슈퍼컴 그리드 어카운팅 정보 수집 시스템 설계 및 구현

장경익° 김법균 황호전 두길수* 곽의종 안동언 정성종 장행진**
전북대학교, 서남대학교*, KISTI**
e-mail : mainclass@duan.chonbuk.ac.kr

Design and Implementation of GRID Accounting Information Gathering System for Supercomputer

장경익° 김법균 황호전 두길수* 곽의종 안동언 정성종 장행진**
전북대학교, 서남대학교*, KISTI**
{mainclass°, kyun, hjhwang}@duan.chonbuk.ac.kr, {dgs, kej, duan, sjchung}@moak.chonbuk.ac.kr
hjjang@hpcnet.ne.kr

요 약

초고속 네트웍을 기반으로 지리적으로 분산된 리소스들을 공유하며 강력한 컴퓨팅 파워를 낼 수 있는 그리드 환경에 관한 연구가 활발히 진행되고 있다. 본 논문에서는 그리드 환경을 구축하기 위한 기술중에 어카운팅 정보의 수집에 관해서 기술한다. 특히 클러스터 형태의 시스템에서 OGSA RSU-WG의 GSAX와 Usage Records Working Group(UR-WG)를 기반으로 하여 시스템을 설계하였다. 설계를 바탕으로 각 노드에서 발생하는 어카운팅 정보와 job 수행 정보를 수집하고, 통합하는 모니터링 툴의 구현에 대해 상세히 설명한다.

1. 서 론

그리드는 차세대 인터넷이 추구하는 고품질, 실시간 가시화, 대용량 정보처리 및 협업연구등이 가능하기 때문에 선진국의 경우 현재 핵심 어플리케이션을 중심으로 고속의 슈퍼컴퓨팅환경과 접목시켜 국가 차세대 인터넷 인프라를 구축하고 있는 추세이다. 국내에서도 현재 그리드 응용 연구가 활발히 진행되고 있다.

그리드 서비스 중 상용 서비스를 실시하기 위해서 반드시 필요한 부분이 바로 어카운팅 시스템이다. 어카운팅 시스템은 각 사이트 자원의 상태정보 및 각 자원에 대한 접근 및 사용권한 해당 사이트에 대한 인증, 사용된 자원에 대한 비용 산출 서비스가 필수적이다.

위에서 언급한 어카운팅 시스템중에 본 논문에서는 GSAX의 구조와 UR-WG에서 제안한 field들을 참고하여 슈퍼컴에서 발생되는 프로세스 어카운팅 정보 및 Job 매니저인 LoadLeveler를 통한 Job 실행시 발생하는 어카운팅 정보까지 수집하고, 모니터링 하는 시스템을 설계 및 구현한다.

2. 관련 연구 / 제안

2.1 GSAX (Grid Service Accounting Extensions)

GGF의 RUS-WG에서 제안한 구조로 OGSA를 기반으로 설계되었다. 어카운팅 기능에 서비스 개념을 도입하여 어카운팅 구조에서 각 기능별 요소들을 서비스화 하고 각 VO에게 과금 서비스 등에 관한 자율성을 부여하였으며, 작업이 실행되는 동안 동적으로 타 서비스를 이용하는 경우를 고려하였고, 그리드 사용자의 잔고에 따라 진행 중인 작업의 지연 및 중단 여부를 결정할 수 있도록 고안되었다.

2.2 DGAS (DataGrid Accounting System)

EGrid의 프로젝트 중의 하나로 현재 진행중에 있으며, 특징적인 것은 시스템 내의 각 구성요소간의 메시지 전달을 XML을 사용하여 구현하고 있다.

2.3 Usage Record Fields

Usage Record Working Group(UR-WG)에서 제안한 어카운팅 정보의 field들에 관한 연구로 성질이 다르게 운영되는 각각의 사이트들에서 필요한 어카운팅 정보의 field 들을 요약 정리하고 mini-mum field를 뽑아냄으로서 추후 서비스될 빌링시스템에 필수적으로 포함해야할 field 들을 정의했다.

3. 설계 기반 및 구현 환경

본 논문에서 설계한 슈퍼컴에서의 그리드 어카운팅 정보 수집 시스템은 Global Grid Forum(GGF)의 OGSA Resource Usage Service Working Group (RUS-WG)에서 제안한 GSAX와 UR-WG에서의 field 들을 기반으로 했기 때문에 다른 그리드 환경과의 연동이 쉽도록 설계하였다.

