http://dx.doi.org/10.3745/KIPSTA.2011.18A.4.123

대규모 분산 컴퓨팅 환경에서 확장성을 고려한 실시간 데이터 공급 기법

김 병 상[†] · 윤 찬 현[††]

요 약

본 논문은 원격지간의 연결된 대규모 분산 환경에서 데이터 분석 작업의 실행을 위해 필수적으로 고려되는 데이터 전송 부하를 감소시키는 기법을 제안한다. 계산 노드들이 밀집된 지역 인근에 다수의 데이터 노드를 배치시킴으로서 계산 노드들이 단일 데이터센터가 아닌 자신과 인접한 데이터 노드에 접근하여 작업을 수행함으로서 전송부하를 감소시키고 확장성을 증가시키는 것이 가능하다. 따라서 본 논문은 지역적으로 분산된 데이터 노드들의 데이터 처리율을 기반으로 실시간 데이터 공급을 수행함으로서 전송 지연을 최소화 할 수 있는 이론적인 모델과 시뮬레이션을 통한 성능 평가를 수행한다. 제안된 기법은 PRAGMA 그리드 테스트베드에서 실험을 통하여 성능의 우수성을 검증하였다.

키워드 : 분산 데이터 분석, 데이터 공급 기법, 안정 상태 스케줄링, 부하 분할 이론

Scalable Data Provisioning Scheme on Large-Scale Distributed Computing Environment

Byungs-Sang Kim[†] · Chan-Hyun Youn[††]

ABSTRACT

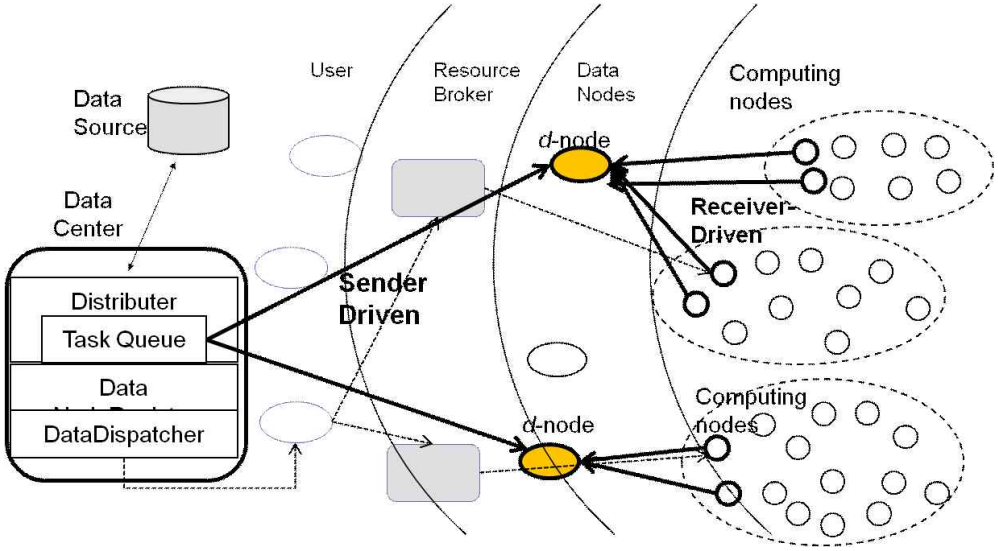
As the global grid has grown in size, large-scale distributed data analysis schemes have gained momentum. Over the last few years, a number of methods have been introduced for allocating data intensive tasks across distributed and heterogeneous computing platforms. However, these approaches have a limited potential for scaling up computing nodes so that they can serve more tasks simultaneously. This paper tackles the scalability and communication delay for computing nodes. We propose a distributed data node for storing and allocating the data. This paper also provides data provisioning method based on the steady states for minimizing the communication delay between the data source and the computing nodes. The experimental results show that scalability and communication delay can be achieved in our system.

