

SaaS 환경에서 SLA 보장을 위한 명세 및 교환 방법[†]

(A Specification and Exchange Method for Supporting SLA in SaaS Environment)

남태우^{*}
(Taewoo Nam)

강태준[§]
(Taejun Kang)

장문수^{*}
(Moonsoo Jang)

안영민[†]
(Youngmin An)

염근혁^{**}
(Keunhyuk Yeom)

요 약 클라우드 컴퓨팅 서비스를 제공하는 사업자는 이용자에게 신뢰성 있고 일관된 품질을 제공하기 위해서 SLA를 보장해야 한다. SLA(Service Level Agreement)는 서비스 사업자가 제공하는 서비스를 대상으로 가용성 등 일정한 서비스 수준을 보장하기 위해 맺는 서비스 사업자와 고객간의 계약이다. 클라우드 컴퓨팅은 다양한 클라우드 서비스의 IT 자원에 따라 IaaS, PaaS, SaaS 등으로 구분되는데 기존의 SLA는 물리적인 네트워크 환경에 대한 요소만 고려하고 있어서 제공되는 서비스의 품질 요소는 반영하기 어렵다. 본 논문에서는 SaaS 레벨에서의 SLA 명세를 위한 XML 스키마를 가지는 명세 언어와 이를 교환하기 위한 UDDI 기반의 교환 프로세스 및 아키텍처를 제안한다. 클라우드 환경에서 SaaS의 품질 요구사항은 제안한 명세 언어로 정의되고 품질 명세 저장소에 저장되며 교환 아키텍처를 기반으로 서비스 바인딩 시 교환된다.

키워드 SLA(Service Level Agreement), 서비스 수준 합의, 클라우드 컴퓨팅 서비스, SaaS

Abstract A cloud computing service provider must assure Service Level Agreement (SLA) to provide reliable and consistent quality of service to a user. The SLA is a contract between the user and the service provider that connects to assure constant level such as availability to target provided service. The cloud computing is classified into IaaS, PaaS, and SaaS according to IT resources of the various cloud service. The existing SLA is difficult to reflect quality factors of service because it only considers factors about the physical Network environment. In this paper, we suggest the UDDI-based interchange process with the architecture and the specification language having a XML schema for the SLA specification. The quality requirements of SaaS are defined by a proposed specification language in the cloud environment. It is stored in the repository of a quality specification and exchanged on during the service binding time based on the exchange architecture.

Key words SLA, Service Level Agreement, Cloud Computing Service, SaaS

[†] 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터 육성 지원사업의 연구결과로 수행되었음 (NIPA-2013-(H0301-13-1012))

^{*} 학생회원 : 부산대학교 전자전기컴퓨터공학과
kaluas@pusan.ac.kr

[§] 학생회원 : 부산대학교 전자전기컴퓨터공학과
erinlily1@pusan.ac.kr

^{*} 학생회원 : 부산대학교 전자전기컴퓨터공학과
anstn088@pusan.ac.kr

[†] 학생회원 : 부산대학교 전자전기컴퓨터공학과
yurizz91@pusan.ac.kr

^{**} 종신회원 : 부산대학교 정보컴퓨터공학부 교수
교신전자
yeom@pusan.ac.kr

1. 서론

최근 스마트폰, 태블릿PC, 스마트TV 등 모바일 기술혁신과 각종 스마트 기기의 이용 및 데이터의 급증으로 국민, 정부, 기업들은 그들의 업무환경에서 편리하게 이용하되 보다 빠르면서 저렴하게 사용할 수 있는 클라우드 컴퓨팅 서비스에 대한 관심이 급증하고 있다.

이런 클라우드 컴퓨팅 서비스를 제공하는 사업자는 이용자에게 신뢰성 있고 일관된 품질을 제공하기 위해서 클라우드 컴퓨팅 환경에서 SLA (Service Level Agreement)의 보장이 필수적이다. IT 서비스를 제공받는 고객과 서비스 공급자는 서로의 상호관계를 올바르게 인식해야 하며 각자의 요구사항 및 기대치에 대한 정의가 필요하다. 이에 대한 명확한 합의가 반드시 이루어져야 하며, 그 관리 방법의 하나가 서비스 수준 합의서 (SLA)이다.

