

# 고가용성 솔루션 구축을 위한 그리드 측면에서의 소프트웨어 아키텍처를 통한 로드밸런싱 구현

Software Architecture for Implementing the Grid Computing of the High Availability  
Solution through Load Balancing

이병엽\*, 박준호\*\*, 유재수\*\*  
배재대학교 전자상거래학과\*, 충북대학교 정보통신공학부\*\*

Byoung-Yup Lee(bylee@pcu.ac.kr)\*, Jun-Ho Park(arionfit@naver.com)\*\*,  
Jae-Soo Yoo(yjs@chungbuk.ac.kr)\*\*

## 요약

인터넷 환경의 급속한 발전과 더불어 국내외 미션 크리티컬한 비즈니스 환경이 온라인에 의해 서비스 되고 있다. 반면 단일 환경 서버환경의 구축을 통해 비효율적인 IT Resources들의 자원 낭비가 가중되고 있는 현실이다. 따라서 웹 환경을 통해 처리되어야 할 정보의 양의 급증과, 이의 처리를 위해 여러 개의 단일 서버를 고속의 네트워크로 연결한 고가용성 구현이 가능한 클러스터 컴퓨팅 시스템이 등장하게 되었다. 하지만 클러스터 컴퓨팅 기술의 다소 제한적인 IT Resource의 단점을 가지고 있다. 이러한 단점을 보완하기 위하여 그리드 컴퓨팅 환경은 기존의 분산 컴퓨팅 기술의 확장된 개념으로, 다양한 분야에서 저비용의 고성능 컴퓨팅 퍼포먼스를 얻기 위하여 그리드를 이용하고 있다. 그러나 공통적으로 이용할 수 있는 정보 시스템의 부재로, 현재의 그리드는 대규모의 단일한 환경으로서의 그리드가 아닌, 기존의 클러스터 컴퓨터와 같은 의미로 사용되고 있다. 따라서 자신이 구성한 분산 컴퓨팅 환경을 그리드 환경의 한 부분으로 포함시키기 위해서는 컴퓨팅 환경에 대한 정보를 그리드의 이용자가 공유할 수 있어야 하며, 정보 서비스를 공개하여 공유할 수 있도록 해야 한다. 본 논문에서는 Grid 기술을 통하여 데이터베이스 로드 밸런싱 목표치에 대한 검증을 제시하고 향후 고가용성 데이터베이스 구현을 위한 아키텍처를 제시한다.

■ 중심어 : | 클러스터 DBMS | High Availability | Grid |

## Abstract

In these days, internet environment are very quickly development as well on-line service have been using a online for the mission critical business around the world. As the amount of information to be processed by computers has recently been increased there has been cluster computing systems developed by connecting workstations server using high speed networks for high availability. but cluster computing technology are limited for a lot of IT resources. So, grid computing is an expanded technology of distributed computing technology to use low-cost and high-performance computing power in various fields. Although the purpose of Grid computing focuses on large-scale resource sharing, innovative applications, and in some case, high-performance orientation, it has been used as conventional distributed computing environment like clustered computer until now because grid middleware does not have common sharable information system. In order to use grid computing environment efficiently which consists of various grid middleware, it is necessary to have application-independent information system which can share information description and services, and expand them easily. This paper proposed new database architecture and load balancing for high availability through Grid technology.

■ keyword : | Cluster DBMS | High Availability | Grid |

\* 이 논문은 2010년 교육과학기술부(지역거점연구단육성사업/충북BIT연구중심대학육성사업단)와 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 결과임.(No. 2009-0089128)

접수번호 : #101206-008

심사완료일 : 2011년 01월 28일

접수일자 : 2010년 12월 06일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

## I. 서론

2000년대 초반부터 급격하게 보급된 인터넷의 보급과 더불어 저비용 고효율 시스템 성능 및 시스템 확장을 유기적으로 용이하게 하는 클러스터 컴퓨팅 시스템이 필요하게 되었다[1][2]. 더욱 강력한 컴퓨팅 파워와 시스템의 안정적 서비스를 제공하기 위해, 클러스터 기반 DBMS에 대한 연구 및 개발이 활발히 이루어지고 있다. 클러스터 기반 DBMS는 대규모 데이터를 여러 노드에 분산 저장하고 일관성 있게 접근 할 수 있는 메커니즘을 제공하며, 또한 클러스터 시스템의 특징인 고성능, 고가용성, 고확장성(high scalability)을 지닌다[10]. 그리드 컴퓨팅 환경[1]은 방대한 분산 컴퓨팅 리소스를 조직, 운용하여 사용자에게 저비용으로 고성능의 컴퓨팅 파워를 제공하는 환경이다[2]. 인터넷과 마찬가지로 그리드 컴퓨팅은 상용이 아닌 과학 프로젝트들을 지원하기 위한 목적으로 개발되어 진화하고 있으며, 현재 e-Science, bioinformatics, astrophysics, Earthquake engineering, weather forecast 등의 다양한 분야에서 그리드 컴퓨팅 환경의 고성능 컴퓨팅 파워를 이용하기 위한 연구들이 진행 중이다. 기존의 분산 컴퓨팅 환경과는 달리 그리드 컴퓨팅 환경은 보다 큰 규모의 컴퓨팅 환경을 구축하고, 다양한 분야에서 단일한 대규모의 환경을 공유하여 작업을 수행하는 환경이다. 그러나 각 분야의 어플리케이션은 필요로 하는 정보의 종류도 다르고, 이용하는 시스템의 종류도 다르기 때문에, 기존의 글로벌스[15]와 같은 그리드 미들웨어는 각 분야에서 공통적으로 필요로 하는 정보, 예를 들어, 리소스의 현재 상태나 명세에 관한 정보만을 제공하였고, 그리드 환경을 이용하는 응용프로그램 개발자는 추가적으로 필요한 정보를 어플리케이션 레벨에서 직접 구현하여 미들웨어의 정보 서비스와 함께 이용하였다. 따라서 응용프로그램 개발자는 미들웨어에서 제공하는 정보 외에 필요한 정보를 독자적으로 정의하여 사용하였고, 이로 인하여 어플리케이션들 사이에서는 구축된 정보를 공유하거나 재사용하기 힘든 구조로 되어 있었다. 따라서 새로운 응용프로그램을 개발하거나, 기존의 응용프로그램의 기능을 확장하고자 할 때, 필요한 정보

