ISSN 1598-0170 (Print)
ISSN 2287-1136 (Onine)
http://www.jksii.or.kr

계산과학공학 플랫폼을 위한 실행-이력 기반의 시뮬레이션 데이터 관리 프레임워크 설계 및 구현☆

Design and Implementation of an Execution-Provenance Based Simulation Data Management Framework for Computational Science Engineering Simulation Platform

마 진[1] 이 식[1] 조 금 원[1*] 서 영 균[2*]
Jin Ma Sik Lee Kum-won Cho Young-kyoon Suh

요 약

지난 수년간 KISTI는 EDISON이라는 온라인 시뮬레이션 실행 플랫폼을 통해 사용자들이 다양한 계산과학공학 분야에서 제공된 사이언스 애플리케이션에 대한 시뮬레이션을 수행할 수 있는 서비스를 제공하고 있다. 일반적으로 이러한 시뮬레이션은 대규모 계산을 수반하므로 대용량의 출력 데이터를 생산해 낸다. 온라인 플랫폼에서 이러한 시뮬레이션을 수행 할 때 발생하는 중요한 문제 중 하나는 많은 사용자가 동일한 (또는 거의 변하지 않는) 입력 매개 변수 또는 파일을 사용하여 시뮬레이션 요청 (또는 작업)을 플랫폼에 동시에 제출함으로써 플랫폼에 상당한 부담을 준다는 점이다. 다시 말해, 동일한 컴퓨팅 작업으로 인해 중복 컴퓨팅 및 스토리지 리소스가 빠른 속도로 소모된다는 점이다. 이와 같은 동일한 시뮬레이션 요청으로 인한 과도한 자원 사용 문제를 극복하기 위해, 본 논문은 실행 메타 데이터, 즉 프로비넌스를 기반으로 시뮬레이션 데이터를 효율적으로 관리하기 위한 IceSheet라는 새로운 프레임워크를 제안한다. IceSheet 프레임워크는 시뮬레이션 실행과 관련된 프로비넌스를 수집하여 저장한다. 수집된 프로비넌스 정보는 중복 시뮬레이션 요청을 제외할 뿐만 아니라 오픈소스 검색 엔진인 ElasticSearch를 통해 기존 시뮬레이션 결과를 검색하는 데도 사용된다. 특히 본 논문은 IceSheet 프레임워크에서 저장된 시뮬레이션 결과를 검색하고 재사용할 수 있는 핵심 구성 요소에 대해 자세히 설명한다. 우리는 온라인 시뮬레이션 실행 플랫폼과 함께 연동하는 검색 엔진을 기반으로 제안된 프레임워크의 프로토타입을 구현하였다. 플랫폼에서 수집된 실제 시뮬레이션 실행 프로비넌스를 기반으로 제안된 프레임워크의 성능 평가를 수행하였다. 플랫폼과 완벽히 연동된 IceSheet 프레임워크는 사용자로 하여금 선택된 시뮬레이션 소프트웨어에 대해 과거에 입력된 매개 변수 값을 빠르게 검색하고 동일한 입력 매개 변수 값이 존재하는 경우 기존의 결과를 곧바로 반환할 수 있도록 할 것으로 기대된다. 따라서 제안된 프레임워크를 통해 이전에 실행된 시뮬레이션과 동일한 요청에 대해 중복 자원 소모를 없애고 실행 시간을 크게 단축시키는 데 도움이 될 것으로 기대한다.

☞ 주제어 : 계산과학공학플랫폼, EDISON 플랫폼, 시뮬레이션, 데이터, 검색엔진, 오픈사이언스, 프로비넌스

1 Dept. of Scientific Platform Development, Korea Institute of Science and Technology Information (KISTI), Daejeon, 34141, Korea.
2 School of Computer Science & Engineering, Kyungpook National University, Daegu, 41566, Korea.
* Corresponding author (ckw@kisti.re.kr, yksuh@knu.ac.kr)
[Received 25 September 2017, Reviewed 10 October 2017(R2 7 November 2017), Accepted 27 November 2017]
☆ 본 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 첨단사이언스·교육허브개발사업의 지원을 받아 수행된 연구임(No. NRF-2011-0020576).
☆ 본 논문은 2017년도 한국인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임.

ABSTRACT

For the past few years, KISTI has been servicing an online simulation execution platform, called EDISON, allowing users to conduct simulations on various scientific applications supplied by diverse computational science and engineering disciplines. Typically, these simulations accompany large-scale computation and accordingly produce a huge volume of output data. One critical issue arising when conducting those simulations on an online platform stems from the fact that a number of users simultaneously submit to the platform their simulation requests (or jobs) with the same (or almost unchanging) input parameters or files, resulting in charging a significant burden on the platform. In other words, the same computing jobs lead to duplicate consumption computing and storage resources at an undesirably fast pace. To overcome excessive resource usage by such identical simulation requests, in this paper we introduce a novel framework, called IceSheet, to efficiently manage simulation data based on execution metadata, that is, provenance. The IceSheet framework captures and stores each provenance associated with a conducted simulation. The collected provenance records are utilized for not only inspecting duplicate simulation requests but also performing search on existing simulation results via an open-source search engine, ElasticSearch. In particular, this paper elaborates on the core components in the IceSheet framework to support the search and reuse on the stored simulation results. We implemented as prototype the proposed framework using the engine in conjunction with the online simulation execution platform. Our evaluation of the framework was performed on the real simulation execution-provenance records collected on the platform. Once the prototyped IceSheet framework fully functions with the platform, users can quickly search for past parameter values entered into desired simulation software and receive existing results on the same input parameter values on the software if any. Therefore, we expect that the proposed framework contributes to eliminating duplicate resource consumption and significantly reducing execution time on the same requests as previously-executed simulations.

