

---

# 계산과학 시뮬레이션을 위한 HPC 작업 관리 프레임워크의 설계 및 구현

유정록\* · 김한기\* · 변희정†

\*한국과학기술정보연구원 †수원대학교

## Design and Implementation of HPC Job Management Framework for Computational Scientific Simulation

Jung-Lok Yu\* · Han-Gi Kim\* · Hee-Jung Byun†

\*Korea Institute of Science and Technology Information † Suwon University

E-mail : {junglok.yu, hgkim}@kisti.re.kr † heejungbyun@suwon.ac.kr

### 요 약

최근, 슈퍼컴퓨터 시스템은 교육, 의료, 국방 등은 물론 계산과학 시뮬레이션까지 다양한 분야에서 활용되고 있다. 특히, 슈퍼컴퓨터, 클라우드 등 이종의 계산 자원을 연동하는 시뮬레이션 환경은 시간, 공간적 한계를 극복하는 차세대 연구 환경으로서 다양한 응용 분야에서 큰 각광을 받고 있다. 그러나 기존의 이종 슈퍼컴퓨팅 환경은 API(Application Programming Interface)의 부재로 터미널 접근을 통해 계산 작업을 수행하는 단조로운 형태를 띠고 있으며, 이는 이종 컴퓨팅 자원의 접근성 및 활용성을 저해하는 주요 요소이다. 이러한 문제점을 해결하고, 슈퍼컴퓨팅 서비스의 다양성을 확보하기 위해, 본 논문에서는 계산과학 시뮬레이션을 위한 웹 기반 HPC (High Performance Computing) 작업 관리 프레임워크의 설계 및 구현에 대해 기술한다. 제안한 프레임워크는 슈퍼컴퓨터, 클라우드, 그리드 등의 다양한 이기종 자원을 활용하여 시뮬레이션 계산 작업을 수행할 수 있도록 플러그-인 기반 표준 인터페이스 및 번들 플러그-인을 제공하며, 플러그-인 관리자를 통해 손쉽게 자원 확장이 가능하다. 또한 시뮬레이션 작업의 생성, 제출, 제어, 상태 모니터링 등의 전주기적인 작업 라이프사이클 관리를 위해 HTTP 표준 방식의 RESTful endpoints들을 제공한다.

### ABSTRACT

Recently, supercomputer has been increasingly adopted as a computing environment for scientific simulation as well as education, healthcare and national defence. Especially, supercomputing system with heterogeneous computing resources is gaining resurgence of interest as a next-generation problem solving environment, allowing theoretical and/or experimental research in various fields to be free of time and spatial limits. However, traditional supercomputing services have only been handled through a simple form of command-line based console, which leads to the critical limit of accessibility and usability of heterogeneous computing resources. To address this problem, in this paper, we provide the design and implementation of web-based HPC (High Performance Computing) job management framework for computational scientific simulation. The proposed framework has highly extensible design principles, providing the abstraction interfaces of job scheduler (as well as bundle scheduler plug-ins for LoadLeveler, Sun Grid Engine, OpenPBS scheduler) in order to easily incorporate the broad spectrum of heterogeneous computing resources such as cluster, computing cloud and grid. We also present the detailed specification of HTTP standard based RESTful endpoints, which manage simulation job's life-cycles such as job creation, submission, control and status monitoring, etc., enabling various 3rd-party applications to be newly created on top of the proposed framework.

### 키워드

Supercomputer, Scientific simulation, Job management, Web-based framework, Implementation

## 1. 서론

최근 마이크로프로세서와 네트워크의 비약적인 발전으로 슈퍼컴퓨터 시스템은 교육, 의료, 국방 등은 물론 계산과학 시뮬레이션까지 다양한 분야에서 활용되고 있다. 특히, 슈퍼컴퓨터, 클라우드 등 이종의 계산 자원을 연동하는 시뮬레이션 환경은 시간, 공간적 한계를 극복하는 차세대 연구 환경으로서 다양한 응용 분야에서 큰 각광을 받고 있다. [1,2]. 그러나 기존의 이종 슈퍼컴퓨팅 환경은 API(Application Programming Interface)의 부재로 터미널 접근을 통해 계산 작업을 수행하는 단조로운 형태를 띠고 있으며, 이는 이종 컴퓨팅 자원의 접근성 및 활용성을 저해하는 주요 요소이다.

앞서 기술한 바와 같이, 이에 대한 근본적인 원인은 이종의 컴퓨팅 자원에 접근하여 계산 작업을 수행하기 위해 필요한 공개된 API의 부재 때문이다. 미국 선진 슈퍼컴퓨팅센터에서는 시뮬레이션 및 IT 기술 경쟁력 강화를 위해 고도화된 사이언스 게이트웨이 플랫폼 확보를 위해 노력하고 있으며, 대표적인 결과물로는 AGAVE (A Grid And Virtualization Environment) [3], NEWT [4] 등이 있다. 이들 모두 대규모의 슈퍼컴퓨팅 자원들을 효율적으로 활용하기 위해 자체 개발된 API들을 제공하고 있다.