시스템을 구축하기 위하여 많은 시간과 비용이 발생하게 되었다. Global Grid Forum(GGF)[16]에서는 분산된 이질적이고 동적 특성을 갖는 가상조직(Virtual Organizations)에서 서비스들의 통합과 관리가 요구되는 시스템과 응용프로그램을 위한 기반으로 그리드 환경의 표준 규약이 되는 Open Grid 서비스 구조(OGSA)[17]를 제안하였고, 변화 및 발전되는 컴퓨팅 환경에 맞추어 개선되고 있다. 또한, 웹서비스(Web Services)[18]와 그리드 기술을 토대로, OGSA는 핵심 그리드 서비스를 정의하고 그리드 환경에 필수적인 여러 가지 기본적인 그리드 서비스들을 정의한 Open Grid 서비스 Infrastructure[19]를 제안하고 있다. 현재의 그리드로 가장 널리 이용되고 있는 Globus Toolkit을 포함하여, Legion의 상업용 버전인 Avaki, Condor[20]등의 다양한 미들웨어로 구성되고 있으며, myGrid[21], GridLab[22], Unicore[23]등의 새로운 그리드 Infrastructure가 연구 진행 중에 있다.

본 논문은 다음과 같이 기술된다. 먼저 2절에서는 본 연구와 관련된 기술 및 연구에 대하여 설명하고, 3절에서는 상용 소프트웨어를 이용한 Grid 기술의 아키텍처의 구조에 대한 설명과 그리드 기반의 솔루션 구축을 통한 각각의 인스턴스에 대한 로드밸런싱 부분의 효율성을 검증하고 향후 Grid 기술 구현의 표준을 제시하였다. 4절에서는 다양한 그리드 기술을 기반으로 한 향후 과제에의 정의를 통해 결론을 내리고자 한다.

## II. 본론

최신 기반의 아키텍처(Architecture)에서 데이터(Data)를 기반으로 하는 접속도(Access), 볼륨(Volume), 복잡도(Complexity)가 극단적으로 증가하고 있는 경향을 [그림 1]에서 볼 수 있다. 이러한 이유는 인터넷 기반의 비즈니스 모델(Business Model)이 네트워크(Network)의 발전으로 거래량이 폭발적으로 증가하였고, 리얼타임(Real-Time)정보에 대한 분석 요구 증대로 인한 데이터 프로세싱(Data Processing)이 급격히 증가 하였다. 또한 SOA(Service Oriented

Architecture), MCA(Multi-Channel Architecture)등의 아키텍처가 요구하는 백엔드 프로세싱(Backend Processing)이 증가하였다.

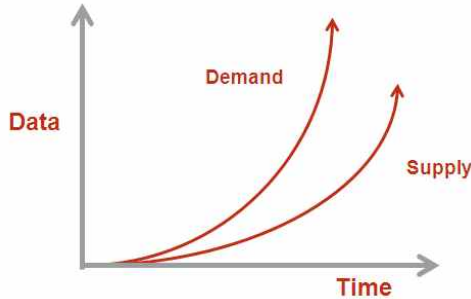


그림 1. 데이터의 요구 증가량

따라서 앞서 설명된 데이터 요구량의 증가에 따른 상황들에 대하여 Grid 기술의 최근 연구 동향을 살펴보고자 한다.

### 2.1 그리드의 정의

웹이 인터넷상의 정보를 공유하기 위한 서비스였던 것에 반하여 그리드는 컴퓨팅 파워와 데이터 스토리지를 인터넷상에서 공유하기 위한 서비스를 뜻한다[16]. 더 나아가 그리드는 단순한 컴퓨터간의 네트워크 연결을 넘어서 궁극적으로 전 세계의 컴퓨터 네트워크를 사용자의 지역적 위치에 상관없이 하나의 광대한 리소스로 사용할 수 있게 하는 서비스를 제공한다. 또한 그러한 서비스를 통한 전 세계 사용자들의 협업을 가능하게 한다는 비전을 가지고 있다.

### 2.2 그리드 관련 연구

최근 IT비용 절감을 위한 혁신적인 기술로서 최근 가장 각광받는 분야는 가상화(virtualization)와 클라우드 컴퓨팅(cloud computing)이다. 시장 조사기관인 가트너는 이 두 가지를 가장 중요하게 여겨질 전략적 기술로 평가한바 있다[12]. 포레스트 리서치(Forrester Research) 또한 현재의 기업들은 클라우드 서비스가 제공하는 비용 절감 효과에 주목해야 하며, 그 중요성은

점점 커질 것이라고 예측하고 있다. 이러한 최선의 기술을 접목하기 위해서 다양한 소프트웨어 벤더 들은 그리드 컴퓨팅이라는 기술들을 연구하고 상용화 소프트웨어를 출시하고 있다. 그리드 컴퓨팅은 최근 활발히 연구가 진행되고 있는 분산 병렬 컴퓨팅의 한 분야로서, 네트워크로 연결된 다수의 이 기종의 컴퓨팅 리소스들을 묶어, 가상의 대용량 고성능 컴퓨팅 환경을 구성 하여, 고도의 연산 (computation intensive jobs) 혹은 대용량 연산 (data intensive jobs)을 수행하는 것이다. 기존 분산 병렬 컴퓨팅 기술인 클러스터 컴퓨터와는 달리 네트워크상에서 서로 다른 기종의 머신들을 연결한다는 점으로 인하여, 클러스터 컴퓨터에서는 고려되지 않았던 여러 가지 표준규약들이 필요해 졌고, 현재 글로벌스 프로젝트를 중심으로 표준들이 정립되고 있다.

