Web services differentiation with service level agreements", May 1, available at: www.ibm.com/developerworks/library/ws-slafram/ (accessed April 18, 2015).

Comuzzi, M., Kotsokalis, C., Spanoudakis, G. and Yahyapour, R. (2009), "Establishing and monitoring SLAs in complex service based systems", *Proceedings of IEEE International Conference on Web Services*, Los Angeles, CA, pp. 783-790.

Cristani, M. and Cuel, R. (2004), "A survey on ontology creation methodologies", *International Journal on Semantic Web and Information Systems*, Vol. 1 No. 2, pp. 49-69.

Czajkowski, K., Foster, I., Kesselman, C., Sander, V. and Tuecke, S. (2002), "SNAP: a protocol for negotiating service level agreements and coordinating resource management in distributed systems", 8th Workshop on Job Scheduling Strategies for Parallel Processing, Edinburgh, pp. 1-13.

Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S. and Vambenepe, W. (2004), "The WS-resource framework", available at: www.globus.org/wsrf (accessed November 12, 2014).

Dan, A., David, S., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M. and Youssef, A. (2004), "Web services on demand: WSLA-driven automated management", *IBM Systems Journal*, Vol. 43 No. 1, pp. 136-158.

Don, B., David, E., Gopal, K., Andrew, L., Noah, M., Henrik, F.N., Satish, T. and Dave, W. (2000), "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 8.

Erik, C., Francisco, C., Greg, M. and Sanjiva, W. (2001), "Web Services Description Language (WSDL) 1.1", W3C Note, March 15, available at: www.w3.org/TR/wsdl/ (accessed April 12, 2015).

Foster, I., Kesselman, C. and Tuecke, S. (2001), "The anatomy of the grid: enabling scalable virtual organizations", *International Supercomputer Application*, Vol. 15 No. 3, pp. 315-320.

George, K., Andreas, M., Dimosthenis, K. and Spyridon, G. (2014), "Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms", *Future Generation Computer Systems*, Vol. 32, pp. 27-40.

James, J.P. (2006), "A management system for service level agreements in grid based systems", thesis of doctor of philosophy, University of Leeds, September.

Keller, A. and Ludwig, H. (2003), "The WSLA framework: specifying and monitoring service level agreements for web services", *Journal of Network and Systems Management*, Vol. 11 No. 1, pp. 57-81.

Mobach, D.G.A., Overeinder, B.J. and Brazier, F.M.T. (2006), "A WS-agreement based resource negotiation framework for mobile agents", *Scalable Computing: Practice and Experience*, Vol. 7 No. 1, pp. 23-36.

Ouelhadj, D., Garibaldi, J., MacLaren, J., Sakellariou, R. and Krishnakumar, K. (2005), "A multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing", *Proceedings of the 2005 European Grid Computing Conference, Amsterdam, February*, pp. 651-660.

Philip, B., Grace, A.L. and Paulo, M. (2008), "Service level agreements in service-oriented architecture environments", Technical Note No. CMU/SEI-2008-TN-021, Carnegie Mellon University.

Tao, F., Zhang, L., Venkatesh, V.C., Luo, Y.L. and Cheng, Y. (2011), "Cloud manufacturing: a computing and service-oriented manufacturing model", *Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, Vol. 225 No. 10, pp. 1969-1976.

Thomas, E. (2005), *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, NJ.

Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Snelling, D. and Vanderbilt, P. (2003), "Open grid services infrastructure (OGSI), version 1.0", technical report, Globus Grid Forum, June 23.

Venugopal, S., Buyya, R. and Winton, L. (2006), "A grid service broker for scheduling e-science applications on global data grids", *Concurrency and Computation: Practice and Experience*, Vol. 18 No. 6, pp. 685-699.

Further reading

Othman, A., Dew, P., Djemame, K. and Gourlay, I. (2003), "Adaptive grid resource brokering", *Cluster Computing*, IEEE Computer Society, Hong Kong, pp. 172-179.

Zhao, J., Goble, C., Stevens, R. and Turi, D. (2008), "Mining Taverna's semantic web of provenance", *Concurrency and Computation: Practice and Experience*, Vol. 20 No. 5, pp. 463-472.

(The Appendix follows overleaf.)

Appendix. SLA contract example of the force analysis query service

```
<?xml version="1.0"?>
<!--
SLA for the force analysis Quote Service.
-->
<SLA xmlns="http://www.ibm.com/wsla"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/wsla e:\Projects\WSLA\wsla.xsd"
name=" force analysis ServiceAgreement" >
<Parties>
      <ServiceProvider
            name=" force analysis Provider">
            <Contact>
                  <Street> Luoshi Road 218, Hongshan district</Street>
                  <City> Wuhan, Hubei, 430070, P. R. China </City>
            </Contact>
            <Action name="measurement"
                  partyName=" force analysis Provider "
                  xsi:type="WSDLSOAPActionDescriptionType">
                  <WSDLFile> measurement.wsdl</WSDLFile>
                  <SOAPBindingName>soapmeasurement </SOAPBindingName>
                  <SOAPOperationName> measurement </SOAPOperationName>
            </Action>
      </ServiceProvider>
   ……// other parties such as customers or coordinators are the same defined
   </Parties>

   <ServiceDefinition
         name=" force analysis Quoteservice">
         <Operation xsi:type="wsla:WSDLSOAPOperationDescriptionType"
               name="WSDLSOAPGetQuote">
                     <Schedule name="MainSchedule">
                           <Period>
                                 <Start>2014-10-01T14:00:00.000-05:00</Start>
                                 <End>2014-10-30T14:00:00.000-05:00</End>
                           </Period>
                     </Schedule>
                     <SLAParameter name="AverageResponseTime"
                           type="float"
                           unit="seconds">
                           <Metric>averageResponseTime</Metric>
                     </SLAParameter>
                     ……// other SLAParameters are the same defined
```

```
            <Metric name="averageResponseTime" type="double" unit="seconds">
                <Source>ms</Source>
                <Function xsi:type="wsla:Divide" resultType="double">
                    <Operand>
                    <Function xsi:type="wsla:Plus" resultType="double">
                <Operand>
            </Metric>
            ……// other SLAMetrics are the same defined
    </Operation>
</ServiceDefinition>

<Obligations>
    <ServiceLevelObjective name="g1" serviceObject="WSDLSOAPGetQuote">
        <Obliged>provider</Obliged>
        <Validity>
            <StartDate>2001-08-15:1400</StartDate>
            <EndDate>2001-09-15:1400</EndDate>
        </Validity>
        <Expression>
            <Predicate xsi:type="wsla:Less">
                <SLAParameter>AverageResponseTime</SLAParameter>
                <Value>5</Value>
            </Predicate>
        </Expression>
        <EvaluationEvent>NewValue</EvaluationEvent>
        </ServiceLevelObjective>
        </ActionGuarantee>
    </Obligations>
    ……// other SLAObligations are the same defined
    ……
    </SLA>
```

Corresponding author
Chaoyong Zhang can be contacted at: zcyhust@126.com