셸 터미널 위주의 단조로운 사용자 환경을 다양하고 슈퍼컴퓨팅 서비스의 다양성을 확보, 지원하기 위해, 본 논문에서는 계산과학 시뮬레이션을 위한 HPC (High Performance Computing) 작업 관리 프레임워크의 설계 및 구현에 대해 기술한다. 제안한 HPC 작업 관리 프레임워크는 슈퍼컴퓨터, 클라우드, 그리드 컴퓨팅 등의 다양한 이기종 자원과 연계하여 시뮬레이션 작업을 수행할 수 있도록 플러그-인 기반의 작업 제어 모듈 인터페이스를 제공하며, LoadLeveler, SGE, PBS 등의 작업 스케줄러를 지원하는 번들 플러그-인을 제공한다. 모듈 형태로 제작이 가능한 표준화된 플러그-인 인터페이스의 적용으로 손쉽게 자원 연계 및 확장이 가능하다. 또한 시뮬레이션 작업의 생성, 제출, 제어, 상태 모니터링 등의 제반 기능에 대해 HTTP 표준 방식의 RESTful endpoints [5]를 제공한다.

논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안한 계산과학 시뮬레이션 위한 HPC 작업 관리 프레임워크의 설계와 그 구조에 대해 설명한다. 3장에서는 구현 상세내용을 기술하고, 4장에서는 제공하는 API들을 설명한다. 마지막으로 5장에서 결론 및 향후 연구에 대해 기술한다.

## II. HPC 작업 관리 프레임워크 설계 및 구조

HPC 작업 관리 프레임워크의 목적은 이기종의 고성능 컴퓨팅 자원들을 연동하여 효율적인 시뮬레이션 작업 수행을 가능하게 하고, 이를 위한 웹 표준 기반의 OpenAPI를 제공하는 것이다. 이를 위한 HPC 작업 관리 프레임워크 설계 요구사항은 다음과 같다.

- 다양한 분야의 응용을 위해 개발된 시뮬레이션 해석도구의 수행 및 관리가 가능하도록 범용성을 지녀야 한다.
- 시뮬레이션 해석도구가 실행되는 컴퓨팅 자원에 대한 중립성을 가져야 하며, 새로운 컴퓨팅 자원 환경과도 연동될 수 있도록 추상화된 작업 제어 모듈 인터페이스 제공을 통해 확장성을 가져야 한다.
- 웹 포털, 어플리케이션, 모바일 등의 다양한 클라이언트를 수용할 수 있도록 개방성을 가진 웹 표준 기반의 인터페이스를 제공해야 한다.

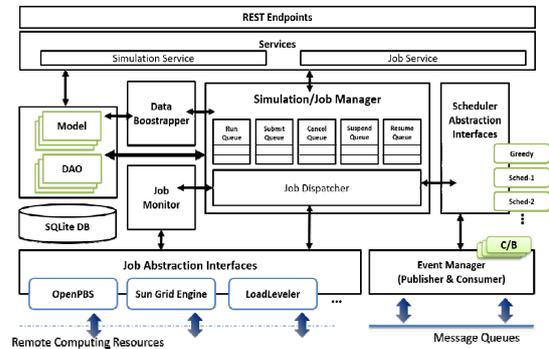


그림 1. 계산과학 시뮬레이션을 위한 HPC 작업 관리 프레임워크 구조도

그림 1은 계산과학 시뮬레이션을 위한 HPC 작업 관리 프레임워크의 전체 구조도이다. 각 주요 컴포넌트별로 그 역할 및 기능은 다음과 같다.

- 이벤트 매니저: 다른 주요 프레임워크 모듈 (예, 인증 관리자, 자원 상태정보 관리자)들과의 통신을 담당한다. 메시지-큐 기반으로 동작하며 다른 프레임워크 모듈들에게 원격 프로시저 호출 서비스를 제공한다.
- 데이터 및 모델 접근 객체: 작업 관리를 위해 프레임워크 내부의 데이터베이스 스키마이다. 데이터베이스 객체의 추가/삭제/갱신/검색 기능을 제공한다.
- 시뮬레이션 및 작업 매니저 : 시뮬레이션 생성 및 작업 제출, 작업 제어, 작업 상태 모니터링의 요구가 있을 때, 작업의 라이프사이클을 관리한다. 계산 자원에서의 작업 제출 및 제어는 작업 추상화 인터페이스 규격에 맞춰 작성된 플러그-인 모듈을 사용하여 수행된다
- 작업 제어 추상화 인터페이스 매니저 : 작업 제출(submit), 작업 제어(cancel, suspend,

resume), 작업 상태정보 모니터링(monitor) 인터페이스를 정의하며, 플러그-인 모듈들을 관리한다.