☞ keyword : Computational Science Engineering Platform, EDISON Platform, Simulation, Data, Search Engine, Open Science, Provenance

1. 서 론

한국과학기술정보연구원(KISTI) 에서 제공하는 계산과학공학 시뮬레이션 플랫폼인 EDISON (EDucation-research Integration through Simulation On the Net) [1-3]은 2017년 현재, 6개 분야(나노물리, 계산화학, 전산열유체, 구조동역학, 전산설계, 전산의학)의 계산과학공학 연구자들이 연구와 대학교 학부수업에 활용할 수 있도록 온라인 시뮬레이션 기반의 웹 포털 서비스를 제공하고 있다. 해당 시뮬레이션 플랫폼의 누적 사용자는 약 48000명 이상이고 사용가능한 시뮬레이션 SW의 수는 300개 이상이다.

그리고 플랫폼에 등록된 계산과학분야의 시뮬레이션 SW들은 실행과 관련하여 다량의 데이터를 발생시킨다. 입력데이터 및 출력데이터 그리고 많은 연산을 수행하는 과정에서 생성되는 I/O파일 및 Log정보 등의 대용량 또는 다량의 데이터들이 시뮬레이션 실행에 의해 생성된다.

그러나 이러한 다양한 분야의 사용자들이 한정된 자원을 가지고 온라인 시뮬레이션을 수행함에 따라, 자원의 분배 및 시뮬레이션 연산 속도, 스토리지 용량 확보 등의 문제가 대두되기 시작하였다. 특히, 대학교 학부수업에 활용되는 시뮬레이션 SW의 경우, 중복된 입출력 데이터를 사용하는 경우가 많아 계산과학 데이터의 검색 및 재사용의 필요성이 제기되고 있다. 또한 최근에는 과학 데이터를 공개 및 공유하여 이용자가 보다 쉽게 접근하고 활용할 수 있도록 하는 오픈사이언스가 트렌드로 자리매김하고 있다. OECD (2015), “Making Open Science a Reality” [4]에 따르면 연구 결과인 출판물, 데이터 등을 공개하면 이를 통해 연구에 대한 후속 검증과 추가 연구를 가능하게 하고 새로운 연구 방법 개발에 활용할 수 있다. 이러한 시대적 흐름과 중복 시뮬레이션 문제점을 해결하기 위해, 본 논문에서는 계산과학공학 시뮬레이션 플랫폼에 적용 가능한 데이터 검색엔진을 설계 및 개발하였다. 개발된 시스템은 중복 데이터의 저장을 방지하고 계산자원의 낭비를 최소화하여 시뮬레이션의 연산 속도를 향상시키고 시뮬레이션 데이터를 재사용할 수 있게 지원하는 계산과학 데이터 검색서비스를 위해 사용될 예정이다.

본 논문은 계산과학공학 시뮬레이션을 수행하여 실행-이력 기반의 시뮬레이션 데이터 프레임워크에 시뮬레이션 데이터를 웹으로 제공하는 EDISON(www.edison.re.kr) [5] 및 EDISON의 시뮬레이션 처리과정과 데이터모델에 대하여 설명한다. 그리고 오픈소스 프로젝트로써 검색엔진 설계 및 개발에 사용된 엘라스틱서치(Elasticsearch)와 오픈사이언스에 대한 내용을 2장 관련연구에 기술하였고, 검색엔진을 포함한 실행-이력기반 시뮬레이션 데이터 관리 프레임워크(IceSheet)의 구성 및 동작과정을 3장에서 설명하고, 검색엔진과 계산과학공학 시뮬레이션 플랫폼의 연결 방법에 대한 내용을 4장에 기술하였다. 그리고 5장에서는 실제

서비스된 데이터를 이용한 적용 및 실험 결과에 대해 평가를 수행하고, 6장은 결론과 향후 연구를 기술한다.

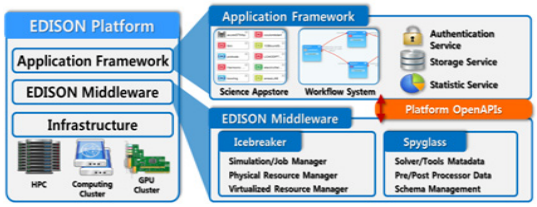
2. 관련연구

2.1 계산과학공학 시뮬레이션 플랫폼 (EDISON: EDucation-research Integration through Simulation On the Net)

한국과학기술정보연구원(KISTI)에서 서비스 중인 계산과학공학 시뮬레이션 플랫폼(EDISON : EDucation-research Integration through Simulation On the Net)은 2017년 9월 현재, 6개 분야(나노물리, 계산화학, 전산열유체, 구조동역학, 전산설계, 전산의학)의 계산과학공학 연구자들이 연구와 대학교 학부수업에 활용할 수 있도록 온라인 시뮬레이션 기반의 웹 포털 서비스를 제공 중이고 신규분야가 추가될 예정이다.

2.1.1 계산과학공학 시뮬레이션 플랫폼

웹 서비스를 제공하는 계산과학공학 시뮬레이션 플랫폼(EDISON)의 구성요소는 3가지로 구분 할 수 있으며 이를 그림 1에서 나타냈다.