클라우드 컴퓨팅 서비스는 사용한 만큼 비용을 지불하는 서비스이므로 SLA가 잘 정의되어야 서비스 제공자와 사용자가 만족하는 서비스를 이루어 나갈 수 있다.

미국 정부는 USG 클라우드 컴퓨팅 기술 로드맵을 수립하였으며[1], 높은 신뢰 수준의 서비스를 위한 기술적인 세부사항을 명시하고 있다. 실질적인 SLA 평가를 위해 높은 수준의 품질과 완성도가 요구되고, 보증, 수행 측정 등은 클라우드 서비스의 핵심 필요 요소이며 서비스 제공자마다 다른 SLA 기준, 자원, 보장 범위 등으로 혼란을 야기할 수 있어서 공통의 용어와 정의가 SLA에서 오해의 소지를 줄일 수 있다고 분석했다. 이를 위한 참조 모델을 만드는 과정에서 NIST에서 클라우드 SLA 연구를 수행 중이긴 하나 아직 클라우드 서비스 산업 전반에 걸친 표준 SLA는 존재하지 않는다.

클라우드 컴퓨팅은 다양한 클라우드 서비스의 IT 자원에 따라 IaaS(Infrastructure as a Service), PaaS(Platform as a Service) 및 SaaS(Software as a Service)로 구분된다[2]. IaaS는 대규모 연산 능력이 필요할 경우 확장성이 풍부한 가상화된, 즉 CPU, 메모리 등과 같은 전산 자원을 제공하거나 이미지·동영상 등의 자료를 저장할 수 있는 스토리지 자원을 제공하는 서비스를 말한다. PaaS는 사용자가 소프트웨어를 개발할 수 있는 토대를 제공하는 서비스를 말한다. 플랫폼 서비스 사업자는 애플리케이션을 개발하는데 필요한 개발환경,

프레임워크 등의 개발 플랫폼을 제공하고, 응용 서비스 개발자들은 플랫폼 서비스 사업자가 제공하는 플랫폼 상에서 IT 자원을 활용하여 새로운 애플리케이션을 만들어 사용할 수 있다. SaaS는 애플리케이션을 서비스 대상으로 하며, 응용 소프트웨어 서비스 사업자가 인터넷을 통해 소프트웨어를 제공하고, 사용자가 인터넷상에서 원격으로 접속하여 해당 소프트웨어를 활용하는 모델이다.

이와 같은 클라우드 서비스 모델 중 SaaS가 제공하는 소프트웨어 서비스는 기존 SLA의 서비스 가용성(장애 없이 서비스를 이용할 수 있는 시간의 비율) 기반의 관리 요소만으로는 제대로 그 품질에 대한 요구사항 관리가 이루어질 수 없다. 실제로 아마존, 구글, MS 등 해외 클라우드 컴퓨팅 서비스를 제공하는 기업은 SLA를 통해, 서비스 책임 범위를 명확히 하고 있으며, 일반적으로 가용성을 중심으로 품질을 보장하고 있고 제시된 가용성 수준에 미달하는 경우 고객에게 무료 서비스 형태로 위약금을 제공한다. 그러나 현재 상용화된 서비스들은 주로 IaaS나 PaaS인 스토리지나 플랫폼 서비스라서 가용성 기반의 SLA 관리가 용이하지만 SaaS의 소프트웨어가 가지는 다양한 비기능/품질 요구사항은 단순히 가용성만 가지고는 관리하기 힘들다.