그리드 컴퓨팅 환경은 지역적으로 넓게 분포된 다수의 이 기종의 컴퓨팅 환경으로 구성되어 있고, 각각의 컴퓨팅 리소스는 각기 다른 플랫폼과 응용프로그램을 가지고 있기 때문에, 사용자가 자신이 수행하고자 하는 작업을 수행할 수 있는 리소스를 찾기 위해서는 그리드 환경에 존재하는 다양한 정보를 이용할 필요가 있다. 그러나 사용자는 동적으로 변화하는 그리드 환경의 변화를 모두 관리할 수 없기 때문에, 이러한 정보들을 그리드 환경을 구성하는 미들웨어로부터 얻어야 한다. 현재 릴리즈 되고 있는 GT4에서는 MDS4[24]를 통하여 사용자가 필요로 하는 정보 서비스를 제공한다. 그러나 Grid환경은 기존의 분산 컴퓨팅 환경과는 달리 다양한 분야의 다양한 목적을 위하여 사용되는 작업 수행 환경이기 때문에 MDS4 에서는 모든 시스템에서 공통적으로 필요로 하는 정보만을 제공하며, 특정 도메인에 한정적으로 필요한 정보의 제공은 할 수 없다. 이 때문에, 글로벌스를 이용하는 많은 응용프로그램에서는 추가적으로 필요한 정보를 얻기 위하여, 자체적으로 새로운 정보시스템을 개발하고, MDS와 함께 이용하는 형태로 개발되었다. 이는 그리드 환경의 이용을 어렵게 만들고, 동일한 환경에서 같은 역할을 하는 서비스가 중복되는 등의 비용의 증가를 야기한다.

## 2.3 그리드의 현황과 비전

초기의 그리드는 방대한 양의 계산이나 데이터를 필요로 하는 계산과학 분야에서 먼저 주목을 받기 시작했다. 기존의 계산 및 데이터 관리가 각 지역에 있는 슈퍼컴퓨터나 클러스터, 대용량 저장장치 등을 이용해서 이뤄지던 것을 그리드가 등장하면서 지금까지는 처리할 수 없었던 대용량의 계산 및 데이터를 처리할 수 있게 되었고, 처리 시간을 단축하거나 비용을 절감하는 등의 효과를 거두고 있다. 이후 대형 IT 기업들이 그리드 기술에 관심을 가지기 시작했고, SOA(Service Oriented Architecture)에 기초한 그리드 서비스의 개념이 자리잡게 되면서 그리드 컴퓨팅 기술은 차세대 비즈니스 모델을 위한 핵심인프라 기술로 인식되고 있다. 특히, 2002년 OGSA(Open Grid Service Architecture)의 출현은 이와 같은 흐름을 가속화하여 비즈니스 환경에 그리드 컴퓨팅 기술을 본격적으로 적용하게 하는 중요한 계기가 되고 있다. 이러한 흐름에 기초하여 그리드 기술은 미국의 국가 방위를 위해 시작됐던 인터넷이나 물리학자들 간의 연구 결과 및 자료 공유를 위해 시작된 웹(WWW)이 오늘날 누구나 사용할 수 있는 인프라가 되고 사람들의 삶과 기업의 형태를 변화시켰듯이, 고성능 컴퓨팅 환경을 요구하는 과학자들을 위해 출발하였지만 결과적으로 많은 기업과 사람들의 환경을 변화시키고 친숙하게 다가오게 될 것이라는 예측에 점차 무게가 실리고 있다. 그리드는 이제 실험 단계를 벗어나 실용 단계로 나아가고 있으며, 이에 따라 세계적 대규모 프로젝트들이 점차 실용 그리드(Production Grid)를 표방하고 그간의 결과들을 통합하기 시작하고 있다. 실제로 다양한 과학 분야에서의 애플리케이션들이 그리드인프라를 통해 더 나은 결과들을 보일 수 있도록 하는 연구가 수행되고 있다. 이러한 경향에 따라 그리드 프로젝트들 사이의 협력도 더욱 가속화되고 있으며 여러 대륙을 아우르는 글로벌 그리드 인프라 프로젝트들도 다양한 성과를 나타내고 있는 상황이다.

### 2.3.1 그리드의 분류

그리드 기술 활용의 확대와 함께 그리드 컴퓨팅은 계속 진화하고 있으며, 일반적으로는 현시점에서 최소 3

종류의 그리드가 현존하고 있다고 설명되고 있으며 그 분류는 다음과 같다[12].

**컴퓨팅 그리드** - 컴퓨팅 그리드는 지역적으로 분산되어 있는 컴퓨팅 파워를 공유하여 마치 한 대의 고성능컴퓨터처럼 사용할 수 있게 해 주는 그리드를 의미함.

예) SETI@Home, Korea@Home, DTF(Distributed Terascale Facility), 테라 그리드 구축 프로젝트 등.

**데이터 그리드** - 데이터 그리드는 대용량의 데이터를 네트워크를 통해 공유하기 위한 그리드.

예) CERN(유럽 입자 물리학 연구소)의 데이터 그리드 서비스, 미국 NDMA(National Digital Mammographic Archive) 그리드 프로젝트 등.

**액세스 그리드** - 액세스 그리드는 분산된 지역의 연구원들이 공동으로 프로젝트를 진행할 수 있는 협업 환경을 제공해 주는 그리드. 즉 고속의 네트워크 인프라 위에 컴퓨터장비, 데이터베이스, 시스템 핵 가속기, 천체 망원경과 같은 고가의 첨단 기자재를 연결하여 전문 인력 자원까지도 공유할 수 있는 협업 환경을 제공하는 그리드를 뜻함.