Keywords : Distributed Data Analysis, Data Provisioning, Steady State Scheduling, Divisible Load Theory

1. 서 론

그리드 혹은 클라우드와 같은 인터넷 기반의 글로벌 컴퓨팅 환경이 성숙됨에 따라 인터넷 상에 흩어져 있는 수천 혹은 수만 대의 컴퓨팅 자원을 동시에 활용하는 것이 가능해지고 있다. 특히, 고에너지물리분야의 LHC 실험데이터[1] 생명정보분야의 DNA서열검색[2] 및 경제학 분야에서의 데이터 마이닝[3] 분야는 대용량(Peta-Scale) 데이터의 분석이 필수적이며 이와 같은 환경을 활용하는 것이 매우 효과적이라 할수 있다. 하지만 컴퓨팅 자원이 규모가 커지더라도 선형적인 확장성을 성취하지 못하는 경우 예상되는 성능을 보장할 수 없을 뿐만 아니라 계산 자원의 낭비를 초래하게 된다. 데이터 집약적인 작업의 경우 성능 저하는 데이터를 계산 노드로 이동할 때 걸리는 전송 부하에 기인한다. 이와같은 전송시간은 (i) 단위 작업당 데이터의 크기, (ii) 작업분배기의 위치 (iii) 작업을 수행하는 계산 노들의 수에 의존성을 가진 랜덤과정이다. 전송 부하를 줄이기 위하여 실시간 데이터 분배 방식이 이용되며 분배기-작업자(MWM: Master to Worker Model) 모형이 대표적이라 할 수 있다. 중앙집중식 작업 분배 방식에서 데이터 분배기는 계산 노드의 성능을 기반으로 작업을 계획하는 송신자기반(Sender-Driven)방식이 일반적이다. 부하 분할 이론 (Divisible Load

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단-미래기반기술개발사업(첨단융복합분야)의 지원을 받아 수행된 연구임[N01100428].
† 준 회 원 : 한국과학기술원 정보통신공학과 박사과정
†† 종신회원 : 한국과학기술원 전자및전기공학과 교수
논문접수 : 2011년 2월 18일
심사완료 : 2011년 3월 4일



(그림 1) 데이터 노드(d-node)기반 데이터 분배 환경

Theory)[4]은 선형 혹은 비선형 기법을 이용하여 정해진 계산 노드에 최적의 데이터 부하(load)를 할당하는 것을 목표로 한다. 하지만 작업의 실행 환경의 규모가 증가하고 작업의 크기 또한 증가할 경우 모든 계산 노드의 성능을 예측하는 것은 거의 불가능하다. 따라서 이러한 정적 스케줄링 기법은 한계가 있다. 반면 수신자기반의 작업 계획기법은 계산 노드가 스스로 자신의 수행성능에 기반을 두어 수행해야 할 작업을 분배기에 요청하여 획득하는 방식을 사용한다. 이러한 접근 방식은 시스템의 구현이 단순할 뿐 아니라 불확실성이 높은 동적인 환경에서도 적용이 용이하다. 하지만 수신자기반의 작업 계획 기법은 작업의 사전계획이 이루어지지 않기 때문에 전송지연시간을 피할 수가 없다. 그리드 환경에서 작업 분배기로 활용되는 DIANE [5], PANDA [6] 등이 대표적인 구현물이라 할 수 있다.

본 논문에서는 위의 두가지 방식을 조합한 하이브리드모형을 통하여 전송 부하를 줄임과 동시에 확장성을 증가시키고자 한다. 계산 자원 중에 일부를 데이터 노드로 활용하여 소량의 데이터를 저장하고 인근의 계산자원에서는 가장 인접한 데이터 노드에 접근하여 단위 작업을 위한 데이터를 획득하게 된다. 이러한 메모리기반의 저장 구조는 확장성을 증가시키고 및 전송부하를 줄일 수 있지만 협소한 저장 공간을 제공하기 때문에 동시에 저장할 수 있는 데이터의 양이 제한될 수밖에 없다. 따라서 본 논문에서는 데이터 노드에서의 데이터 소비율(처리율)에 기반한 실시간 감시를 통한 동적인 데이터 공급 기법을 제안하여 안정적인 데이터 노드를 유지하고 확장성을 보장하고자 한다.

2. 확장성을 고려한 데이터 분석 환경

모형의 기술에 앞서 환경 및 노드의 특성에 따른 분류를 하고자 한다.