따라서 본 연구를 통해 SaaS 환경에 적합한 SLA 제공을 위해 소프트웨어적인 품질 특성을 명세할 수 있는 방법을 제시하고 이를 서비스 제공자와 소비자간에 교환할 수 있는 시스템과 구조를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 본 논문과 관련된 선행 기술에 대해 살펴보고, 3장에서는 SaaS 환경에서 SLA 명세를 위한 메타모델과 명세 방법을 설명하고, 4장에서는 SLA 명세를 저장하고 교환할 수 있는 시스템을 제안한다. 그리고 5장에서는 제안한 명세 방법을 이용한 적용사례를 보이고, 마지막으로 6장에서 결론 및 향후 연구 목표를 도출한다.

2. 관련 연구

2.1 클라우드 컴퓨팅 서비스 유형

서비스 제공자가 다양한 컴퓨팅 자원들의 공유 풀(Pool)을 구축하여 사용자는 최소한의 관리 부담과 커뮤니케이션만으로 자원을 할당 받고 인터넷을 통해 쉽게 접근할 수 있도록 하는 새로운 컴퓨팅 모델인 클라우드 컴퓨팅은 제공되는 서비스 내용과 어떤 수준까지 자원 접근 권한을 제공하는가에 따라 인프라형 서비스(IaaS), 플랫폼형 서비스(PaaS), 그리고 소프트웨어형 서비스(SaaS)로 분류할 수 있다.

IaaS는 시스템 관리자의 수작업을 최소화하고 자동화된 프로세스에 의해 처리되는 특징을 가진다. PaaS는 애플리케이션 개발을 위한 개발 및 테스트 도구, DBMS, 미들웨어, API등을 제공하고, 개발된 애플리케이션을 실행하여 서비스를 제공할 수 있는 인프라와 소프트웨어 환경을 제공하는 서비스다. 그리고 SaaS는 비즈니스 애플리케이션 및 업무지원을 위한 소프트웨어를 제공하는 서비스다. 하위 인프라형에 대한 어떤 것도 관리하거나 통제할 필요가 없는 서비스로의 소프트웨어를 의미하고, IaaS기반의 인프라형을 통해 제공되는 소프트웨어 및 애플리케이션으로 사용자 별로 개별적인 개인화(Personalization)가 제공되며 독립적인 서비스 사용 환경으로 구성된다.

따라서 SaaS 환경에서 서비스 수준에 대한 협약은 보다 구체적이고 소프트웨어의 품질 특성을 보다 잘 표현할 수 있어야 한다.

2.2 SLA

SLA(Service Level Agreement)[3,4,5,6]란 서비스의 제공자와 서비스 사용자 상호간에 IT 서비스를 제공할 대상 서비스에 대해 제공자에게 최소한의 요구 수준과 측정 기준, 문제 해결 방법을 명확한 기준으로 기술하여 상호간 사전에 협의

하여 결정해둔 약속이다. 즉, SLA는 협상 과정을 통해 고객인 사용자가 사업을 수행함에 있어 직접적인 영향을 미치는 서비스가 어떤 것이며, 이러한 서비스가 어떠한 수준으로 제공되어야 하는지 언급되어야 한다. 그러므로 사전에 상호 연구, 협약을 맺을 서비스, 측정할 방법, 최소 요구 수준, 서비스 수행 결과에 대한 협의된 처벌과 보상, 문제 발생시의 대안과 해결 절차 및 방법을 넣어서 상호협의 하에 명시하게 된다. 그리하여 현실적인 목표와 서비스 수준을 측정하고 산출하여 서비스 비용 산정의 근거로 삼고 이것을 정보 시스템에 자세히 기록하여 목록을 만드는 것이 목적이다.