시스템을 구현하는데 있어서 사용된 실제적인 코딩 도구는 Java 이다. Java를 선택한 이유는 그리드 환경에서의 'Heterogeneous' 특성과 Java의 플랫폼 독립성이 부합되기 때문이다. 다시 말해서 이기종의 모든 시스템에 적용하기 위해서는 플랫폼 또는 OS 로부터 독립적이어야 한다는 것이다. 본 논문에서 설계 및 구현한 시스템의 환경은 표1과 같다.

목 록	내 용
Machine	*IBM RS/6000 SP 9076-550 / 32 Node(2cpu)*
OS	AIX 4.3.2
Environment	Java 1.1.6 / 1.2
DataBase	Mysql 4.0.12
Middleware	Globus Toolkit 1.1.4
Job Manager	LoadLeveler 2.1

표 1. 시스템 구현 환경

4. 어카운팅 시스템 설계 및 구현

4.1 AIX 프로세스 어카운팅 파일(pacct) 세부 구조

시스템 상에서 지원하는 표준 C/C++ Language의 pacct 파일 관련 API를 이용하면 AIX의 각 노드에서 생성되는 프로세스 어카운팅 정보를 손쉽게 얻을 수 있지만, 본 논문에서 설계 및 구현한 시스템은 Java의 여러 이점들로 인하여 도구를 Java로 선택했기 때문에 pacct 파일에 대한 분석이 필요하다.

표2는 분석한 pacct 파일의 실제 구조이다.

순 서	Field	Size	Type
1	ac_flag	1	char
2	ac_stat	1	
3~4	padding	2	*
5~8	ac_uid	4	unsigned int
9~12	ac_gid	4	
13~16	ac_tty	4	signed long
17~20	ac_btime	4	
21~22	ac_utime	2	unsigned short
23~24	ac_stime	2	
25~26	ac_etime	2	
27~28	ac_mem	2	
29~30	ac_io	2	
31~32	ac_rw	2	
33~40	ac_comm	8	char[8]

표 2. pacct 파일 세부 구조.

위의 표에서 보이는 것처럼 파일 중간에 명시적이지 않은 padding byte 들이 존재하며 이것은 8, 16, 32 byte 등의 단위로 잘라서 분석해 보면 어느곳에 padding byte 들이 존재하는지 알 수 있다. 또한 프로그램 상에서 byte 단위로 읽을 때 2 byte 이상의 값들은 순서가 바뀌어 있기 때문에 그대로 읽으면 않되며, 반드시 앞뒤 byte를 바꿔서 읽어야만 제대로 된 값들을 얻을 수 있다.

AIX의 프로세스 어카운팅 정보는 UR-WG에서 제안한 field들과 일치되는 부분이 많으며 부족한 부분은 구현시에 추가적인 연산을 통하여 확보했다.

4.2 LoadLeveler 의 어카운팅 정보

LoadLeveler의 경우 어카운팅 정보를 얻어내기 위해서는 어카운팅 관련 설정을 해주어야 한다. 환경설정을 위한 파일은 LoadLeveler가 설치된 경로에 바로 존재하는 'LoadL_config' 이다. 표 3은 어카운팅과 관련해서 설정해야할 요소들과 실제 설정 예를 보여준다.

keyword	설 명	비 고
GLOBAL_HISTORY	*global history* 파일이 저장될 전체 경로.	$(spool)
ACCT	어카운팅 시스템을 활성화 시킴.	A_OFF/ A_ON
실제 설정 예		
# # Specify accounting controls # ACCT = A_ON A_DETAIL ACCT_VALIDATION = $(BIN)/acctval GLOBAL_HISTORY = $(SPOOL)		

표 3. LoadLeveler 환경 설정

LoadLeveler는 어카운팅 정보를 각 노드별로 가지고 있을 뿐만 아니라 노드별 정보를 통합한 형태의 정보도 가지고 있다. 본 논문에서 사용된 통합된 형태의 어카운팅 파일은 '설치경로/spool/history' 이다. LoadLeveler의 어카운팅 정보는 하나의 job에 대한 정보와 그 job이 수행된 노드들에서의 정보도 전부 포함하고 있기 때문에 프로세스 어카운팅 정보와는 달리 한 job에 대한 어카운팅 정보를 활용하기에 충분하다. 또한 UR-WG의 field들과도 대부분 일치했고, 마찬가지로 구현시 추가적인 연산을 통해 부족한 정보를 더했다.