(그림 1) EDISON 플랫폼 구성요소
(Figure 1) EDISON Platform Component

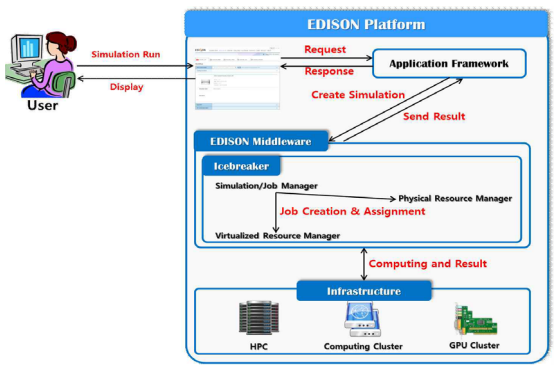
첫째, 통합 웹 포털 서비스를 목적으로 통합 유저 인증서비스, 시뮬레이션 통계 서비스, 데이터 저장소 서비스, 시뮬레이션 SW 등록 및 관리, 워크플로우 서비스를 제공하는 응용 프레임워크(Application Framework)는 라이프레이(Liferay)[6]기반으로 개발되었다.

둘째, 시뮬레이션-작업 관리와 물리-가상 자원관리, 시뮬레이션 SW의 데이터관리, 전-후 처리데이터 처리, 스키마 관리를 담당하는 EDISON 미들웨어는 JAVA와 Spring Framework [7]기반으로 개발하였다. 그리고 응용프레임워크가 사용자에게 웹을 통해 제공하는 서비스들은 EDISON 미들웨어에서 개발하여 제공하는 RESTful API [8]를 이용한다.

마지막 구성요소인 인프라스트럭처는 웹 포털 사용자들이 계산과학공학 시뮬레이션을 수행하는데 필요한 고성능 컴퓨팅 (HPC : High-Performance Computing) 자원, 계산 클러스터, CPU 클러스터 등으로 구성되어 있으며, 미들웨어와 응용 프레임워크의 개발 및 서비스가 가능하도록 구성된 시스템 자원을 포함한다.

2.1.2 계산과학공학 시뮬레이션 플랫폼의 시뮬레이션 처리과정

그림 2는 계산과학공학 시뮬레이션 플랫폼(EDISON)이 사용자로부터 시뮬레이션 수행 요청을 받아 처리하는 과정을 나타낸다[2, 17].



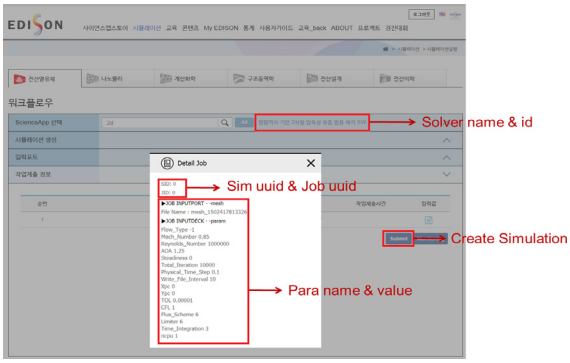
(그림 2) EDISON 플랫폼의 시뮬레이션 처리과정
(Figure 2) Simulation Process of EDISON Platform

응용 프레임워크에서 제공하는 웹 화면을 통해 사용자는 수행하고자 하는 시뮬레이션 프로그램을 선택하고 입력 값을 설정한다. 입력된 시뮬레이션은 응용 프레임워크를 통해 EDISON 미들웨어로 전달된다. 하나의 시뮬레이션은 하나 또는 다수의 작업(Job)을 생성하며, 응용 프레임워크에서 요청된 시뮬레이션은 미들웨어의 시뮬레이션-작업 관리자를 통해 시뮬레이션과 작업을 생성하고 계산 처리를 위해 인프라스트럭처의 물리-가상 자원에 할당된다. 인프라스트럭처를 통해 처리된 계산 결과는 미들웨어를 통해 응용 프레임워크에 전달되어 사용자에게 웹상에서 제공된다.

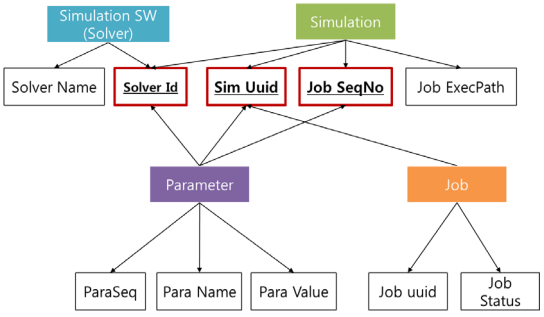
2.1.3 계산과학공학 시뮬레이션 플랫폼의 시뮬레이션 데이터 모델

2.1.2절에서 시뮬레이션을 처리하는 과정을 설명하였고,

본 절은 계산과학공학 시뮬레이션 플랫폼(EDISON)에서 시뮬레이션을 처리하는데 이용되는 데이터 모델과 관계에 대하여 설명한다. 그림 3은 EDISON 웹 사이트에서 전산열유체 분야의 '정렬격자 기반 2차원 압축성 유동 범용 해석 SW(2D_Comp_P)'시뮬레이션의 실행화면 및 그림 4의 데이터 모델과 대응하는 항목을 나타낸다.



(그림 3) EDISON의 시뮬레이션 실행화면
(Figure 3) Simulation Execution Screen of EDISON



(그림 4) EDISON 시뮬레이션 데이터 모델
(Figure 4) EDISON Simulation Data Model

그림 4는 EDISON 실행-이력 기반의 시뮬레이션 데이터 관리 프레임워크(IceSheet)를 위해 설계한 EDISON 시뮬레이션 데이터 모델이고, 시뮬레이션 처리에 사용되는 데이터 구성요소와 관계를 나타내었으며 각 요소는 다음과 같다 [9].