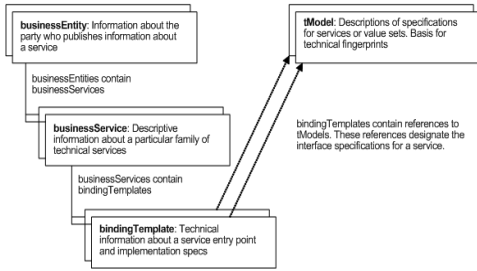
클라우드 환경에서 SLA는 고객에게 제공하는 클라우드 서비스의 품질 수준을 정량적으로 측정하고 서비스 성과를 평가, 보상하여 제공자와 사용자 간의 서비스를 보증하기 위한 품질 상세 약정이다. 현재 클라우드 서비스의 SLA로 정의되어있는 요소는 서비스 가용성, 데이터 백업 준수율 정도이다. 가용성은 예정 가동시간에 대한 실제 가동 시간의 비율이며 데이터 백업 준수율은 계획된 총 백업 건수에 대한 실제 백업 건수의 비율이다. 이런 SLA 요소로는 SaaS의 품질 특성을 반영하기 어렵다.

2.3 UDDI

UDDI(Universal Description, Discovery, and Integration)는 웹 서비스에 관한 정보를 등록, 탐색하기 위한 레지스트리로 서비스 설명과 서비스 발견에 대한 표준 스펙을 제공한다. UDDI는 각 표준과 프로토콜별로 가상의 모든 인터페이스를 사용하여 매우 유연한 서비스를 구성한다. [그림 1]은 UDDI Version 3.0.2의 표준 데이터 구조를 보인다[7].

UDDI의 각 구성요소는 XML 형태로 표현되어 저장되며, Business Entity, Business Service, Binding Template, tModel로 이루어진다. Business Entity는 제공자에 대한 정보를 담고 있으며, Business

Service는 제공자가 제공하는 웹 서비스에 대한 다양한 타입을 기술하고, Business Template은 서비스 진입점에 대한 정보를 담고 있다. 마지막으로 tModel은 웹 서비스와 어떻게 상호작용할 수 있는지에 대한 방법을 담고 있다.



[그림 1] UDDI Version 3.0.2 표준 데이터 구조

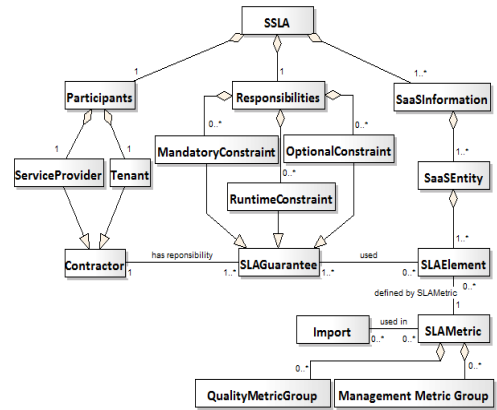
본 연구에서는 SaaS의 SLA 명세 기반 검색을 위해 tModel을 확장하여 명세 질의 처리가 가능한 SLA 명세 저장소로 이용하였다

3. SaaS SLA 명세

SaaS 환경에서 SLA 요소를 전사적으로 관리하기 위해서는 SLA 명세할 수 있는 언어가 필요하다. 본 연구에서는 이를 SSLA(SaaS Service Level Agreement)라고 명칭하고 메타모델과 XML 스키마를 제시한다.

3.1 SSLA 메타 모델

[그림 2]는 SSLA의 메타모델이다. SSLA 메타 모델은 크게 세 부분으로 구성된다. SaaS 계약에 관계된 참여자들(Participants) 부분, 하나 혹은 그 이상의 SaaS에 관련된 정보(SaaS Information) 부분, 참여자들의 의무(Responsibilities) 부분이다.



[그림 2] SSLA의 메타모델

참여자들(Participants) 부분은 SaaS 환경에서 계약을 성립하는 두 가지 종류의 참여자들을 가진다. 하나는 SaaS를 제공, 관리, 모니터링 등의 의무가 있는 서비스 제공자(Service Provider)와 SaaS를 임차해서 사용하는 임차인(Tenant)이다. 이 둘은 SLA를 이용하여 서비스를 계약하는 주체이며 이를 목적으로 계약자(Contractor)로 일반화된다.