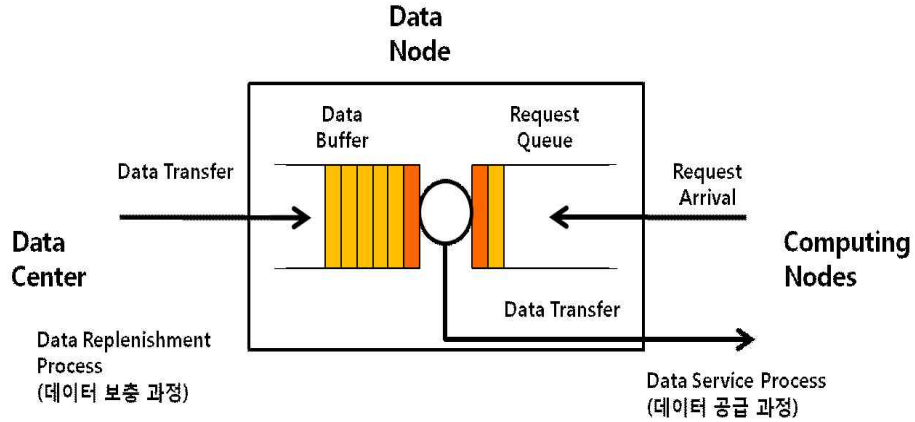
- 데이터 노드 - 데이터 노드는 앞에서 설명한 데이터 분배자로서 단위 작업의 실행을 위한 입력 데이터를 보유하고 있는 임시 저장소이며 해당 데이터를 계산 노드로 전송해 주는 데이터 서버의 역할을 수행한다.
- 계산 노드 - 계산 노드는 데이터 노드로부터 분석하고자 하는 데이터를 요청하고 획득하여 데이터 분석 작업을 수행 후 결과를 다시 데이터 노드에 저장하는 일을 담당한다. 계산 노드는 자신과 인접해 있는 데이터 노드를 인지할 수 있다고 가정한다.

(그림 1)에서 보는것과 같이 데이터 센터와 계산 노드들간의 위치한 데이터 노드는 분석하고자 하는 데이터를 실시간으로 전달 받는다. 따라서 주변의 계산 노드에서는 데이터 노드에서 데이터를 획득하여 작업을 수행하게 된다. 작업 분배기는 데이터 노드내의 데이터 처리율을 기반으로 주기적으로 데이터를 공급하여 데이터 노드의 가용성을 유지한다. 따라서 작업 분배기와 데이터 노드간에는 송신자기반의 데이터 전달 기법이 적용되며 데이터 노드와 계산 노드들간에는 수신자 기반의 데이터 전달 기법이 적용된다. 이러한 분할 기법은 각 데이터 노드 내에 데이터를 안정적으로 유지시키는 것이 핵심이다. 계산노드에서 작업을 요청했을 경우 만약 데이터 노드에 데이터가 없을 경우 계산 노드는 작업을 수행하지 못하고 기다려야 하기 때문이다. 데이터 노드에 도착하는 계산 노드의 작업 요청의 수는 데이터 노드에 포함되어 있는 계산 노드의 수와 단위 작업당 수행시간의 길이에 따라 도착률이 결정된다. 3장에서는 데이터 노드의 데이터 요청 과정과 데이터 공급기법에 대하여 서술한다.

3. 실시간 데이터 공급 기법 및 전송 부하 분석

3.1 데이터 요청 및 공급 과정

각각의 계산 노드는 데이터 노드에서 데이터를 가지고와



(그림 2) 데이터 노드 및 데이터 전달 과정

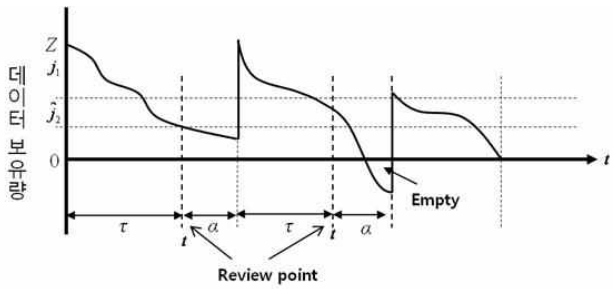
서 단위 작업을 수행한다. 작업의 실행이 완료되면 결과를 다시 데이터 노드에 저장하고 또 다른 데이터를 획득하여 작업을 수행한다.