예) 바이오 그리드, 나노 그리드, 기상 그리드 등.

기존 그리드 애플리케이션으로는 대부분의 사람들이 계산용 그리드를 생각하고 있는 반면, 기업들은 데이터용 그리드를 그리드 기술의 주요한 활용 방안으로 이야기하고 있다. 이것은 컴퓨팅 파워를 공유하는 대신, 표준화된 방식을 통해 데이터를 내외부적으로 교환해 데이터 마이닝과 의사결정지원에 사용할 수 있도록 하는 서비스를 의미한다. 한편 액세스 그리드 및 협업용 그리드는 여러 곳에 흩어져 있는 원격지의 사용자가 초대형의 데이터 세트들을 공유해 함께 일할 수 있게 하여 주는 기능을 목표로 한다.

### 2.3.2 비즈니스 그리드의 분류

NEESGrid4는 지진 연구자들이 인터넷을 통해 가상

팀을 구성한 후 데이터뿐만 아니라 조사장비도 함께 공유하여 사용하는 협업 그리드를 구성하고 여기에 유틸리티 그리드와 기업 최적화 그리드 등 하부장르의 그리드가 계속 개발되고 있으며 그리드를 통해 이익을 얻을 수 있는 비즈니스 기능과 애플리케이션들을 파악하기 위한 산업계의 노력이 활발하게 지속되고 있는 상황이다. 그리드 기술 상용화로 기대되는 서비스 및 비즈니스 모델에 따른 그리드의 분류는 다음과 같다[12].

**엔터프라이즈 그리드 :** 개개의 기업 내에 존재하는 대형서버와 개인용 워크스테이션 등을 하나로 묶고 여기에 다른 지역에 위치한 동일기업 내 컴퓨팅 리소스까지 통합하는 인트라넷의 확장개념.

**파트너 그리드 :** 기업과 기업 간의 파트너십을 통한 기업 간 그리드 네트워크 형성.

**서비스 그리드 :** 기업마다 고정된 플랫폼을 대신하여 네트워크를 통해 공급되는 서비스를 통해 기업마다 중복된 투자 없이 특화된 기능을 공급받음.

현재까지 산업계의 그리드 채택 현황은 엔터프라이즈 그리드에 머물러 있는 상황이지만 다양한 조직 간의 정보공유에 따른 보안과 인증 관련 문제들이 해결됨에 따라 근시일 내에 파트너 그리드 형태의 서비스가 공급될 것으로 예상되고 있다. 그리드 인프라의 활용에 대한 산업체들의 장기적 비전은 다음과 같은 두 가지 사항으로 정리 될 수 있다. 첫째로는 최적화된 전사적 시스템 관리 기술을 제공(네트워크 전반에 걸친 성능 최적화, 보안향상, 주변 환경의 요구조건에 반응하는 자율 컴퓨팅 기반의 시스템 관리) : 궁극적으로 그리드는 전 지구적 인터넷 기반의 가상 컴퓨팅 환경을 제공하는 것을 목적으로 하고 있고, 두 번째로는 가정이나 회사가 전력이나 수도와 같은 공공 인프라 서비스를 사용하는 것처럼 그리드 인프라를 통해 컴퓨팅 자원들을 소싱 등의 유틸리티 서비스 형태로 제공하는 것이다.

산업계의 그리드 상용화 관련 움직임은 대형 IT 벤더들이 모여 엔터프라이즈 그리드 솔루션의 개발과 기업

내 그리드 컴퓨팅의 적용을 위해 출범한 EGA(Enterprise Grid Alliance) 컨소시엄을 중심으로 활성화되고 있는 상황이다. 현재 참여하고 있는 기업은 후지쯔, HP, 인텔, NED, 네트워크 어플라이언스, 오라클, 썬 마이크로시스템즈 등을 포함하고 있다. 이들은 그리드의 동적인 자원 공급 기술을 통해 기업 내의 컴퓨팅 수요의 변화에 따라 최소화된 비용으로 기업 내 자원을 활용할 수 있는 IT 환경을 추구하고 있다. 이러한 업계의 움직임을 따라 그리드 기술을 기업 환경에 적용하기 위한 노력이 급물살을 타고 있으며, 현존 하는 다양한 어플리케이션, 데이터베이스에 적용되어 상용화 되고 있는 실정이다.

#### 2.4 기존 시스템 성능 향상의 문제점

시스템의 설계에서 가장 중요한 관심 사항은 성능이다. 성능 향상을 위해서는 하드웨어, 소프트웨어, 어플리케이션의 측면에서 접근하며 튜닝을 통해서 해결할 수 있는 성능의 영향의 한계를 3가지 측면에서 정리하면, 첫째 소프트웨어의 측면에서 살펴보면 서로 연관된 소프트웨어들의 최적의 설정을 찾아 튜닝을 해도 성능을 향상 시키는 것에는 한계가 있고, 하드웨어의 영향이 절대적이다. 또한 하드웨어의 수평적 확장(Scale-out)의 경우 수동적인 소프트웨어의 변경이 발생한다. 반면 하드웨어의 측면에서 살펴보면 수직적 확장(Scale-Up)의 경우 확장은 간편하나 비용이 증대하며 확장의 한계가 존재한다. 수평적 확장(Sales-Out)의 경우 이론적으로 확장의 한계는 없으나 비용이 증대하며 하드웨어의 관리 대수가 증가 할수록 유지보수와 관련된 비용, 인력 및 관리 포인트가 증가 한다는 단점이 존재한다[11]. 어플리케이션의 측면에서 살펴보면 성능 향상을 위해 어플리케이션 소스 레벨 튜닝으로 성능 향상을 꾀하는 것은 분명 한계가 있다. Bug만 잡아도 상당한 성능향상을 가져올 수 있고, 하드웨어적인 확장인 경우에 어플리케이션의 변경이 발생하는 단점을 가지고 있다. 따라서 앞서 기술한 세 가지 관점에서의 문제점들을 상용화된 소프트웨어들을 통해 해결 할 수 있는 방안을 본 논문에서 제안하고자 한다.