SaaS 정보(SaaS Information) 부분은 SLA를 기반으로 서비스의 오퍼레이션, 매개변수, 메트릭의 측면에서 계약 참여자들의 공통 이해를 기술한다. SaaS 정보는 하나 혹은 그 이상의 서비스 엔티티(Service Entity)를 가지는데 이는 SaaS가 단일 서비스 개체로 제공되기도 하지만 경우에 따라 멀티테넌시 보장을 위해 복합 서비스로 구성되기 때문이다. 또한 각각의 서비스 엔티티는 SLA 요소(SLA Element)에 의해 매개변수화 되어서 서술되고 이는 SLA 보증(SLA Guarantee)에서 사용된다. SLA 요소의 매개변수가 나타내는 의미는 SLA 메트릭(SLA Metric)에 의해 정의되는데 미리 정의된 품질 항목(Predefined Metric Group)의 메트릭은 메타모델로 정의되어 있고, 계약자들의 요구에 의해 추가할 수 있는 가져오기(Import)로 정의할 수 있다.

Predefined Metric Group의 메트릭 항목, 지표, 지표 값 범위는 <표 1>과 같다. Predefined Metric Group 이외의 품질 메트릭은 계약자(Contractor)의 필요에 의해서 가져오기(Import)를 통해 추가할 수 있다.

의무(Responsibilities) 부분은 그 성격에 따라 강제적으로 어떠한 상황에서라도 지켜야 하는 강제적 제약(Mandatory Constraint), 특정 외부 상황 조건에 따라서는 지켜야 하는 선택적 제약(Optional Constraint), SaaS가 가동되는 환경인 내부적인 상황 조건에 의해 결정되는 실행 시 제약(Runtime Constraint)으로 구성된다. 의무는 SLA 요소를 사용하여 계약자들 사이에 SLA보장(SLAGuarantee)으로 일반화되어 사용된다.

3.2 SSLA의 XML 스키마

SSLA는 XML 스키마로 정의된 XML 형식 기반이며 별도의 네임스페이스(saassla)를 가진다.

<표 1> SaaS SLA의 타입 정의

```
<xsd:complexType name="SaaSslAType">
  <xsd:sequence>
    <xsd:element name="Participants"
      type="saassla:ParticipantsType"/>
    <xsd:element name="SaaSInformation"
      type="saassla:ServiceDefinitionType"
      maxOccurs="unbounded"/>
    <xsd:element name="Responsibilities"
      type="saassla:ResponsibilitiesType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="SSLA"
  type="saassla:SaaSslAType"/>
```

<표 1>은 SSLA의 타입 정의이다. SaaSslAType의 SSLA는 SaaS SLA의 루트 요소이며 SSLA는 하나의 참여자 부분, 복수개의 SaaS 정보 부분, 하나의 의무 부분으로 구성된다. SSLA의 타입 정의를 이용하여 표2와 같이 특정 SaaS의 SLA에 대하여 최상위 구조를 작성할 수 있다.

<표 2> SSLA의 최상위 구조의 예

```
<?xml version="1.0">
<saassla:SSLA
  xmlns:saassla="http://selab.pusan.ac.kr/saassla"
  name="BuyaBookSaaSslA#102">
  <Participants>
  ...
  </Participants>
  <SaaSInformation>
  ...
  </SaaSInformation>
  <Responsibilities>
  ...
  </Responsibilities>
</saassla:SSLA>
```

SSLA의 대표적인 특징인 SLA Metric 부분을 살펴보면 SLA Metric은 서비스 제공 시스템으로부터 측정되거나 다른 외부 메트릭으로부터 계산될 수 있는 서비스 속성 값에 대한 정의이다. SLA Metric은 SLA Element가 어떤 매개변수를 이용하여 어떻게 값을 측정하고 계산하는지 명시하기 위한 중요한 요소이다. SLA Metric은 complex 타입이며 메트릭 값의 형식에 대한 정보들을 찾을 수 있는 곳과 획득할 수 있는 곳 등의 모든 관련 정보를 정의한다. "Source"는 SLA Metric을 획득하고 사용하는 계약자(Contractor)에 대한 참조이며 "SLA Metric URI"는 런타임 중에 메트릭 값을 구할 수 있는 주소이다. 기술/비기술 특성에 대한 품질 요소와 외부의 참조 메트릭을 가져오기(Import) 위한 요소를 포함한다.