(그림 2)는 데이터 노드내의 데이터의 보충 및 공급과정을 보여주고 있다. 다수의 계산 노드가 단일의 데이터 노드에 접근하여 동일한 데이터 전송 요청을 하게 된다. 따라서 데이터 공급과정에서 데이터 노드는 계산 노드들의 요청을 도착과정으로 하고 데이터 전송시간을 서비스 과정으로 하는 대기행렬 시스템으로 간주 할 수 있다. 하나의 데이터 노드에 도착하는 계산 노드의 요청 간격은 단위 작업 수행 시간이 지수분포를 따르지 않는 한 포아송 과정을 따른다고 할수 없다. 하지만 데이터 노드에 접근하는 계산 노드의 수가 충분히 크다고 가정할 경우 Palm-Khintchine 이론 [7]에 의해 데이터노드의 데이터 요청 분포를 계산 노드의 개수와 작업 수행 시간에 의해 결정되는 포아송 분포로 정의하는 것이 가능하다. 따라서 계산노드의 수와 단위 작업의 평균 수행시간을 알지 못하더라도 데이터 노드에 도착하는 계산 노드의 요청 간격 및 요청률만으로 데이터 노드의 성능 분석이 가능하다.

3.2 안정 상태 데이터 공급 과정 분석

안정 상태에서의 데이터 공급 과정이란 데이터 노드 내에 저장된 데이터의 양을 일정하게 유지하는 기법을 의미한다. 데이터센터에서는 주기적으로 데이터노드를 감시하여 데이터의 양을 확인하고 필요한 만큼의 데이터를 공급한다. 따라서 데이터 공급 과정은 계산 노드들에서의 데이터 요청률(λ_d), 데이터 센터로부터의 단위 데이터 전송 시간 (π) 및 데이터 공급을 위한 검사 주기(τ)에 의해서 결정되는 확률 과정이다.

(그림 3)은 데이터 노드내의 데이터 보유량을 시간축상에서 나타내고 있다. 최초의 데이터의 보유량을 Z 라 할때 시간이 흐름에 따라 보유량은 감소할 것이며 검사 시점 t 에서 보충해야할 데이터의 양은 주기 τ 동안의 소비량 $j_1(t)$과 소비량을 보충하는데 걸리는 시간동안의 소비 예측치인



(그림 3) 데이터 노드내의 데이터 보유량의 샘플 경로

$\hat{j_2}(t)$의 합으로 표현될수 있다. 즉, 시간 t 에서의 데이터 보충량 $J(t)$는 다음과 같다.

$$J(t) = j_1(t) + \hat{j_2}(t). \quad (1)$$

$j_1(t)$의 값은 검사 시점 t 에서의 실측값이며 이것을 기반으로 조달시간을 예측할 수 있다. $j_1(t)$만큼의 데이터를 전송하기 위해 필요한 예측 조달 시간 α_0 라고 하면 $\alpha_{0=}j_1(t)/\pi$ 이다. 또한 조달시간동안에 소모되는 데이터의 양은 위의 조달 시간 α_0 의 재귀적 방법으로 표현된다. 즉, α_i를 조달기간 α_{i-1}동안에 소모될 데이터의 조달 시간의 예측치로 가정하면 $\alpha_{i=}\lambda_d\alpha_{i-1}/\pi$ 이며 주기 τ 동안 소비된 데이터의 보충을 위한 예측 조달 시간 α는

$$\alpha = \sum_{i=0}^{\infty}\alpha_i = \frac{j_1(t)}{\pi - \lambda_d} \quad (2)$$

로 주어진다. 따라서 데이터 노드에서 조달 시간동안의 소비 예측치 $\hat{j_2}(t)$는 시간 α 동안의 소비율로 정의된다.

$$\hat{j_2}(t) = \lambda_d\alpha = \frac{\lambda_d j_1(t)}{\pi - \lambda_d}. \quad (3)$$

결과적으로, 검사 시점 t 에서 보충해야 할 총 데이터의 양 $J(t)$의 예측값은 다음과 같다.

$$J(t) = j_1(t) + \frac{\lambda_d j_1(t)}{\pi - \lambda_d} = \frac{\pi j_1}{\pi - \lambda_d}. \quad (4)$$