- 시뮬레이션 SW(또는 Solver): 사용자가 선택한 시뮬레이션 SW를 뜻하며, 실행한 시뮬레이션 해석기의 이름(Solver Name)과 식별자(Solver Id)는 시뮬레이션 데이터 저장소(Simulation Data Repository)에 저장되고 사용자가 시뮬레이션을 요청할 때 동일한 시뮬레이션 SW인지 체크하는데 사용한다.
- 시뮬레이션(Simulation): 시뮬레이션 SW를 선택하고 파라미터 값을 입력한 다음, 제출(Submit)하여 생성된 시뮬레이션을 뜻한다. 각각의 시뮬레이션은 고유한 시뮬레이션 식별자(SimUuid) 값을 생성하고, 하나의 시뮬레이션은 1개 혹은 다수의 작업(Job)을 생성하기 때문에 작업의 순서(SeqNo)를 구분하기 위해 저장하고, 해당 작업의 실행파일의 경로(Job ExecPath)를 저장한다.
- 파라미터(Parameter): 실행되는 시뮬레이션의 입력 값에 해당하며, 파라미터의 순서(Sequence)와 이름(Name), 값(Value)이 필수 요소로 저장된다.
- 작업(Job): 시뮬레이션에서 파생된 하위 요소로, 생성된 시뮬레이션의 Sim Uuid와 Job uuid 그리고 작업상태(성공, 실패, 실행 중)를 저장한다.

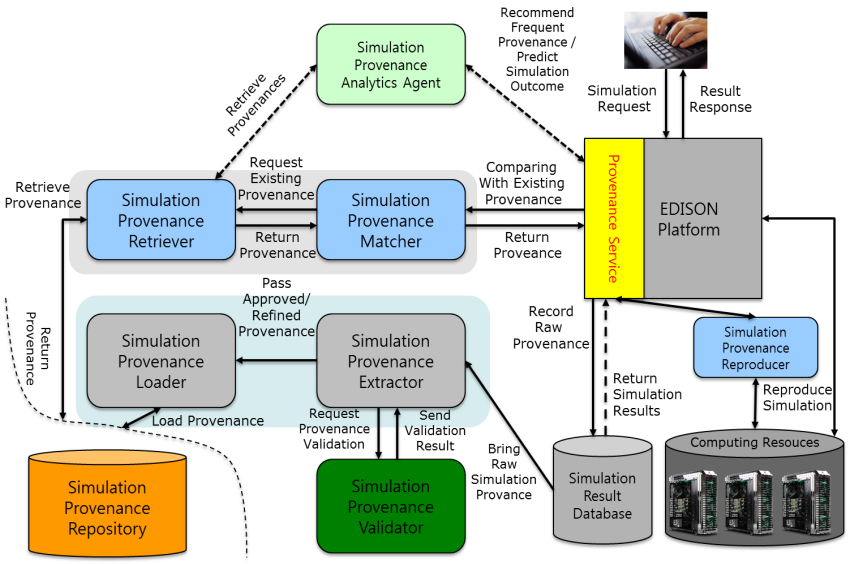
2.2 엘라스틱서치 (Elasticsearch)

Elasticsearch [10]는 오픈소스 분산 시스템이고 아파치 루씬(Apache Lucene)을 기반으로 만들어졌다. 수평적인 확장성과 안정성 및 간편한 관리를 위해 설계된 JSON [11] 문서 기반의 검색 및 분석엔진이다. RESTful API [12]기반으로 색인, 검색, 매핑, 분석, Query DSL 등의 기능을 오픈소스를 Github 다운로드 페이지 [13]에서 제공하고 있으며 Java, C#, Python, Javascript, PHP, Perl, Ruby 등의 다양한 프로그래밍 언어를 지원한다. 2017년 9월 현재 릴리즈 버전으로는 5.5.2가 제공되며, 6.0 버전은 베타 버전으로 제공되고 있다. 그리고 공개라이선스 중 하나인 Apache 2 [14] 라이선스를 따르기 때문에 다운로드 및 변경이 자유롭다.

2.3 오픈사이언스 (Open Science)

오픈사이언스는 OECD에서 2015년에 발표한 "Making Open Science a Reality" [4]에 따르면, 공공자금으로 지원된 연구 성과(출판물 및 데이터)를 디지털 포맷으로 공개하여 이용자가 보다 쉽게 접근하고 활용할 수 있도록 하여 사회 문제해결 및 기업부분에도 이익을 제공하고자 한다. 오픈사이언스는 기초과학뿐만 아니라 인문 사회를 포함한 모든 연구 활동을 대상으로 하며 개방 및 공유의 대상을 다음과 같이 6가지로 정의한다.

⊙ Open Data: 연구데이터의 개방과 공유



(그림 5) 실행-이력기반 시뮬레이션 데이터 관리 프레임워크(IceSheet) 처리과정

(Figure 5) Execution-Provenance based Simulation Data Management Framework(IceSheet) Process

- ⊙ Open Source: 하드웨어 및 소프트웨어의 코드 공개 및 자유로운 이용 허용
- ⊙ Open Methodology: 연구에 이용된 방법론 공개
- ⊙ Open Peer Review: 논문심사의 투명성을 위해 심사자의 리뷰결과 공개
- ⊙ Open Access: 연구 활동의 결과물인 논문에 대한 자유로운 접근 허용
- ⊙ Open Educational Resources: 학교에서 활용되는 교육자료의 공유

오픈사이언스를 추진하는 주요 기관은 Allen Institute for Brain Science, Center of Open Science, Publice Library of Science, Creative Commons(Science Commons), Open Knoledge Foundation 등이며 이 중 Creative Commons [15]는 오픈사이언스를 주도적으로 추진하는 기관으로 2008년부터 오픈사이언스 커먼즈(OSC: Open Science Commons) [16]를 통해 오픈사이언스 원칙을 정의하고 실행을 추진하고 있다.

