응용 그리드를 위한 PSE 설계

심규호[0*] 허대영* 황선태* 정갑주** 박형우***
국민대학교, 건국대학교, KISTI
{simsaint[0*], phinoccio*, sthwang*}@cs.kookmin.ac.kr {jeongk**}@imc.konkuk.ac.kr, {hwpark***}@kisti.re.kr

Problem Solving Environment (PSE) Design of Application Grid

Gyuho Sim[0*] Daeyoung Heo* Suntae Hwang* Kapjoo Jeong** Hyoungwoo Park***
*Department of Computer Science, Kookmin University
**College of information and Communication, Konkuk University
***Korea institute of Science and Technology and Information

요 약

그리드 기술은 많은 응용 분야에서 보다 효율적인 실험을 위한 인프라를 제공한다. 이는 지리적으로 분산되어 있는 휴지상태의 리소스를 많은 컴퓨팅 리소스를 필요로 하는 응용 연구자에게 제공함으로써 High Performance와 High Throughput을 실험자에게 제공할 수 있다. 하지만 이러한 성능적 도움에도 불구하고 응용 실험자는 그리드 환경에서의 실험은 접근의 비용이성과 응용 플랫폼의 부족으로 여전히 어려움을 가지고 있다. 이러한 어려움 해결하기 위해 응용 플랫폼 개발자는 투명한 그리드 접근을 제공하는 동시에 실험자에게 친숙한 환경을 제공해야 한다. 또한 반복되는 단순 작업을 단순화 할 수 있는 환경과 실험자 자신의 작업을 정의할 수 있는 환경이 필요하다. 본 논문에서는 그리드 응용 플랫폼에서의 사용자에게 제공되어야 요구 사항인 투명한 그리드 접근 이외의 실험자를 위한 워크플로우의 정의와 데이터 지향적인 설계 방법을 제안하고 있다.

1. 서 론

그리드 컴퓨팅 기술의 발전은 자연과학 분야와 응용 분야에서의 커다란 혁명으로 다가왔다. 많은 컴퓨팅 파워를 요구하는 이러한 분야에서는 초고속의 네트워크 망으로 연결된 많은 컴퓨팅 파워를 제공하는 기관으로부터 리소스를 공유하여 사용할 수 있음은 자연과학 분야와 응용 분야에서의 급속한 발전이 이루어 질 수 있음을 의미한다. 그러나 초고속 인터넷 망, 클러스터, 슈퍼 컴퓨터등 그리드 기술을 지원하기 위한 인프라는 거의 완성 단계에 머무른 반면, 리소스를 공유하여 사용하기 위한 사용자 환경 및 응용 시스템들은 아직 부족한 단계이다. 본 논문에서는 그리드 응용 시스템 중 분자 시뮬레이션에 관련하여 실험자에게 제공되어야 할 PSE 기술을 제안하고 있다.

2. 관련연구

2.1 WfMC 레퍼런스 모델

표준 워크플로우 구조를 기반으로 일반적인 워크플로우 시스템의 구조를 정의해 보면 워크플로우 시스템은 프로세스 정의구조, 워크플로우 엔진, 워크리스트 핸들러 및 UI, 애플리케이션의 4가지 핵심 구성 요소와 이들의 기능을 지원하는 데이터 영역으로 정의된다. WfMC 워크플로우 서비스를 5개의 기능적 인터페이스로 구분하여 보면 다음과 같다.

가. 정의된 프로세스의 상호 교환을 위한 인터페이스
나. 워크플로우 클라이언트 애플리케이션 인터페이스
다. 워크플로우 Invoked 애플리케이션 인터페이스
라. 워크플로우 엔진간의 상호 연동을 위한 인터페이스
마. 수행결과 내역의 감시 및 통계 처리를 위한 인터페이스

WfMC에서는 이들 5가지 인터페이스에 대한 WfMC 레퍼런스 모델[1]을 정의했으며 각각의 인터페이스를 구성하는 API들도 정의하고 있다.