비록 추정값에 의해 데이터를 조달한다고 하더라도 데이터의 요청은 불확실성을 내재한 확률적 분포로 나타나기 때문에 실제 계산 노드가 데이터를 요청하는 순간 데이터 노드에 데이터가 없는 경우가 발생할 수 있다. 데이터 노드의 데이터가 없을 경우 계산 노드의 요청은 대기 장소에서 기다려야 한다. 데이터 노드에 데이터가 없을 확률은 최초 보유량(Z), 검사 시점(τ) 및 계산 노드들의 요청률(λ_d) 에 의해 결정된다. P_{empty}를 노드에 데이터가 없을 확률, $D(t)$를 시간 t 동안의 요청의 수로 정의면

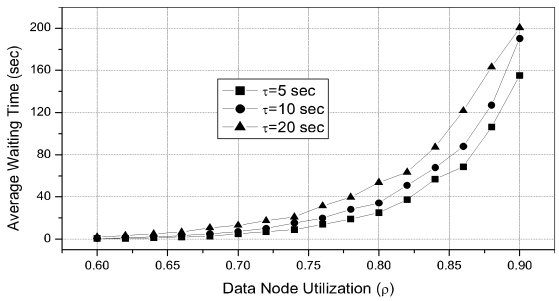
$$\begin{aligned} & P_{empty} \\ &= P(D(\tau) > Z) \\ &\quad + P(D(\alpha) > Z - j_2(t) | P(D(\tau) \le Z)) \\ &= 1 - P(D(\alpha) \le Z - j_2(t)) P(D(\tau) \le Z) \\ &= 1 - \sum_{i=1}^{Z-j_2(t)} \frac{(\lambda_d \alpha)^i}{i!} \sum_{j=1}^{Z} \frac{(\lambda_d \tau)^j}{j!}. \end{aligned} \quad (5)$$

위의 식(5)로부터 우리는 요청률에 따라 보유량 및 검사주기를 적절하게 결정하여야 한다. 보유량이 크면 클수록 P_{empty}는 줄어들지만 그만큼 저장해야할 메모리 공간을 소모하기 때문이다.

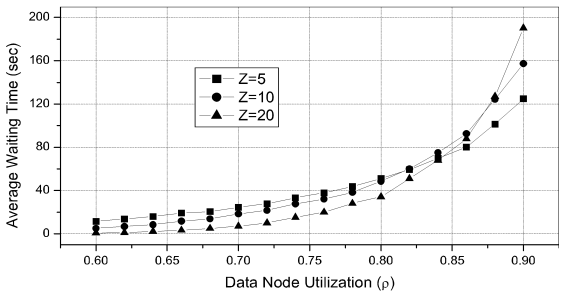
4. 성능 평가

4.1 대기 시간 평가

계산노드가 데이터 노드에서 데이터 획득하는 과정은 앞에서 설명한 대기행렬과정으로 모델링할 수 있다. 따라서 우리는 대기시간의 성능 평가를 위하여 시뮬레이션 기법을 이용한다. 시뮬레이션 툴은 SimJava[8]를 활용하였다. 우리는 데이터 요청 간격과 데이터 전송 과정은 지수분포를 따르는 랜덤과정으로 가정한다. 이것은 대규모 그리드 환경의 이질성을 반영한다. 또한 데이터노드의 이용률(Utilization) $\rho = \lambda_d / \pi$ 로 정의한다. (그림 4(a)) 는 최대 데이터 보유량을 20으로 고정하고 이용률(ρ)과 검사주기(τ)의 변화에 따른 계산노드의 데이터획득 대기시간의 분포를 나타내고 있다. 그림에서 보는것과 같이 동일 검사주기에서 이용률이 증가할수록 대기시간은 증가한다. 또한 검사주기가 길수록 대기시간의 분포가 높이 위치하는 것을 볼 수 있다. 한편 (그림 4(b))는 검사시간(τ)을 10 sec로 고정하고 이용률(ρ)과 최대보유량(Z)의 변화에 따른 계산노드의 데이터획득 대기시간의 분포를 나타내고 있다. 동일한 최대보유량일 경우 이용률이 올라갈수록 대기시간은 함께 증가하는 것을 볼 수 있다. 하지만 특정 이용률 이상에서는 오히려 최대 보유량이 높을수록 대기시간이 높아지는 것을 볼수 있다. 이것은 이용률이 일정 수위 이상 올라가면 최대보유량을 채우기 위한 소비예측치가 높아지고 이것은 결국 감시주기의 길이를 기하급수적으로 높이기 때문에 나타나는 현상이다.