### III. 그리드 아키텍처 서버의 구성

#### 3.1 데이터베이스 그리드

##### 3.1.1 O사의 Real Application cluster

고 가용성을 구축해야 하는 크리티컬한 비즈니스의 IT환경의 요구에 따라 O사는 과거 OPS(Oracle Parallel Server)를 지원하였으며 현재 버전에서는 각 노드간 캐시의 일치성을 보장하기 위한 서버간의 통신 방식을 디스크를 이용한 방식에서 초고속 인터커넥트를 이용한 캐시 퓨전(Cache Fusion)으로 변경하면서 고가용성 구현의 완성도를 높였다. 현재 O사의 10g에서 그리드 컴퓨팅을 지원하는 더욱 발전된 RAC(real application cluster)구조를 상용화 하였다. O사의 RAC는 다중 노드를 지원하는 공유 디스크 구조를 사용하여 한층 강화된 HA 솔루션을 데이터베이스에 제공 한다. RAC는 동일 데이터베이스 또는 스토리지를 여러 인스턴스에서 동시에 액세스할 수 있는 장점을 가지고 있으며, 시스템 확장 즉 유기적으로 인스턴스 노드의 추가가 가능하기 때문에 탁월한 로드밸런싱 및 향상된 성능을 구현할 수 있다. 또한 RAC 구조는 모든 노드가 동일한 데이터베이스를 액세스하기 때문에 한 인스턴스에서 장애가 발생해도 데이터베이스에 대한 액세스가 손실되지 않는 장점을 가지고 있다.

##### 3.1.2 M사의 Cluster

M사는 비공유 구조(shared nothing architecture)를 하여 데이터 분할을 사용하는 클러스터링을 제공한다. 이 방법은 특히 데이터웨어하우징 애플리케이션의 확장성에 유용하다. 하지만 이 경우 하나의 노드가 전체 애플리케이션에 영향을 미칠 수 있으므로 HA를 위해서는 모든 노드에 데이터가 균일하게 분산되도록 해야 한다. 또한 각각의 노드의 장애시 데이터의 사용이 불가능하며, 새로운 노드가 추가시 데이터의 재분배(repartition)를 반드시 수행하여야 한다. 따라서 독립서버구조(federated)의 구성은 자동적인 장애극복을 위해서는 추가적인 시스템(witness)이 필요하며, 단하나의 미러서버(mirror server)만을 허용한다.

##### 3.1.3 I사 pureScale

I사는 최근 애플리케이션과 데이터베이스의 고가용성 및 확장성을 지원하는 클러스터링 기술인 DB2 퓨터스케일(pure scale)을 구현한다. DB2의 퓨터스케일은 기존 메인프레임의 디스크 클러스터링 기술인 시스플렉스를 유닉스 환경에서도 이용할 수 있도록 구현한 것으로 중요하고 민감함 업무를 진행시 시스템 확장을 통해 비즈니스 요건을 충족시킬 수 있는 솔루션이다[6].

#### 3.2 어플리케이션 그리드

그리드 컴퓨팅은 기본적으로 수요(demand)와 자원(resource)을 분리(decoupling)하는 것이다. 그리드 컴퓨팅은 수요로부터 분리되어 통합 공유된 IT자원을 다양한 수준의 수요에 대해 동적으로 할당할 수 있는 것을 말한다. 애플리케이션 그리드는 기존에 존재 하지 않았던 전혀 새로운 개념은 아니다. 특히 어플리케이션 서버, JVM, TP모니터는 그리드 컴퓨팅을 가능하게 하는 핵심 기술 중의 하나이다. 이러한 구조는 첫째 각각의 애플리케이션이 명확하게 정의된 스택 상에 구현된다는 점에서 간단명료했고, 둘째 시스템 장애발생시 다른 애플리케이션에 영향을 끼치지 않는다는 점에서 매우 안정적이었다. 하지만 기업의 업무에서 IT가 차지하는 비중이 지속적으로 커지고 애플리케이션의 규모가 확장되면서, 몇 가지 문제점들이 부각되고 있다. 보유 장비의 활용도에 있어서의 비효율성과 수평적 확장의 어려움, 결과적으로 나타나는 성능의 한계 등이 그것이다. 각각의 문제를 살펴보면 먼저 비효율성의 문제이다. 기존의 방식에서 각 시스템은 애플리케이션이 수용해야 할 가장 열악한 상황(worst case), 즉 최대 부하치(peak load)에 맞추어서 구성되어 있다. 최대 부하치를 수용해야 하는 시점이 아주 드물게 일어난다고 해도, 이 때문에 서비스의 문제가 발생하는 것을 허용할 수 없기 때문이다. 두 번째는 유연성의 문제이다. 애플리케이션의 사용량이 증가함에 따라 시스템을 확장할 필요가 발생하게 되고 이를 위해 동일한 용도의 서버 장비를 증설하는 수평적 확장(scale-out)을 해야 하는데, 이 때 별도의 장비를 인수하여 커스터마이징하고 설치하는 일련의 과정은 업무의 신속성을 방해하는 요소 중

의 하나이다. 따라서 애플리케이션 그리드는 IT자원(서버장비, 네트워크 리소스등)과 이들 자원의 소비자 즉 개별 애플리케이션(패키지 소프트웨어), SOA 서비스, 레거시 등을 분리함으로써 자원을 공유하고 이를 효율적으로 배분하는 아키텍처를 제공해야 한다. 이를 통해 처리 용량의 측면에서 전체 시스템의 반응 패턴을 예측 가능한 수준에서도 제어할 수 있어야 한다. [그림 2]와 같이 하드웨어를 얼마나 추가 했을 때 어느 정도의 성능 향상을 보일 것인지를 예측하는 것은 용량 산정 및 투자비용을 절감하는데 반드시 필요하다. 따라서 선형적인 확장성을 가지고 있는 솔루션이 Grid Application 구현에 필요한 기술이라고 할 수 있다.