4.3 그리드 어카운팅 시스템 구조

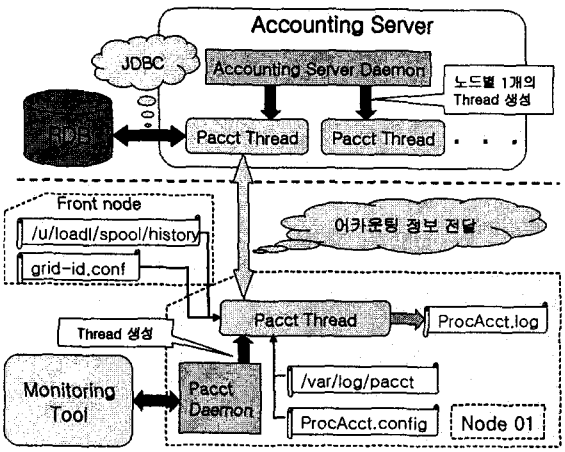

그림 1. 어카운팅 시스템 구성

그림1은 본 논문에서 구현한 어카운팅 정보 수집 시스템의 상세한 전체 구조이다. 슈퍼컴의 각각 노드별로 pacct 파일과 LoadLeveler의 history 파일의 내용을 읽는 Pacct Daemon이 존재하며, Daemon은 파일의 내용을 읽어 들여 Accounting Server Daemon에게 socket 통신을 이용하여 전송한다. 또한 각 노드별 Daemon 들은 Monitoring Tool 에 의해서 동작이 되며, 동작된 Daemon 들은 pacct 파일과 history 파일을 통해 어카운팅 정보를 서버측에 보내게 된다.

4.2 각 노드의 Pacct Daemon

슈퍼컴의 노드 1개당 하나씩 존재하는 Pacct Daemon은 사전에 ProcAcct.config 파일에서 Daemon 이 수행되기 위한 환경변수들을 읽어 들인다. 그림2는 Pacct Daemon 의 실행환경을 결정하는 ProcAcct.config 파일이다.

```
# File Name : ProcAcct.config
# This file is used by Process Accounting System

# Process Accounting File Location.
ProcAcctFilePath = /var/log/pacct

# Grid User List File Location.
GridIdFilePath = ./grid-id.conf

# Continuous connection time.
ConnectionTime = 300

# The inteval of time searching new Accounting data.
SearchTimeInterval = 1
```

그림 2. ProcAcct.config

위 그림에서 ConnectionTime 은 각 노드에서 추가되는 어카운팅 정보가 없을 경우 서버측과의 연결을 끊기 전 대기 시간이다. 영구적인 연결 유지는 서버측 또는 노드측에도 부하의 원인이 되기 때문에 필요한 요소이다. SearchTimeInterval 은 pacct파일과 history파일에 새로운 어카운팅 정보가 기록되었는지를 감시하는 주기적인 시간 간격을 말한다.

Pacct Daemon은 실제로 작업을 수행하기 위해 thread를 생성시키며 pacct와 history 파일들을 읽어들이는 루틴은 생성된 thread가 담당하게 된다. Daemon은 thread에 대한 제어권을 가지고 있기 있으며 실행 및 중단을 할 수 있다. 그렇기 때문에 thread는 실행 중에 수행한 작업에 대한 로그정보를 남겨, 작업의 진행 정도를 기록하고, 차후에 thread를 재가동 할때 log 파일을 참고하여 작업이 중복되는 현상을 방지한다. log 파일은 그림3과 같다.

```
# Pacct file Read Deamon log file.

PacctFileDate = 20030820013214
PacctFileSize = 1802560
ReadRecordCount = 28165
DaemonStopTime = 20030820013308
-----------------------------------------
# History file Read Deamon log file.
HistoryFileDate = 20030901181622
HistoryFileSize = 47829378
ReadRecordCount = 248251
DaemonStopTime = 20030901181749
```

그림 3. ProcAcct.log / HistoryAcct.log

그림1에서 Pacct Daemon은 Front node에 있는 grid-id.conf 파일을 참조하게 되어있다. 이것은 그리드 사용자 계정(Local ID)에 관한 정보이며, 본 논문에서 설계 및 구현한 시스템은 그리드 환경이 그리드 사용자를 위한 계정을 따로 만들어 관리하게 되어 있다는 전제하에 구성되었다. pacct/history 파일들로부터 어카운팅 정보를 추출할때, grid-id.conf

파일에 있는 ID들과 일치하는 정보만을 추출하도록 구성했다. 그림4는 grid-id.conf 파일의 구조를 보여준다.

```
# 'GRID_ID' is not user id.
# It is a record that has general property

GRID_ID
{
        max-binding = 5
}
# each local id has some fields
griduser01
{
        bindable = 1     # bindable
}
griduser02
{
        bindable = 1     # bindable
}
```

그림 4. grid-id.conf

Pacct Daemon에 의해 생성된 thread는 pacct /history 파일들을 주기적으로 검사하여 파일의 변화 유무를 체크하고, 변화되었을 경우 추가된 어카운팅 정보를 추출하여 Accounting Server Daemon에게 전송하게 된다.