3. 응용 그리드를 위한 PSE 모델 설계

많은 컴퓨팅을 요구하는 응용 분야의 실험들을 살펴보면 실험자 자신만의 작업 흐름을 가지고 있으며, 일정한 단순한 반복적인 작업을 한다. 이는 실제 컴퓨팅을 하는 시간과 더불어 많은 시간을 소요하게 된다. 본 논문에서 제안하는 PSE 모델에서는 실험자 자신의 워크플로우를 정의함으로써 작업 프로세스를 자동화 하며, 프로세스에서의 일부 데이터 부분을 파라미터화 하여 단순 반복 작업을 최소화 하는 모델을 제시하고 있다.

3.1 워크플로우 모델 설계

워크플로우는 문서, 정보, 혹은 작업 프로세스에 대한 전체 혹은 일부를 미리 정의된 규칙을 통한 자동화로 정의한다.[2] 본 논문에서 제시하는 워크플로우 모델은 작업 프로세스간의 자동화와 데이터간의 의존성 관계를 분리하여 정의한 워크플로우 모델을 제시한다. 각각의 작업 프로세스

는 입력 데이터로부터 처리하여 결과 데이터를 생산한다. 이러한 흐름은 각 작업프로세스의 흐름을 정의하는데 있어서 데이터간의 의존성이 나타나게 된다. 이는 각 프로세스 실행하기 전 실행하는 작업공간에서의 파일 이동이 이루어져야 함을 의미한다. 여기서 정의하는 워크플로우 모델은 각 작업 프로세스간의 실행순서와 작업 프로세스에서 필요로 하는 입력 파일과 작업프로세스의 실행 후 생성되는 결과 파일들 간의 의존성을 서로 다른 공간에서 정의함으로 워크플로우 정의하는 사람으로 하여금 보다 직관적인 워크플로우를 정의할 수 있도록 한다.

3.2 데이터 지향 모델 설계

자연과학 및 응용 분야의 시뮬레이션들은 데이터에 관련된 작업들이 대부분이다. 이러한 데이터는 툴에 의해 생성되거나 시뮬레이션 작업을 통해 생성된다. 이렇게 생성된 데이터는 분석 툴에 의한 입력 데이터로 사용되거나 연관된 다른 시뮬레이션 작업의 입력 데이터로 생성된다. 또한 각 시뮬레이션들은 파라미터 값의 변경을 통한 반복적인 작업이 이루어진다. 본 논문에서는 이러한 반복적인 작업을 실험자에게 손쉽게 이루어질 수 있는 데이터 지향적인 파라미터화 기법을 제안한다.

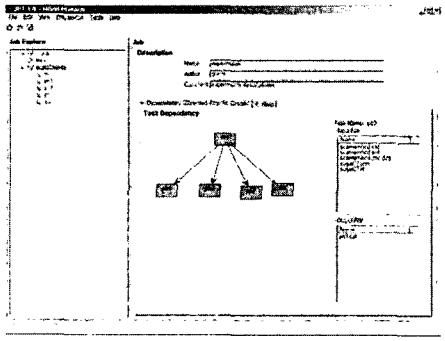
분자 시뮬레이션 작업을 살펴보면 입력파일에 대한 다중의 셀렉션이 있고, 각 시뮬레이션 마다 시뮬레이션을 위한 스크립트 파일에서의 파라미터 값을 변경한 반복적인 작업이 이루어진다. 이러한 반복적인 작업에서 입력 파일과 스크립트 파일에서의 변경 가능한 값들을 파라미터화하여 반복적인 작업에서의 각 프로세스 복제와 파라미터 값들의 변경만으로 반복적인 작업을 손쉽게 할 수 있게 된다.