- 스케줄러 추상화 인터페이스 : 효율적인 자원 활용을 위해 작업 메타 스케줄러에 대한 인터페이스를 정의하며, 다양한 스케줄링 알고리즘을 지원한다.
- 서비스 및 RESTful Endpoints : 웹 표준 기반의 HTTP(S) 인터페이스를 제공한다.

### III. 상세 구현

본 논문에서 제안한 HPC 작업 관리 프레임워크는 파이썬 기반의 피라미드(Pyramid) 웹 프레임워크[6]를 사용하여 개발되었다. 피라미드는 모델-뷰-컨트롤러 모델을 사용하기 때문에 경량의 확장성 있는 웹 서비스 개발이 용이하며, JSON/XML과 파이썬 객체간의 상호 매핑, 파이썬 객체 및 데이터베이스 엔트리간의 상호 매핑 등의 고수준 기능들을 제공한다.

HPC 작업 관리 프레임워크 내부에서 공통적으로 사용되는 시뮬레이션 서비스, 작업 서비스, 데이터 접근 객체, 작업 스케줄러 별 번들 플러그인들은 메모리 소비를 최소화하고 플러그인-인 관리의 편의성을 위해 싱글톤ingleton) 디자인 패턴에 따라 구현되었으며, 시뮬레이션/작업 관리자 및 이벤트 관리자는 다중 쓰레드를 사용하여 구현되었다. 이벤트 관리자의 메시지-큐는 RabbitMQ [7]를 사용하였다.

시뮬레이션은 작업의 그룹을 관리하는 개념으로 다수개의 작업이 하나의 시뮬레이션을 구성하는 1:N 매핑을 가질 수 있다. 이는 파라미터 스위프(parameter sweep)과 같은 형태의 일련의 시뮬레이션 작업들을 처리하기 위함이다. 시뮬레이션 및 작업 데이터 관리는 고수준의 SQLAlchemy[8]를 사용하였으며, 그림 2와 그림 3은 각각 시뮬레이션 및 작업 데이터베이스 스키마를 보여준다.

```
class Simulation(Base):
    __tablename__ = "simulations"
    id = Column(String, primary_key=True)
    title = Column(String, nullable=False)
    description = Column(String, default='')
    userid = Column(String, default='')
    lastmodified = Column(DateTime)
    props =
    Column(MutableDict.as_mutable(JSONEncodedDict))
    jobs = relationship("Job", backref='simulations',
    lazy='dynamic', cascade='all, delete, delete-orphan')
```

그림 2. 시뮬레이션 데이터베이스 스키마

그림4는 작업 관리 프레임워크에서 정의한 표준화된 작업 제어 추상화 인터페이스이다. 이러한 표준화된 추상화 인터페이스 제공으로, 작업 제어

```
class Job(Base):
    __tablename__ = "jobs"
    id = Column(String, primary_key=True)
    title = Column(String, nullable=False)
    description = Column(String, default='')
    userid = Column(String, nullable=False)
    executable = Column(String, nullable=False)
    args = Column(String)
    category = Column(PickleType)
    links = Column(PickleType)
    cluster =
    Column(MutableDict.as_mutable(JSONEncodedDict))
    lastmodified = Column(DateTime)
    props =
    Column(MutableDict.as_mutable(JSONEncodedDict))
    state = Column(Enum(*JobState._asdict().values(),
    name='job_states'))
    jobid = Column(String)
    submittedtime = Column(DateTime)
    starttime = Column(DateTime)
    endtime = Column(DateTime)
    localstate = Column(String)
    workingDir = Column(String)
    req =
    Column(Enum(*RequestJobState._asdict().values(),
    name='req_states'))
    simid = Column(String,
    ForeignKey('simulations.id'))
```

그림 3. 작업 데이터베이스 스키마

플러그-인 [9]이 모듈화된 형태로 제작이 가능하며, 이중의 작업 스케줄러와 연동하여 작업의 제출, 제어 및 상태정보 모니터링이 가능하다.

```
class IJobControllerPlugin(object):
    storageRoot = None
    def setStorageRoot(self, storageRoot):
        self.storageRoot = storageRoot
    @abc.abstractmethod
    def submit(self, job):
        log.debug("TEST")
    @abc.abstractmethod
    def monitor(self, job):
        pass
    @abc.abstractmethod
    def cancel(self, job):
        pass
    @abc.abstractmethod
    def suspend(self, job):
        pass
    @abc.abstractmethod
    def resume(self, job):
        pass
    @abc.abstractmethod
    def getResult(self, job):
        pass
    @abc.abstractmethod
    def getLogs(self, job):
        pass
```