3. 시스템 구성 및 동작과정

3장에서 소개하는 실행-이력기반 시뮬레이션 데이터 관리 프레임워크는 2.3절 오픈사이언스의 정의 중 Open Data 개념을 적용하고, 2.1.2절 시뮬레이션 처리과정에서 발생하는 중복 데이터 저장과 계산자원의 낭비를 최소화하기 위해 설계하였다. 설계 된 시스템은 중복 데이터의 저장을 방지하여 계산자원의 낭비를 막아주고 시뮬레이션 데이터를 재사용할 수 있는 기능을 제공하며, 이로 인해 계산자원의 연산 속도 향상을 기대할 수 있다. 본 논문에서 설계 및 개발한 검색엔진은 그림 5에서 Simulation Provenance Extractor, Validator, Loader, Retriever, Matcher에 해당한다.

3.1 실행-이력기반 시뮬레이션 데이터 관리 프레임워크 (IceSheet)

3.1.1 IceSheet 구성 및 역할

그림 5는 실행-이력기반 시뮬레이션 데이터 관리 프레임워크(IceSheet)의 구성요소 및 처리과정을 나타내었으며 각 항목은 다음과 같다.

1. **Simulation Provenance Extractor**

   EDISON 플랫폼의 시뮬레이션 결과 데이터베이스(Simulation Result Database)에서 생성된 결과를 추출하는 컴포넌트로써, 시뮬레이션에 사용된 입력 값 및 결과 파일등의 정보를 레코드(Record) 단위로 추출하여 해당 정보를 검증하기 위해 Validator로 전달한다.

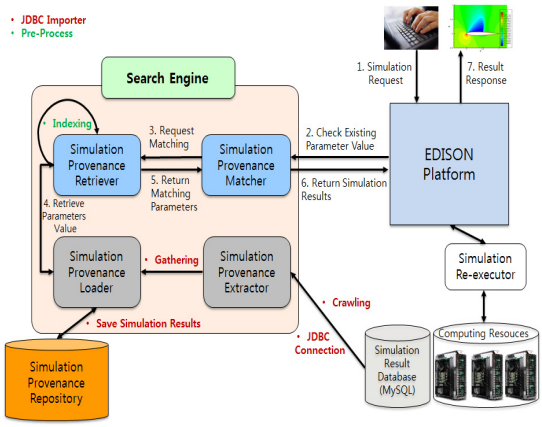
2. **Simulation Provenance Validator**
   Extractor로부터 전달받은 값을 시뮬레이션 SW의 입력 파라미터 값이 범위를 벗어나거나 잘못된 값인지 검증하는데 사용되는 컴포넌트이다.
3. **Simulation Provenance Loader**
   Loader는 Validator를 통해 검증된 레코드를 가져와서 시뮬레이션 이력 저장소(Simulation Provenance Repository)에 저장하는 역할을 수행한다.
4. **Simulation Provenance Retriever**
   본 논문의 주요 컴포넌트로써, 검색엔진의 역할을 수행한다. EDISON 서비스의 웹 UI(응용프레임워크)에서 사용자가 요청한 시뮬레이션 파라미터 정보를 수집하고 색인(Indexing)을 생성한 다음, Matcher에게 전달한다. 해당 파라미터가 존재하는지 Matcher에게 결과를 요청하고 일치하는 결과가 있다면 시뮬레이션 이력 저장소(Simulation Provenance Repository)에 저장한다.
5. **Simulation Provenance Matcher**
   본 논문의 또 다른 주요 컴포넌트로써, Retriever로부터 전달받은 파라미터 값이 존재하는지 확인하고 일치하는 결과가 있다면, 그 결과를 EDISON 플랫폼에 전달한다.
6. **Simulation Provenance Reproducer**
   Reproducer는 EDISON 플랫폼으로부터 재실행할 시뮬레이션 정보(입력 값, 결과 파일 이름 및 사이즈, 파일 경로)를 전달받아 시뮬레이션을 재실행하는 역할을 한다. 검색 결과로 전달받은 시뮬레이션 정보를 이용하면 시뮬레이션 결과의 재사용이 가능하며, 이를 통해 플랫폼에서 발생하는 중복 시뮬레이션들의 데이터 저장으로 인한 스토리지 낭비 및 시뮬레이션 처리시간의 감소가 기대된다.
7. **Simulation Provenance Analytics Agent**
   Analytics Agent는 시뮬레이션 이력 저장소(Simulation Provenance Repository)에 축적된 실행-이력 데이터를 기반으로 분석서비스를 제공하기 위한 컴포넌트다. 예를 들어, 시뮬레이션 실행-이력 저장소에 Top-K 알고리즘을 적용하여 자주 사용되는 시뮬레이션을 분석 및 추천할 수 있다. 그리고 과거 시뮬레이션의 수행 시간을 이용한 시뮬레이션 수행시간 예측서비스 또는 시뮬레이션 파라미터의 오류를 감지하여 알려주는 서비스가 제공 가능하다. 이러한 분석 서비스를 통해 사용자는 시뮬레이션을 실행하기 이전에 도움을 받을 수 있으며, 이를 통해 잘못된 시뮬레이션의 수행횟수가 감소하여 계산 자원 및 스토리지의 낭비를 줄일 수 있을 것으로 기대된다.