4.2 Server Daemon

슈퍼컴의 Front Node에 존재하는 Server Daemon은 가동과 동시에 각 로컬 노드로 부터 socket 접속을 기다리며, 로컬 노드에서 접속을 시도할 때 노드당 1개씩 thread를 생성하여 각 노드에서 전송되는 어카운팅 정보를 받아 지정된 RDB에 저장하는 역할을 한다. 생성된 thread들은 각각 개별적으로 DB Server에 대해 접속을 하기 때문에, DB Server는 트랜잭션 처리가 가능한 DB로 구성을 하였다.

4.3 어카운팅 시스템 모니터링 Tool

슈퍼컴에서의 어카운팅 정보 수집은 노드가 많아질수록 관리가 까다롭다. 그런 이유로 본 논문에서 구현한 어카운팅 시스템은 관리가 수월하도록 하기 위해 GUI 기반의 모니터링 Tool을 제작했다. 그림5에서 보는바와 같이 모니터링 Tool에는 각 노드별로 어카운팅 정보의 수집 상태를 보여주며, 각 노드에 있는 Pacct Daemon에게 socket 통신을 통한 메시지를 전달함으로서 어카운팅 수집 thread를 생성 또는 소멸할 수 있도록 하였다. 그리고 실시간으로 수집되는 어카운팅 정보를 보여줌으로서 각 node들의 좀더 상세한 상태정보를 볼 수 있도록 하였다.

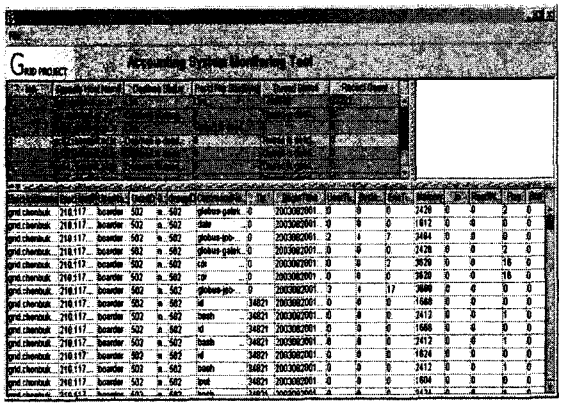

그림 5. 어카운팅 시스템 모니터링 Tool

5. 결론 및 향후 연구

본 논문에서 언급된 슈퍼컴은 그 형태가 리눅스 클러스터와 유사하다. 클러스터 형태의 시스템은 다른 상용 유닉스 시스템(MPP 등)들도 비슷한 형태를 취하고 있기 때문에 본 논문에서 설계 및 구현한 시스템은 적용범위가 매우 넓고 서론에서도 언급했듯이 Java를 사용함으로서 확장성을 고려하였다.

향후 연구에서는 기 구현된 어카운팅 정보 수집 시스템에 더하여 PBS를 통한 job의 수행시 발생하는 어카운팅 정보에 대해 분석하고 본 논문에서 구현한 시스템에 분석된 내용을 함께 포함하여 설계 및 구현할 예정이다.

참고 문헌

[1] S. Mullen et al, "Grid Authentication, Authorization and Accounting Requirements Research Document", (draf), GGF8, 2003

[2] A. Beardsmore et al, "GSAX(Grid Service Accounting Extensions)", (draft), GGF6, 2002

[3] I. Foster, C. Kesselman(eds), S. Tuecke Q.25, "The Anatomy of the Grid:Enabling Scable Virtual Organizations", Intl. J. Supercomputer.

[4] I. Foster, C. Kesselman(eds), S. Tuecke Q.25, "The Anatomy of the Grid: Enabling Scable Virtual Organizations", Intl. J. Supercomputer.

[5] S. Tuecke et al, "Open Grid Services Infrastructure (OGSI)", (draft), GGF, 2003

[6] http://www.ggf.org

[7] http://www.globus.org

[8] http://www.psc.edu/~lfm/UR-WG

[9] http://www.gridforumkorea.org