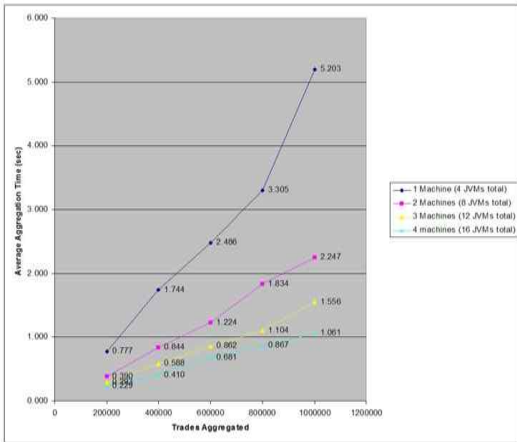


그림 2. 하드웨어 증가에 따른 응답시간

### 3.2.1 애플리케이션 그리드를 위한 기술 요건

애플리케이션 그리드를 구현하기 위해서는 자원의 공유(Share Resources), 공유된 자원의 할당을 제어(Control Allocation), 그리드 전체에 대한 가시성 확보(View Behavior)등이 전제 되어야 한다. 자원의 공유는 클러스터링 기술을 이용하여 이를 구현할 수 있으며, 복수 개의 애플리케이션 또는 SOA 서비스가 복수 개의 서버 상에서 운용되는 환경이 마련된다. 이때 부하 분산(load balancing)은 당연히 필요한 조건이다. 또한 자원이 공유가 되면, 이 자원이 애플리케이션의 수요에 따라 동적으로 할당 할 수 있게 제어해야 한다. 공유된 자원의 할당이 적절히 제어가 되면, 이러한 시스

템 운용 상태를 감시할 수 있는 모니터링 시스템 혹은 대시보드(dashboard)가 필요하다. 그리드를 구축하기 위해 필요한 4가지 요소인 공유, 제어, 감시, 자동화 외에 보다 구체적으로 고려해야 할 사항이 있다. 첫 번째는 그리드를 구성하는 각각의 노드가 안정적이어야 한다. 두 번째는 이들 노드들이 어떻게 클러스터를 구성할 것이냐 하는 것이다. 세 번째는 관리의 측면이다. 애플리케이션 그리드의 관리는 노트레벨, 클러스터 레벨, 그리드 레벨 모두에서 통합적으로 이루어져야 한다.

#### (1) 고가용성(High Availability)

고가용성(High Availability)은 클러스터링, 서비스 마이그레이션 등의 기술을 사용하여 어떤 장애 상황에서든 애플리케이션이 중단되지 않고 서비스 되어야 한다. 아래 [표 1]은 애플리케이션 특성에 따른 가용성에 대한 요건들을 보여 주고 있다.

표 1. 애플리케이션의 가용성의 요건

애플리케이션	가용성 조건
웹 애플리케이션	온라인 쇼핑몰, 예약시스템, 금융 서비스 등의 웹 애플리케이션은 24*7의 가용성을 요구한다.
SOA 애플리케이션	파트너, 고객 등의 다양한 타입의 소비자 유형이 존재하며, 다수의 서비스 제공자 시스템과 연계되어 있는 구조로서, 연계된 모든 서비스 제공자가 동일한 가용성 요건을 갖추어야 한다.
포탈	포탈 애플리케이션은 여러 시스템 및 정보에 접근하기 위한 단일화된 창구로서의 역할을 수행하기 때문에 주변의 다른 시스템들보다 강화된 가용성이 요구된다.
인증 및 권한관리	통합된 인증 및 권한 관리 시스템을 도입한 경우, 이 시스템의 down은 비록 연관된 다른 시스템에 문제가 없다 하더라도 전체적인 가용성의 저하를 수반하게 된다. 따라서 이렇게 공통적인 핵심 기능을 담당하는 시스템의 가용성은 더욱 중요하다.

#### (2) 예측 가능한 성능(Predictable Performance)

애플리케이션의 성능은 두 가지로 정리할 수 있다. TPS(Transaction Per Second)와 응답시간이다. 자바로 작성된 애플리케이션은 GC Time으로 인하여 TPS와 응답시간이 고르지 못하고 튀는 구간이 존재 한다. 일반적으로 성능에 대한 요건 충족은 평균 TPS, 평균 응답시간을 기준으로 하기 때문에 문제가 되지 않을 수 있지만, Trading 시스템이나, 주문, 예약 등의 애플리케이션은 일률적이고 예측 가능한 응답 시간이 더욱 중요

한 요소가 될 수 있다. 이런 요건을 만족해야 하는 애플리케이션에서는 GC Time으로 인해 전체 애플리케이션 수행이 중단되는 시간을 분산하여 애플리케이션에 대한 각 요청이 예측된 응답을 보내 줄 수 있도록 해주어야 한다. 또한 응답 시간의 일관성이 매우 중요한 애플리케이션에 대하여 최대 기대 응답시간을 보장해 줌으로써 서비스의 품질과 신뢰성을 높일 수 있어야 한다.