3.1.2 검색엔진의 동작과정

그림 6은 실행-이력기반 시뮬레이션 데이터 관리 프레임워크에서 Extractor, Loader, Retriever, Matcher로 구성된 검색엔진의 동작과정을 나타낸다. 검색엔진은 MySQL로 생성된 EDISON 플랫폼의 시뮬레이션 결과 데이터베이스(Simulation Result Database)와 JDBC 연결을 통해 데이터 크롤링(Crawling)을 하고, 크롤링을 통해 추출된 데이터를 게더링(Gathering)하여 시뮬레이션 이력 저장소(Simulation Provenance Repository)에 저장한다. 이 과정은 MySQL로 관리되던 기존의 시뮬레이션 결과를 Elasticsearch기반의 검색엔진에 적합한 JSON형태로 저장하는 과정에 해당하며 이를 위해 설정하는 JDBC Import는 4장에서 소개한다.

그리고 데이터 검색 전처리 과정으로 JSON 형태로 저장된 시뮬레이션 결과 데이터를 검색하기 위해 검색엔진은 색인(Indexing)을 미리 수행한다.

이러한 과정이 완료된 다음, 사용자가 EDISON 서비스에서 시뮬레이션을 요청할 때 수행되는 검색엔진의 동작과정은 다음과 같다. 사용자가 EDISON 웹에서 시뮬레이션 실행을 요청하면 시뮬레이션 실행을 위해 입력한 파라미터 값 또는 입력파일이 존재하는지 검색엔진의 Matcher와 Retriever을 통해 확인하고 일치하는 파라미터 값이 존재한다면 검색된 시뮬레이션 결과를 EDISON 플랫폼에 전달하여 시뮬레이션을 수행하지 않고 사용자가 결과를 확인할 수 있도록 제공한다.



(그림 6) 검색엔진 구성 및 동작과정
(Figure 6) Search Engine Configuration and Operation Process

4. 플랫폼 연동 및 적용사례

4.1 검색엔진과 계산과학공학 시뮬레이션 플랫폼의 연동

본 논문의 검색엔진은 그림 6에서 EDISON 플랫폼의 시뮬레이션 결과를 저장하고 있는 Simulation Result Database와 JDBC importer for Elasticsearch [18]의 JDBC 연결 안정성을 보장하기 위해서 Elasticsearch 2.3.3 버전의 소스코드를 사용하였다.

Elasticsearch 코드를 적용한 서버에 JDBC importer를 다운로드 받은 다음, 표 1과 같이 Shell Script파일을 작성한다. Shell Script파일은 Elasticsearch와 JDBC importer를 연결하기 위한 환경 및 변수를 선언한다. JDBC를 사용하고 URL은 “jdbc:mysql://EDISON Simulation Result Database Address:Port/Simulation Result Database TableName”, Port는 3306을 사용한다.

(표 1) JDBC importer for Elasticsearch 설정
(Table 1) JDBC importer for Elasticsearch Configuration

```
#!/bin/bash
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
bin=${DIR}/../bin
lib=${DIR}/../lib
echo ' {
"type" : "jdbc",
"jdbc" : { "url" :
"jdbc:mysql://150.***.***.***:3306/edison_TableName",
"user" : "userName",
"password" : "password",
"sql": "select sim.groupid, sov.scienceappid, sov.name,
sov.version, sim.simulationuuid, sj.jobuuid,
TO_SECONDS(sj.jobEndDt)-TO_SECONDS(sj.jobStartDt)
as jobElapsedTime, jobdata from EDAPP_ScienceApp
sov, EDSIM_Simulation sim, EDSIM_SimulationJob sj,
EDSIM_SimulationJobData sjd where sov.scienceappid
= sim.scienceappid and sim.simulationuuid =
sj.simulationuuid and sj.jobuuid = sjd.jobuuid order by
groupid, name, version, simulationuuid, jobuuid;",
"treat_binary_as_string" : true,
"elasticsearch" : {
"cluster" : "prov01",
"host" : "150.***.***.**", //Elastic code가 설치된 서버
"port" : 9300   },
"max_bulk_actions" : 20000,
"max_concurrent_bulk_requests" : 10,
"index" : "test"}
} ' | java \
-cp "${lib}/*" \
-Dlog4j.configurationFile=${bin}/log4j2.xml \
org.xbib.tools.Runner \
org.xbib.tools.JDBCImporter
```

EDISON DB에는 여러 Table이 존재하는데 예를 들면 ‘edison_resultDB’ 또는 ‘edison_simDB’와 같이 시뮬레이션 결과 정보를 관리하는 Table명을 “jdbc:mysql://DB주소:port번호/” 다음에 입력한다.

sql은 연결된 시뮬레이션 결과 정보 테이블에서 추출하고자 하는 데이터의 컬럼명을 질의로 설정한다. sql 질의문 다음 정보인 “elasticsearch” 항목의 host와 cluster는 Elasticsearch가 적용된 서버의 주소 및 이름, port는 Elasticsearch의 기본 포트인 9300을 입력한다. Index는 Elasticsearch의 검색에 사용할 색인 이름을 입력하는데, 입력된 색인 이름은 JDBC연결을 통해 그림 5에서 Extractor와 Validation, Load 과정을 거쳐 추출된 시뮬레이션 데이터를 검색하는데 사용된다.

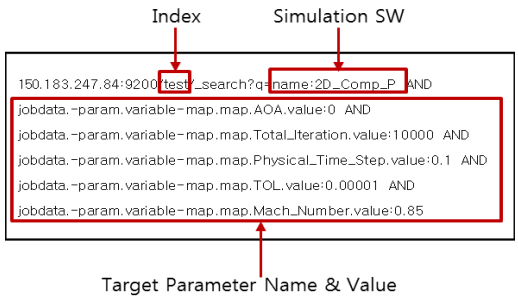
4.2 검색 적용 사례

표 2는 개발한 검색엔진을 테스트하기 위해 현재 전산열유체(CFD) 분야에서 제공되고 있는 “정렬격자 기반 2차원 압축성 유동범용해석 SW(2D_Comp_P)”의 데이터 검색에 사용한 파라미터 이름과 의미를 기술하였다.