(a) 검사주기 및 이용률 변화 대기 시간 분포



(b) 최대 보유량 및 이용률 변화에 따른 대기 분포

(그림 4) 검사주기 및 최대보유량에 따른 대기 시간

4.2 실험 환경 구성

이번 절에서는 실제 그리드 컴퓨팅 인프라를 통하여 제안된 기법의 효율성을 검증하고자 한다. 실험을 위하여 우리는 PRAGMA(Pacific Rim Applications and Grid Middle ware Assembly) [9] 자원을 활용한다. PRAGM는 환태평양 지역의 국가들간의 그리드 응용의 실행 및 미들웨어를 연구하는 조직으로서 총 800대 이상의 계산 노드의 수를 운영하고 있다.

〈표 1〉 자원 구성

Country	Data Nodes	Nodes
Japan	sakura.hacc.jp	8
Japan	tea01.exp-net.osaka-u.ac.jp	20
Malaysia	aurora.usmgrid.myren.net.my	20
Puerto Rico	komolognma.ece.uprm.edu	20
USA	rocks-52.sdsc.edu	20
Thailand	sunyata.thaigrid.or.th	12
Korea	luke.kisti.re.kr (task server)	-

우리는 이중에서 총 6개의 국가에 100대의 계산 노드를 활용하였다. 위의 <표 1>은 실험을 위한 자원의 구성을 보

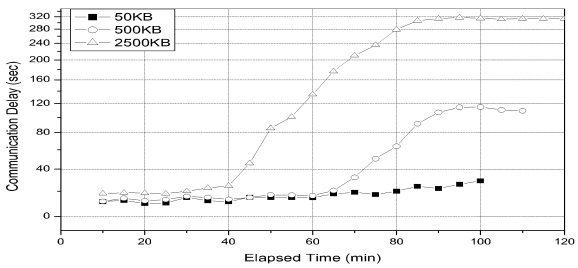
여주고 있다. <표 2>는 실험을 위한 파라미터들과 실험값을 보여주고 있다. 제안된 구조의 확장성을 위하여 우리는 각 계산노드에서 총 8개의 독립된 프로세스를 생성하였다. 따라서 매 2분마다 4대의 계산노드에 총 32개의 독립된 계산 프로세스를 생성하였으며 총 100대의 노드에 최대 800개의 계산 스레드를 생성하도록 하였다. 각 계산 스레드는 KISTI에 있는 작업 분배기(task server)를 통하여 데이터를 공급받는다. 성능 비교를 위하여 우리는 데이터 노드를 사용하지 않는 기법(legacy)와 제안된 기법(proposed)을 비교 실험한다.

〈표 2〉 실험 파라미터들

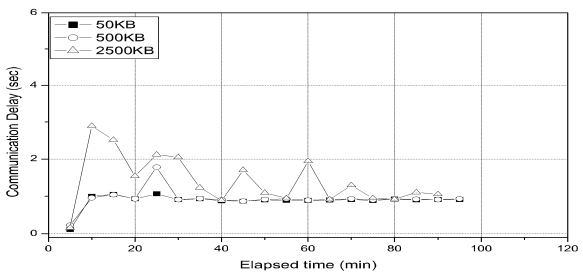
Parameters	Values
Maximum number of nodes	100
Maximum number of computing process	800
Total number of tasks	9000
input data size	50KB, 500KB; 2500KB
Average execution time	240 sec
Applied scheduling algorithms	Legacy, Proposed

4.3 계산 노드 증가에 따른 전송 지연 평가

(그림 5(a))에서 보는바와 같이 기존의 방식은 특정 시간 이후 전송 지연이 급격히 증가하는것을 볼수 있다. 데이터의 크기가 클수록 증가시점이 더 짧아지고 있음을 알 수 있다. 또한 그림 5(b)에서 보는 것과 같이 제안된 기법에서는 계산 노드가 증가하더라도 전송 지연을 거의 경험하지 않는다. 특히 이 기법은 데이터의 크기와도 크게 관계없이 일정한 전송시간을 유지하는 것을 볼 수 있다.