**(3) 확장성(Scalability)**

애플리케이션의 발전 단계에서 시스템의 규모가 커지게 되면 이에 따른 데이터 계층의 확장이 문제가 된다. 이런 문제를 물리적인 데이터 계층의 확장을 통하지 않고 해결하는 방안이 논리적 데이터 계층을 통한 확장이다. 논리적인 데이터 계층은 데이터를 요구하는 다수의 애플리케이션 및 클라이언트에 대한 요청을 처리하여 물리적 데이터 계층의 부하를 줄임으로써 데이터 요청에 대한 병목 및 확장성의 제약성을 제거하는 역할을 담당한다. 이런 확장성은 증가되는 사용자, 데이터, 트랜잭션에 따라 빠르고 유연하게 대처 할 수 있어야 한다. 즉 확장을 위해 복잡한 구성 과정을 거쳐야 하거나 시스템 전체를 중단하지 않고, 추가된 노드가 바로 그리드의 구성원이 되어 업무를 분배 받을 수 있어야 하며, 이를 통해서 성능 및 자원의 활용이 구성된 모든 노드들에 골고루 분산 되어야 한다. 만약 확장을 위해 복잡한 구성과정을 거쳐야 한다면, 수십 수백 개의 노드들에 대한 구성을 해야 하며, 장애시를 대비한 Failover, Failback에 대한 정책들 또한 수동으로 지정해 주어야 만 하는데, 이런 과정은 오류를 유발 하고 확장성에 제약을 주게 된다, 따라서 사용자, 데이터, 사용률의 증가에 따른 용량 증가에 효과적으로 대응 할 수 있고, 확장에 따른 선형적인 증가를 보장해야 한다.

**(4) 관리성(Manageability)**

애플리케이션 그리드에는 다수의 노드들과 애플리케이션이 존재하게 된다. 제대로 된 관리 기능이 존재하지 않는다면, 확장성으로 인한 장점들을 무색하게 할 만큼의 관리적인 손실이 발생 할 수 있다. 그리드 내의 수백 개의 애플리케이션 서버들에 패치를 적용하거나

구성 변경을 해야 한다고 가정하면, 반복적인 작업의 과정보도 만만치 않겠지만, 수동 작업으로 인한 오류의 가능성을 무시 할 수 없다. 따라서 시스템 구성 변경, 애플리케이션 배포, Fail-Over등의 작업이 사용하기 편한 툴을 통해 이루어지거나, 자동화 되어야 한다.

**3.3 그리드 솔루션 구현을 통한 로드밸런싱**

본 논문에서 제안한 다양한 가용성을 보장하기 위한 Grid 측면에서의 어플리케이션, 데이터베이스의 아키텍처를 살펴보았다. 따라서 상용화된 데이터베이스 측면에서의 그리드 솔루션을 구현하여 엔터프라이즈 환경의 측면에서 그리드 아키텍처를 통한 로드 밸런싱의 목표치를 설정하고, 이의 성능을 검증하여 차후 고가용성을 구현하기 위한 아키텍처로 활용가능할 수 있다.

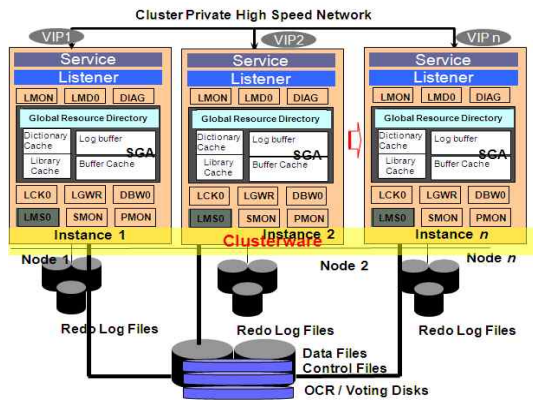


그림 3. 데이터베이스 Grid 구현 아키텍처

따라서 [그림 3]에서와 같이 서로 다른 데이터베이스 인스턴스의 클러스터링을 위하여 상용화 소프트웨어의 클러스터웨어를 이용하여 노드간의 상호 공유 영역을 만들고, 노드간의 데이터의 공유 부분은 캐시퓨전(cache fusion) 메커니즘을 통하여 각각의 클러스터화된 공유영역의 메모리 부분의 데이터 lock 메커니즘 구현하여, cash to cash 직접전송 방법을 선택 하였다. 또한 클러스터 환경위에 각각의 서버에 인스턴스를 설치하여 고 가용성 솔루션을 구현하였다. 또한 향후 수평적인 확장을 할 수 있는 아키텍처로 구현하였다.



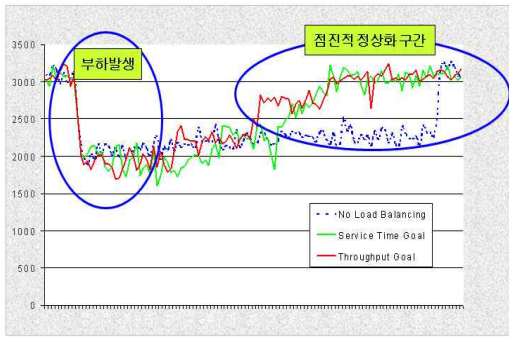


그림 4. Grid 구현을 통한 로드밸런싱 결과

또한 [그림 4]는 구현된 클러스터 환경의 데이터베이스에서 대 단위의 배치작업에 의해 발생하는 부하를 측정하였다. [그림 4]의 결과와 같이 배치 작업이 일어나는 순간에는 성능이 다소 수직 하강하는 결과를 볼 수 있지만 CPU 자원이 상대적으로 여유 있는 서버로 커넥션의 재분배가 되면서 성능이 점진적으로 회복됨을 볼 수 있다. 따라서 이는 서비스 타임의 목표치를 하회하는 매우 우수한 워크 로드 밸런싱이라고 할 수 있다.