(표 2) 전산열유체 분야의 정렬격자 기반 2차원 압축성 유동 범용 해석 SW (2D_Comp_P) 데이터 검색에 사용한 파라미터 목록
(Table 2) List of parameters used for data retrieval Compressible Euler/Navier-Stokes Flow Solve(2D_Comp_p) for Computational Fluid Dynamics

파라미터	의 미
AOA	받음각
Total_Iteration	최대 반복 계산 횟수
Physical_Time_Step	시간 전진 간격
TOL	허용 오차
Mach_Number	마하수

본 논문에서 논의된 검색엔진은 RESTful API 기반의 검색 메서드와 Query DSL을 이용하여 시뮬레이션 SW의 입력 데이터 검색을 요청할 수 있다. 그림 7은 개발한 검색엔진을 사용하여 전산열유체(CFD) 분야에 등록되어 있는 시뮬레이션SW인 “정렬격자 기반 2차원 압축성 유동범용해석 SW(2D_Comp_P)”를 실행한 시뮬레이션 중 일치하는 파라미터 값의 검색을 요청하는 GET 메서드 사용법을 나타낸다.



(그림 7) 2D_Comp_P 시뮬레이션의 데이터 검색 API 및 Query DSL

(Figure 7) 2D_Comp_P Simulation of Data Search API and Query DSL

5. 실험 결과 및 평가

본 논문의 실험은 2017년 1월부터 8월까지(1/1~8/25) EDISON 서비스를 이용하여 발생 된 시뮬레이션 결과 141,166건을 대상으로 개발한 검색 API와 QueryDSL를 통해 검색에 소요되는 시간을 측정하였다. 실험에 사용된 시뮬레이션 SW는 EDISON 서비스의 전산열유체 분야와 나노물리 분야에서 중복 파라미터를 많이 이용하거나, 잦은 I/O 작업으로 인해 계산자원의 속도저하에 영향을 미치는 시뮬레이션 SW를 선정하였다.

실험에 사용된 시뮬레이션 SW는 전산열유체 4종(KFLOW, 3D_Comp, 2D_Incomp, 2D_Comp)과 나노물리 2종(gravityslingshot, Wave Simulation)이며, 표 3에는 각 시뮬레이션 SW의 총 실행 횟수(Execution Count)와 평균 작업 완료시간(Avg. Job Elapsed Time), 그리고 검색엔진을 사용한 평균 검색 완료시간(Avg. Search Engine Took Time)을 기록하였다. 검색엔진의 실험방법은 시뮬레이션 SW마다 각각 20번씩 수행하였으며, 검색을 요청하는 파타미터 값은 계속 변경하여 실험을 진행하였다.

(표 3) 시뮬레이션 데이터 검색 실험 결과

(Table 3) Simulation Data Search Experiment Results

시뮬레이션SW	Execution Count (실행수)	Avg. Job Elapsed Time(초)	Avg. Search Engine Took Time(초)
KFLOW	790	533	0.01495
3D_Comp	223	763	0.00486
2D_Incomp	24573	732	0.00409
2D_Comp	17220	484	0.00591
gravityslingshot	19461	124	0.00562
Wave Simulation	28699	5	0.00424

표 3을 통해 실험에 사용된 모든 시뮬레이션SW의 평균 작업 완료시간(Avg. Job Elapsed Time)보다 평균 검색완료시간(Avg. Search Engine Took Time)이 최소 4초에서 최대 760초 이상 적게 소요되는 것을 확인 할 수 있다. 실험 결과의 평균 검색완료 시간이 평균 0.1초 미만이기 때문에 검색엔진을 통해 최소 333배 이상의 성능향상을 기대할 수 있다. 하지만 해당 실험의 평균 작업완료시간은 EDISON 서비스에서 시뮬레이션 실행 요청부터 웹페이지에 결과가 전달되는 모든 과정이 포함된 시간이고, 평균 검색완료시간은 검색엔진의 검색 완료시간을 측정한 것이기 때문에 본 논문에서 개발한 검색엔진과 EDISON 플랫폼의 응용 프레임워크(웹 UI)가 연동하여 처리하는 시간은 제외되었다.

그러나 Wave Simulation을 제외한 다른 시뮬레이션의 평균 작업 완료시간과 평균 검색 완료시간의 차이가 최소 100초 이상 이기 때문에 응용 프레임워크와의 데이터 처리 시간이 추가되어도 기존의 시뮬레이션 수행 시간보다 많은 시간단축이 일어날 것으로 기대된다.

6. 결론 및 향후 연구

본 논문은 시뮬레이션 플랫폼에서 생성되는 중복 데이터의 저장을 방지하고 계산자원의 낭비를 최소화하여 시뮬레이션의 연산 속도의 향상과 데이터 재사용을 지원하는 실행-이력 기반의 시뮬레이션 데이터 관리 프레임워크(IceSheet)를 설계하였다. 그리고 실행-이력 기반 데이터 관리 프레임워크의 핵심 요소인 검색엔진을 설계 및 개발하여 중복 파라미터를 많이 이용하거나 잦은 I/O작업으로 인해 계산자원의 연산 속도에 영향을 미치는 시뮬레이션을 대상으로 실험을 진행하였다. 실험을 통해 최소 333배의 성능향상을 기대할 수 있다.