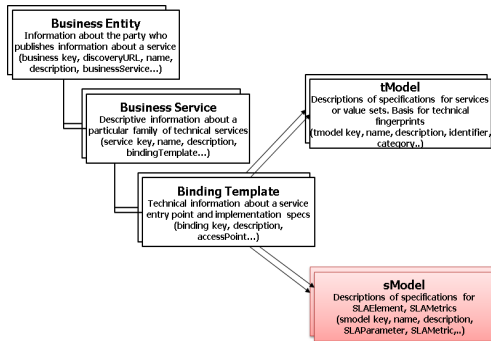
이와 같이 메타모델의 각 요소를 XML 형식으로 타입을 정의하였고 saassla라는 네임스페이스를 사용한 SSLA 정의 형식을 이용해서 SaaS의 서비스 수준 합의를 전사적으로 명세할 수 있다.

4. SSLA 명세 교환 구조

3장의 SSLA 명세 언어를 이용하여 SaaS의 SLA를 명세하였으면 이를 교환하여 계약을 맺고 서비스를 바인딩할 수 있는 아키텍처가 필요하다.

본 연구에서는 SaaS의 단위 서비스를 플랫폼 독립적으로 구동이 가능한 웹서비스 방식의 소프트웨어로 정의하고 이런 SaaS 환경에서 SLA 명세를 등록하고 검색, 바인딩이 가능한 구조를 제안한다.

기존의 WSDL(Web Service Description Language)를 교환하기 위한 용도로 사용하던 UDDI의 표준 데이터 구조를 SSLA의 등록 및 검색 질의가 가능한 구조로 확장하였다. UDDI는 웹서비스의 등록과 검색을 위해 tModel이라는 데이터 구조를 이용한다. tModel은 웹서비스의 타입, 프로토콜, 시스템 카테고리 등 서비스 제공자나 세부 서비스가 사용하는 여러 가지 기술 정보, 분류 정보, 네임스페이스 등에 대한 메타 정보를 표현하는 객체이다. 이 tModel의 데이터 구조와 유사하지만 SSLA의 SLA 요소 (SLA Element)를 처리할 수 있도록 sModel을 추가하여 SLA 명세를 등록, 검색 질의 할 수 있는 구조로 표준 UDDI를 확장하였다. [그림 3]은 UDDI 표준 데이터 구조에 sModel을 추가하여 기능 확장한 SSLA 명세 저장소의 등록 데이터 구조이다.

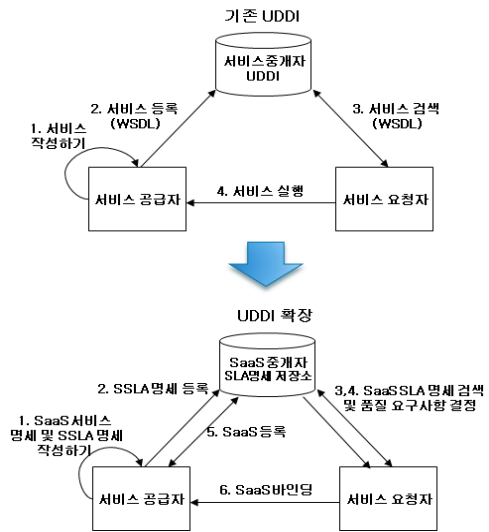


[그림 3] UDDI 표준 데이터 구조 확장

SSLA 명세 저장소는 기존의 UDDI와는 달리 SLA 품질 요소 결정 단계가 추가된다.