#### IV. 결론 및 향후 연구

그리드(Grid)는 단일 제품도 아니고 또한 새로운 표준도 아니다. Grid의 궁극적인 목적은 고품질의 서비스, 유연한 확장성, 효율적인 아키텍처를 위한 새로운 접근법이자 비전이다. 따라서 그리드 기술을 접목시킬 수 있는 대상이 정해져 있는 것이 아니며, 애플리케이션이든 데이터베이스 이든 제한적이지 않다. 어떤 업무 시스템이든 고가용성과 예측 가능한 성능, 확장성, 관리성이 주요한 환경이라면 더욱이 그리드 기술의 접목을 검토해 볼 수 있다고 사료된다. 또한 최근 이슈가 되고 있는 Green IT의 측면에서 IT 자원을 최적화함으로써 기업들이 환경에 미치는 영향을 최소화할 수 있도록 기여하리라 생각된다. 따라서 본 논문을 통해 고찰된 Grid기술을 이용한 고가용성 솔루션을 보다 효율적으로 운영 관리하는 방안을 필두로 보다 진보된 다양한 그리드 기술을 이용한 메모리 기술 및 캐싱(caching) 방법 또는 서버 및 소프트웨어의 서비스 가상화, 어플리케이션, 데이터베이스의 가상 화에 따른 기술들에 대한 연구 및 고찰이 필요하다.

#### 참 고 문 헌

- [1] 김진미, 은기원, 김학영, 지동해, “클러스터링 컴퓨팅 기술”, 1999.
- [2] R. Buyya, High Performance Cluster Computing Vol 1&2, Prentice Hall, 1999.
- [3] 홍태희, 구본준, 김학배, “고가용성 클러스터링 가상서버의 로드밸런서를 위한 고장극복 기법에 관한 연구”, 대한전기학회 하계학술대회, pp.17-20, 2000.
- [4] Gregory, F. Pfister, In Search of Cluster 2nd Edition, Prentice-Hall, 1998.
- [5] <http://dpm.postech.ac.kr/cluster/index.htm>
- [6] [http://www.imaso.co.kr/?doc=bbs%2Fgnuboard\\_pdf.php&bo\\_table=article&page=1&wr\\_id=34395&publishdate=20100101](http://www.imaso.co.kr/?doc=bbs%2Fgnuboard_pdf.php&bo_table=article&page=1&wr_id=34395&publishdate=20100101)
- [7] 마이크로소프트웨어 특집 2-4부, pp.231-234, 2008.
- [8] [http://ko.wikipedia.org/wiki/%ED%81%B4%EB%9D%BC%EC%9A%B0%EB%93%9C\\_%EC%BB%B4%ED%93%A8%ED%8C%85](http://ko.wikipedia.org/wiki/%ED%81%B4%EB%9D%BC%EC%9A%B0%EB%93%9C_%EC%BB%B4%ED%93%A8%ED%8C%85)
- [9] 클러스터 기반 DBMS를 위한 고가용성 클러스터 관리기의 설계 및 구현, 정보과학회, 제 12권, 제1호, pp.21-30, 2006.
- [10] 최재영, 황찬석, “클러스터를 위한 소프트웨어 도구” 정보과학회지, 제18권, 3호, pp.40-47, 2000.
- [11] Oracle Korea Magazine, Vol.57, pp.33-38, 2009.
- [12] 한병진, 김형래, 정창성, “그리드 컴퓨팅을 위한 오토로지 기반의 시맨틱 정보 시스템, 한국 인터넷 정보학회, 10권, 4호, pp.87-103.
- [13] 한국과학기술 정보연구원, “Grid Application 의 QOS 보장을 위한 차세대 인터넷 연동 및 지원연구, pp.6-21, 2002.
- [14] 황준석, 박상욱, 이충희, “Grid Economy 기반

Grid 상용화 및 Business Model을 위한 자원거래 정책 연구, 한국과학기술정보연구원, pp.21-79, 2004.

[15] Globus Toolkit, <http://www.globus.org/toolkit/>.  
 [16] Global Grid Forum, <http://www.ggf.org>  
 [17] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002. [www.globus.org/research/papers/ogsa.pdf](http://www.globus.org/research/papers/ogsa.pdf).  
 [18] W3C. Web Services, <http://www.w3.org/2002/ws/>.  
 [19] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, and D. Snelling, "Open Grid Services Infrastructure", <http://www.ggf.org/documents/GWD-R/GFD-R.015.pdf>  
 [20] The Condor Project, <http://www.cs.wisc.edu/condor/>  
 [21] The myGrid project, <http://www.mygrid.org.uk/>  
 [22] The GridLab project, <http://www.gridlab.org/>  
 [23] The unicore project, <http://www.unicore.org>  
 [24] GT Information Services : Monitoring & Discovery System (MDS), <http://www.globus.org/toolkit/mds/>

• 2003년 3월 ~ 현재 : 배재대학교 전자상거래학과 부교수  
 <관심분야> : XML, 지능정보시스템, 데이터베이스시스템, 전자상거래학

박 준 호(Jun-Ho Park)

정회원



• 2008년 2월 : 충북대학교 정보통신공학과(공학사)  
 • 2010년 2월 : 충북대학교 정보통신공학과(공학석사)  
 • 2010년 3월 ~ 현재 : 충북대학교 정보통신공학과(박사과정)  
 <관심분야> : 분산 데이터베이스 시스템, 센서 네트워크, RFID, 차세대 웹, U-Learning(LMS, LCMS)

유 재 수(Jae-Soo Yoo)

종신회원



• 1989년 : 전북대학교 컴퓨터공학과(공학사)  
 • 1991년 : 한국과학기술원 전산학과(공학석사)  
 • 1995년 : 한국과학기술원 전산학과(공학박사)  
 • 1995년 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사  
 • 1996년 9월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 및 컴퓨터정보통신연구소 교수  
 <관심분야> : 데이터베이스 시스템, XML, 멀티미디어 데이터베이스, 분산 객체 컴퓨팅

저 자 소 개

이 병 엽(Byoung-Yup Lee)

종신회원



• 1991년 2월 : 한국과학기술원 전산학과(공학사)  
 • 1993년 2월 : 한국과학기술원 전산학과(공학석사)  
 • 1997년 2월 : 한국과학기술원 경영정보공학(공학박사)

• 1993년 1월 ~ 2003년 2월 : 대우정보시스템 차장