그러나 이번 실험에서는 검색 완료시간만을 측정하였기 때문에 향후에는 계산과학공학 시뮬레이션 플랫폼의 응용프레임워크와 연계하여 검색서비스를 GUI로 확장하여 실험을 진행한 뒤 서비스를 제공할 예정이다.

또한 현재 검색엔진과 기존 시뮬레이션 Database를 연결하고 데이터를 수집하는 과정을 사전작업으로 진행하여 실시간이 수집이 가능한 시스템이 아니므로, 향후에 실시간 또는 특정시간에 데이터 수집이 가능하도록 하는 연구가 필요하다.

참고문헌(Reference)

[1] Jin Ma, Jerry Seo, Jong Suk Ruth Lee and Minjae Park, “Implementation and Application of the EDISON platform’s integrated file management service,” Journal of Internet Computing and Services (JICS), Vol.17, No.6, pp.71-79, 2016.
http://dx.doi.org/10.7472/jksii.2016.17.6.71

[2] Jin Ma, Jongsuk Ruth Lee, Kumwon Cho and Minjae Park, “Design and Implementation of Information Management Tools for the EDISON Open Platform,” KSII Transactions on Internet and Information Systems, Vol. 11, No. 2, pp. 1089-1104, 2017.
https://doi.org/10.3837/tiis.2017.02.026

[3] Young-kyoon Suh, Kum won Cho, “Construction and Service of a Web-based Cyber-learning Platform for the Computational Science and Engineering Community in Korea”, Journal of Internet Computing and Services (JICS), Vol.17,No.4, pp.115-125, 2016.
http://doi.org/10.7472/jksii.2016.17.4.115

[4] OECD, “Making Open Science a Reality”, Oct, 15, 2015. http://www.oecd-ilibrary.org/science-and-technology/oecd-science-technology-and-industry-policy-papers_23074957

[5] EDISON(EDucation-research Integration through Simulation On the Net), http://edison.re.kr

[6] Liferay, https://www.liferay.com/

[7] Spring, https://projects.spring.io/spring-framework/

[8] Fielding, Roy Thomas, “Chapter 5: Representational State Transfer (REST),” Architectural Styles and the Design of Network-based Software Architectures (Ph.D.), University of California, Irvine, 2000.

[9] Jin Ma, Young-Kyoon Suh, Jong-Suk Ruth Lee, “Design of Data Model for Execution-Provenance Management in an Online HPC Simulation Platform”, in Proc. of KSII Fall Conference, Vol.17, No.2, pp.153-154, 2016.

[10] Elasticsearch,
https://www.elastic.co/products/elasticsearch

[11] JSON, https://json.org

[12] Fielding, Roy Thomas, Richard N. Taylor, “Principled Design of the Modern Web Architecture,” ACM Transactions on Internet Technology, Vol. 2, No. 2, May 2002, pp.115 – 150, ISSN 1533-5399, 2002.
http://dx.doi.org/10.1145/514183.514185

[13] Github-elastic, https://github.com/elastic/elasticsearch

[14] Apache License Version 2.0, January, 2004.
http://www.apache.org/licenses/LICENSE-2.0.html

[15] Creative Commons(CC),
https://creativecommons.org/about/program-areas/open-science/

[16] Open Science Commons(OSC),
https://www.opensciencecommons.org/

[17] Jin Ma, Young-Kyoon Suh, “Design and Development of Data Search Engine for Computational Science Engineering Simulation Platform”, in Proc. of KSII Spring Conference, pp.87-88, 2017.

[18] Java Database Connection(JDBC)importer for Elasticsearch, https://github.com/jprante/elasticsearch-jdbc

◐ 저 자 소 개 ◑

마 진 (Jin Ma)
2010년 광운대학교 컴퓨터소프트웨어학과 졸업(학사)
2012년 광운대학교 대학원 컴퓨터과학과 졸업(석사)
2012년~2015년 (주)비스텔 선임연구원
2015년~현재 한국과학기술정보연구원(KISTI) 계산과학공학센터 연구원
관심분야 : 데이터 통합, 빅 데이터, 분석시스템, 분산컴퓨팅, 정보검색
E-mail : majin@kisti.re.kr

이 식 (Sik Lee)
1989년 서울대학교 화학과 졸업(학사)
1993년 포항공과대학교 화학과 졸업(석사)
1996년 포항공과대학교 화학과 졸업(박사)
2000년~현재 한국과학기술정보연구원 (KISTI) 융합연구플랫폼개발실 책임연구원 (실장)
관심분야 : 계산과학, 바이오 인포매틱스, 오픈 사이언스, 이공계 교육·연구 융합
E-mail : siklee@kisti.re.kr

조 금 원 (Kum Won Cho)
1993년 인하대학교 항공우주공학과 졸업(학사)
1995년 KAIST 항공우주공학과 졸업(석사)
2000년 KAIST 항공우주공학과 졸업(박사)
2000년~현재 한국과학기술정보연구원 (KISTI) 계산과학공학센터 책임연구원 (센터장)
관심분야 : 계산과학, 항공우주, 유체해석, 이공계 교육·연구 융합
E-mail : ckw@kisti.re.kr

서 영 균 (Young-Kyoon Suh)
2003년 경북대학교 컴퓨터과학과 졸업 (학사)
2005년 KAIST 전자전산학과 전산학 전공 졸업(석사)
2015년 Dept. of Computer Science, Univ. of Arizona 졸업(박사)
2005년~2017년 한국과학기술정보연구원 (KISTI) 계산과학공학센터 선임연구원
2017년~현재 경북대학교 컴퓨터학부 교수
관심분야 : 데이터베이스 시스템/설계, 컴퓨팅의 과학, 빅데이터, HPC, 소프트웨어 테스팅
E-mail : yksuh@knu.ac.kr