[그림 4]에서와 같이 서비스 공급자는 품질 요소와 제공할 수 있는 품질 메트릭이 명세된 SSLA 명세서를 SSLA 명세 저장소에 저장하면 서비스 요청

자는 sModel 기반의 서비스 검색을 통하여 특정 서비스의 품질 메트릭을 찾을 수 있고 이를 기반으로 자신이 원하는 서비스의 기능/비기능 요구사항을 결정하게 된다. 요구사항이 결정된 SSLA 명세서를 서비스 공급자가 채택하면 SLA 합의가 되며 서비스 공급자는 그에 따른 SaaS를 등록하고 서비스 요청자는 해당 서비스를 바인딩하게 된다.



[그림 4] SSLA 명세 저장소의 동작 단계

5. 사례 연구

사례 연구의 시나리오는 주식 시세를 알아보기 위한 SaaS를 서비스 제공자와 임차인이 SLA를 합의하기 위해 SSLA를 이용하여 명세를 작성한다. 서비스 제공자는 고객의 모든 요청에 대해 평균 응답 속도를 설정하며 지정한 응답 속도를 위반할 경우 계약자들에게 알림 메시지를 전송하는 것에 동의한다. <표 3>은 SLA 품질 특성으로 정의한 가용성, 성능, 관리 항목 등에 대한 SSLA 명세 작성의 간단한 예이다.

<표 3> SSLA 명세 작성 예

항목	SSLA 명세
가용성	<pre> <Metric name= "PercentOverUtilized" type= "float" unit= "percentage" > <Source>%</Source> <Schedule>BusinessDay</Schedule> <Metric>utilizationTimeSeries</Metric> <Value> <LongScalar>0.96</LongScalar> </Metric> </pre> <p>가용성 항목의 SLA Element는 Percent Over Utilized 라는 SLA Metric을 가지며 서비스 보장일의 96% 만큼의 사용 시간을 보장받아야 함</p>
성능	<pre> <Metric name= "averageResponseTime" type= "double" unit= "seconds" > <Source>ms</Source> <Metric>avgResponseTimeHost1</Metric> <Metric>avgResponseTimeHost2</Metric> </Metric> </pre> <p>성능 항목의 SLA Element는 average Response Time은 SLA Metric을 가지며 단위는 "초" 이고 Host1과 Host2 사이의 통신 시간을 ms로 측정 함</p> <pre> <Metric name= "averageResponseTimeHost1" type= "double" unit= "seconds" > <Source>ms</Source> <Function xsi:type= "saassla:Mean" resultType= "double" > <Metric>responseTimesHost1</Metric> </Metric> </pre> <p>average Response Time Host1은 response Times Host1의 평균(mean)으로 측정하며 Measurement-URI로의 응답 속도로 측정함</p>
가용성	<pre> <Action xsi:type= "WSDLActionDescription-Type" name= "Notification" contractorName= "provider" > <WSDLFile>violationNotification.wsdl </WSDLFile> <SOAPBindingName> SOAPNotificationBinding </SOAPBindingName> <SOAPOperationName>Notify </SOAPOperationName> </Action> </pre> <p>Notification.wsdl 서비스를 이용하여 장애 시간/내역을 통지 함</p>

6. 결론 및 향후 연구

SLA는 클라우드 컴퓨팅 서비스가 비즈니스 시스템으로서의 역할을 제대로 수행해 낼 수 있을 지를 객관적으로 판단할 수 있는 핵심적인 기준임에 틀림없다. 그러나 아직까지 클라우드 컴퓨팅 서비스 제공에 있어서 SLA 도입이 추세이나 이와 관련된 상세 지침이 존재하지 않을뿐더러 SaaS 환경의 SLA는 극히 초보적인 단계라서 SaaS 도입에 있어서 제약사항으로 작용하고 있다.