그림 4. 작업 제어 추상화 인터페이스

#### IV. RESTful Endpoints 규격

HPC 작업 관리 프레임워크에서 제공하는 REST 기반의 인터페이스는 그림 5와 같다. 시뮬레이션 생성, 삭제, 정보 확인 기능을 제공하며, 시뮬레이션 내의 작업의 제출, 제어, 상태 확인 등의 작업 실행을 위한 OpenAPI들을 제공한다. 그림 6은 구현된 OpenAPI를 사용하여 시뮬레이션 생성 및 작업 제출을 수행하는 예시이다.

	Endpoint URL	의미
POST	/api/{userid}/simulation/create	시뮬레이션 생성
GET	/api/{userid}/simulation/list	시뮬레이션 리스트 요청
GET	/api/{userid}/simulation/{simid}/info	시뮬레이션 정보 요청
PUT	/api/{userid}/simulation/{simid}	시뮬레이션 정보 수정
DELETE	/api/{userid}/simulation/{simid}	시뮬레이션 삭제
POST	/api/{userid}/simulation/{simid}/job/submit	작업 생성 및 제출
GET	/api/{userid}/simulation/{simid}/job/list	시뮬레이션에 속한 작업 리스트 요청
GET	/api/{userid}/simulation/{simid}/job/{jobid}/status	작업 상태정보 요청
PUT	/api/{userid}/simulation/{simid}/job/{jobid}/cancel	작업 취소
PUT	/api/{userid}/simulation/{simid}/job/{jobid}/suspend	작업 일시정지
PUT	/api/{userid}/simulation/{simid}/job/{jobid}/resume	작업 실행 재개
DELETE	/api/{userid}/simulation/{simid}/job/{jobid}	작업 삭제
GET	/api/{userid}/job/{jobid}/result	작업 결과데이터 요청
GET	/api/{userid}/job/{jobid}/log	작업 실행 로그 요청

그림 5. HPC 작업 관리 프레임워크 OpenAPIs

#### V. 결론 및 향후 연구

본 논문에서는 계산과학 시뮬레이션을 위한 HPC 작업 관리 프레임워크의 설계 및 구현에 대해 상세히 기술하였다. 추후, 안정성 향상을 위해 어플리케이션 서버 다중화에 대해 추가적인 연구, 개발을 수행할 예정이며, 실제 다분야 계산과학 응용(고에너지물리, 천문우주 등)에 시범 적용하여 기능성을 평가할 예정이다. 이와 더불어, 제안한 작업 관리 프레임워크에 대한 성능 시험 및 기능 확장을 수행할 계획이다.

```

POST -/api/{userid}/simulation/create
{ "title": "simulation title", "description": "simulation description" }

POST -/api/{userid}/simulation/{simid}/job/submit
{
  "title": "job test title",
  "description": "job test description",
  "executable": "$solver",
  "args": "Simputdeck",
  "category": {
    "type": "parallel",
    "procs": "16"
  },
  "links": {
    "solver": "/public/KISTI/solver2_pnjuncLAB",
    "inputdeck": "/user1/pnjuncLABinput.cmd"
  },
  "cluster": {
    "uuid": "0185958570ab4c7db8f0040403a1e82c"
  }
}
    
```

그림 6. 시뮬레이션 생성 및 작업 제출 예시

#### 참고문헌

- [1] Xiaoyu Yang, et. al, "Cloud computing in e-Science: research challenges and opportunities", The Journal of Supercomputing, Vol.70, Issue 1, pp. 408-464, 2014
- [2] John Towns, et. al, "XSEDE: Accelerating Scientific Discovery", Computer Science Engineering 16, 62, 2014
- [3] Dooley, Rion, et al. "Software-as-a-Service: The iPlant Foundation API", 5th IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS). IEEE, 2012.
- [4] Cholia, Shreyas, David Skinner, and Joshua Boverhof. "NEWT: A RESTful service for building High Performance Computing web applications." Gateway Computing Environments Workshop (GCE), 2010. IEEE, 2010.
- [5] RESTful Web APIs, [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- [6] Pyramid Web Framework, <http://docs.pylonsproject.org/projects/pyramid/en/latest>
- [7] RabbitMQ, <http://www.rabbitmq.com/>
- [8] SQLAlchemy: The Python SQL Toolkit and Object Relational Mapper, <http://www.sqlalchemy.org>
- [9] 김한기, 유정록, "SAGA python 기반의 LoadLeveler Adapter 개발", 2013 한국정보과학회 추계학술대회 학술발표논문집