(a) Legacy Algorithm



(b) Proposed Algorithm

(그림 5) 전송 시간(Distributed Cost) 비교

5. 결 론

본 논문에서는 다수의 데이터 노드를 활용하여 전송 부하를 줄이고 계산 노드의 증가에 유연하게 적응할 수 있는 예측 기반의 데이터 공급 기법을 제시하였다. 특히 데이터 노드의 데이터 보유량의 변화 및 대기 시간의 분석을 위한 이론적 연구를 포함하여 시뮬레이션 및 실험을 통하여 제안된 기법에 대한 성능 분석을 하였다. 시뮬레이션 결과는 확률적 안정 상태에서의 데이터의 요청률과 조달율에 근거한 최적의 데이터 보유량 및 대기시간을 이론적 분석과 함께 비교 검증하였다. 또한 대규모의 그리드 테스트베드상에서 제안된 기법을 적용하여 전송 지연 시간을 기존의 방법과 비교하였다. 실험 결과에서는 원격지간의 분산된 작업 실행 환경에서 기존의 기법보다 전송지연시간을 획기적으로 감소시키는것을 확인할 수 있었다.

참 고 문 헌

[1] J. Andreeva, S. Campana, F. Fanzago, and J. Herrala, ''High-Energy Physics on the Grid: the ATLAS and CMS Experience,'' Journal of Grid Computing, Vol.6, No.1, pp.3-13, Mar., 2008.

[2] Y. Asim and J. J. Dongarra, ''Biological sequence alignment on the computational grid using the GrADS framework,'' Future Generation Computer Systems, Vol.21, No.6, pp.980-986, June, 2005.

[3] P. Luo, K. Lu , Z. Shi , and Q. He, ''Distributed Data Mining in Grid Computing Environments,'' Future Generation Computer Systems, Vol.23, No.1, pp.84-91, Jan., 2007.

[4] C. Banino, O. Beaumont, L. Carter, J. Ferrante, A. Legrand, and Y. Robert, ''Scheduling Strategies for Master-Slave Tasking on Heterogeneous Processor Platforms,'' IEEE Trans. Parallel Distributed Systems, Vol.15, No.4, pp.319-330, 2004.

[5] Moscicki and T. Jakub, ''DIANE - Distributed analysis environment for GRID-enabled simulation and analysis of physics data,'' Proc. IEEE Nuclear Science Symposium Conference Record, Vol.3, pp.1617-1620, 2003.

[6] https://twiki.cern.ch/twiki/bin/view/Atlas/PanDA

[7] David R. Cox, et.al, "The theory of stochastic processes" Chapman & Hall/CRC, 2001.

[8] The SimJava Tutorial, http://www.dcs.ed.ac.uk/home/hase/simjava/

[9] Pacific Rim Applications and Grid Middleware Assemply, http://www.pragma-grid.net/

김 병 상

e-mail : bs.kim@kaist.ac.kr

2002년 동국대학교 산업공학과(학사)

2004년 한국정보통신대학교 전자통신공학 (공학석사)

2006년～2009년 한국과학기술정보연구원 연구원

2004년～현 재 한국과학기술원 정보통신공학과 박사과정

관심분야 : 분산 시스템, 대용량 작업 스케줄링, 그리드, 클라우드 컴퓨팅 및 응용 환경

윤 찬 현

e-mail : chyoun@kaist.ac.kr

1981년 경북대학교 전자공학과(학사)

1985년 경북대학교 전자공학(공학석사)

1994년 일본 Tohoku 대학 전자통신공학 (공학박사)

1986년～1997년 KT 네트워크 연구소 연구팀장

1997년～2009년 ICU 공학부 교수

2003년～2004년 MIT 방문 교수, MIT-HST 연구원

2009년～현 재 KAIST 전기및전자공학과 교수

2002년～현 재 KAIST 그리드 미들웨어 센터장

2009년～현 재 IEICE 한국지회 컴퓨터분야 위원장

2009년～현 재 한국정보처리학회 논문지편집위원장

2009년 공공클라우드협의회 의장

2010년 정부통합전산센터 클라우드 구축협의회 위원장

2010년～클라우드서비스협회 자문위원

2011년～현 재 한국정보처리학회 부회장

관심분야 : 고성능 컴퓨팅 구조 및 응용서비스, 컴퓨팅 미들웨어, 클라우드, 그리드 컴퓨팅, Biomedical 응용 시스템 등