4. 응용 그리드를 위한 PSE 모델 구현

4.1 워크플로우 구현

여기서 제시한 워크플로우에서는 각 프로세스간의 실행순서의 정의와 각 프로세스에서의 입력파일에 대한 의존성 관계를 정의함으로써 워크플로우를 정의한다. 프로세스간의 실행순서는 사용자의 워크플로우와 일치하며 사용자에 의해 정의된다. 여기서의 프로세스 정의는 Job, DAG, Task의 구조를 가지며 Task는 사용자에 의해 정의된 프로세스로 DAG는 실행순서를 가지는 Task의 집합으로 정의하며 Job은 사용자의 실험 전체의 워크플로우를 정의하는 Task의 집합으로 정의한다. DAG안에 포함되는 Task는 실행순서로써 Pre와 Post 속성을 가지고 있으며 Pre에서는 자신이 실행하기 위해 선행되어서 수행되어야 할 프로세스를 표현하고 Post 속성에서는 프로세스 수행 후 수행되어야 할 프로세스를 정의한다.

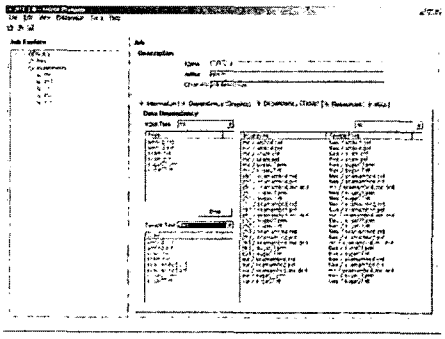
[그림 1]은 각 프로세스간의 실행 순서를 정의하는 화면을 보여준다. 사용자는 각 프로세스를 정의하고 각 프로세스간의 실행순서를 화살표를 이용하여 정의하게 된다.



[그림 1] 실행 순서 정의

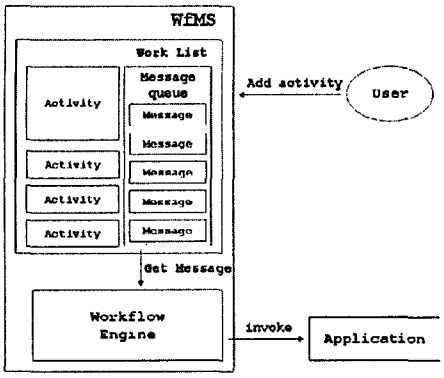
워크플로우 흐름에서 각 프로세스의 입력파일은 기존 파일을 모아 놓은 Repository 혹은 이전 실행된 프로세스의 결과 파일과의 연관성을 가진다. 이는 각 입력파일마다 Reference 속성을 주어 프로세스가 수행되기 전 Reference 속성에 따라 각 파일을 자동으로 프로세스 작업공간으로 복사할 수 있다.

[그림 2]는 각 프로세스에서 프로세스를 실행하기 위해 필요한 입력 파일에 대한 파일 의존성을 정의한 GUI 화면이다. 각 입력파일들은 파일들을 모아 놓은 Repository나 다른 프로세스에서의 결과파일과의 의존성을 정의한다. 이러한 실행순서 파일 의존성들은 XML 형태로 정의되며 이러한 정의는 워크플로우 시스템에 의해 처리된다.



[그림 2] 파일 의존성 정의

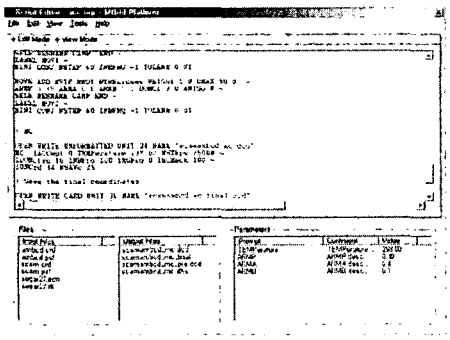
본 논문에서 구현한 워크플로우 시스템은 액티비티, 워크플로우 엔진, 워크리스트로 구성된다. 각 실행 프로세스들은 워크플로우 시스템내에서 액티비티와 연관되며 각 실행 프로세스의 액션은 메시지로 정의하여 워크리스트의 메시지 큐로 삽입한다. 메시지로 정의된 액션들에 대해 워크플로우 엔진은 메시지큐로부터 각 메시지들을 꺼내와 처리하게 된다. 실행 프로세스의 인터페이스는 allocate, active, stop, delete, update로 구성되며 각 인터페이스의 실행은 각 시스템에 따라 한개의 메시지 혹은 여러개의 메시지로 구성된다. [그림 3]은 본 논문에서 제시한 워크플로우 관리 시스템의 구조를 보여주고 있다.