보다 구체적이고 체계적으로 SaaS의 SLA 명세를 위해 본 논문은 SSLA라는 XML 기반의 명세 언어를 제안하고 이를 저장하기 위한 SSLA 명세 저장소와 교환 아키텍처를 제시하였다. SSLA를 이용하여 기존의 SLA 품질 지표인 가용성이나 백업 준수를 등으로 표현하지 못하던 SaaS의 소프트웨어측면의 품질 요소를 기술할 수 있으며 SSLA 명세를 시스템적으로 관리 가능한 기반 구조를 마련하였다. 향후 SSLA 기반의 품질 컨택스트 추론 기술을 접목시켜 실시간으로 SaaS의 SLA 준수 여부를 모니터링하고 품질 추론 규칙을 통해 위반을 감지하는 관리 기술을 연구할 예정이다.

참고 문헌

- [1] NIST, "High-Priority Requirements to Further USG Agency Cloud Computing Adoption," U.S. Government Cloud Computing Technology Roadmap Volume I, 2011.
- [2] C. Gong, J. Liu, et al. "The characteristics of cloud computing," Parallel Processing Workshops (ICPPW), 2010 39th International Conference on. IEEE, pp. 275-279, 2010.
- [3] H. Ludwig, A Keller, et al. "A service level agreement language for dynamic electronic services," Electronic Commerce Research 3.1-2, pp. 43-59, 2003.
- [4] W. Maurer, R. Matlus, N. Frey., "Strategic Analysis Report, A Guide to Successful SLA Development and Management," Gartner Group, 2000.

- [5] K. Xu, X. Zhang, et al. "Research on SLA management model in Service Operation Support System," Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing, published IEEE, pp. 4912-4915, 2009.
- [6] HJ. Lee, MS. Kim, et al, "QoS Parameters to Network Performance Metrics Mapping for SLA Monitoring," The Asia-Pacific Network Operations and Management Symposium(APNOMS), pp. 42-53, 2002.
- [7] Clement, Luc, et al. "UDDI Version 3.0. 2-UDDI Spec Technical Committee Draft." OASIS UDDI Spec TC, 2004.

저자 소개



남 태 우

2007년 부산대학교 정보컴퓨터공학부 졸업(학사)
 2009년 부산대학교 컴퓨터공학과(석사)
 2009년~현재 부산대학교 전자전기컴퓨터공학과 박사과정
 관심분야: 클라우드 컴퓨팅(SaaS), 빅데이터, 상황인식 기법, 소프트웨어 아키텍처 등



강 태 준

2013년 부산대학교 정보컴퓨터공학부 졸업(학사)
 2013년~현재 부산대학교 전자전기컴퓨터공학과 석사과정
 관심분야: 클라우드 컴퓨팅, 모바일 시스템



장 문 수

2013년 동서대학교 컴퓨터정보컴퓨터공학부 졸업(학사)
 2013년~현재 부산대학교 컴퓨터공학과 석사과정
 관심분야: 클라우드 컴퓨팅(SaaS), 빅데이터, 소프트웨어 아키텍처 등



안 영 민

2010년~현재 부산대학교 정보컴퓨터공학부 학사과정
 관심분야: 클라우드 컴퓨팅, 온톨로지 추론 기법



염 근 혁

1985년 서울대학교 계산통계학과(학사)
 1992년 플로리다대학교(Univ. of Florida) 컴퓨터공학과 (공학석사)
 1995년 플로리다대학교(Univ. of Florida) 컴퓨터공학과 (공학박사)
 1985년~1988년 금성반도체 컴퓨터연구실 연구원
 1988년~1990년 금성사 정보기기연구소 주임연구원
 1995년~1996년 삼성SDS 정보기술연구소 책임연구원
 1996년~현재 부산대학교 정보컴퓨터공학부 교수
 관심분야: 소프트웨어 재사용, 프로덕트 라인 공학, 소프트웨어 아키텍처, 컴포넌트 기반 소프트웨어 개발, 적응형 소프트웨어 개발, 클라우드 컴퓨팅, 빅데이터 등