[그림 3] 워크플로우 관리 시스템 구조

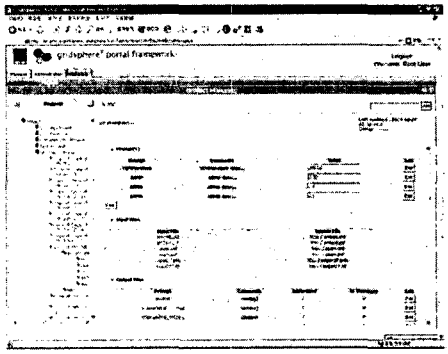
4.2 데이터 지향 모델 구현

사용자에 의해 정의된 각 프로세스들은 사용자에 요구에 의해 각 요소들을 파라미터화 할 수 있다. 각 프로세스의 입력파일과 의존성을 가지는 Repository에 있는 파일들이나 혹은 [그림 4]과 같이 프로세스 정의 도중 작성된 스크립트 중 일부를 파라미터로 표현한다.



[그림 4] 스크립트 파일에서의 파라미터화

파라미터화 된 각 프로세스들은 각 프로세스의 복제를 통해 다수의 프로세스를 정의할 수 있으며 복제된 각 프로세스는 [그림 5]와 같이 사용자의 요구에 따라 파라미터 값을 변경을 통하여 다중의 시뮬레이션 프로세스로 정의할 수 있다. 이러한 모델을 통해 Job 상위에 Project의 개념을 정의할 수 있다. Project는 유사한 Job들간의 집합으로 정의하고 Project 내에 포함된 Job간의 구별은 파라미터 값으로 정의한다. 하나의 프로젝트 내의 Job들은 사용자에 의한 파라미터를 변경하며 유사한 Job들을 정의하게 된다. 이러한 형태의 Project는 하나의 Project Sheet로 표현하여 파라미터값에 대한 각 Job들을 분석할 수 있는 환경을 제공한다. Project Sheet의 애트리뷰트 값은 파라미터화 된 값들과 결과물로 사용자의 정의에 의해 구성된다.



[그림 5] 파라미터 변경

5. 결론 및 향후 과제

본 논문에서는 응용 연구자의 워크플로우를 분석하고 단순한 반복적인 작업을 최소화하기 위하여 응용 그리드를 위한 워크플로우 모델과 데이터 지향적인 파라미터화 모델을 제시하고 있다. 응용 그리드를 위한 워크플로우에서는 프로세스 실행 순서와 데이터 의존성으로 분리하여 보다 직관적이고 데이터간의 연관성을 표현하여 보다 유연한 워크플로우 모델을 제시하고 있으며 이를 실행하기 위한 워크플로우 시스템을 정의하였다. 또한 응용 실험자의 워크플로우 분석을 통한 단순한 반복 작업을 최소화하기 위한 파라미터화를 통한 모델을 제공하고 있다. 파라미터화를 통해 사용자는 하나만의 프로세스 흐름을 정의함으로써 다중의 실험에 사용할 수 있고 유사한 프로세스간의 분석할 수 있는 Project Sheet를 제시하고 있다.

향후 과제로써 결과 데이터를 사용자가 요청할 경우 데이터 의존성을 기반으로 하여 실행순서를 역 추적하여 실행 할 수 있는 모델을 제시할 것이며 이는 사용자에게 실험의 흐름보다는 결과 데이터를 중심으로 하여 연구할 수 있는 환경을 제공할 것이다.

6. 참고문헌

[1] David Hollingsworth "Workflow Management Coalition The Workflow Reference Model" 19-Jan-95 http://www.wfmc.org

[2] Document Number WFMC-TC-1025, "Workflow Management Coalition Workflow Standard"