

클라우드 컴퓨팅 연구 동향

- 김창현(한양대학교 컴퓨터공학과)
- 이원주(인하공업전문대학 컴퓨터정보과)
- 전창호(한양대학교 컴퓨터공학과)

1. 서론

가트너는 클라우드 컴퓨팅을 ‘인터넷 기술을 활용하여 다수의 고객들에게 높은 수준의 확장성을 가진 IT자원들을 서비스로 제공하는 컴퓨팅’으로 설명한다. 또한 포레스터 리서치는 ‘공통적인 특징으로 표준화된 IT기반 기반들이 IP(Internet Protocol)을 통해 제공되며, 언제나 접근이 허용되고, 수요의 변화에 따라 가변적이며, 사용량이나 광고에 기반한 과금모형을 제공하며, 웹 또는 프로그램적인 인터페이스를 제공하는 컴퓨팅’으로 정의하고 있다. 이 같은 정의에 의하면 클라우드 컴퓨팅은 인터넷을 통한 IT자원의 온디맨드 아웃소싱 서비스로 볼 수 있다.

클라우드 컴퓨팅 환경에서 사용자들은 애플리케이션, 스토리지, OS, 보안 등 필요한 IT자원을 원하는 시점에 원하는 만큼 선택해서 사용할 수 있으며, 사용량에 기반하여 대가를 지불한다. 따라서 클라우드 컴퓨팅을 이용하는 일반사용자, 어플리케이션 개발자, 그리고 서비스 제공자들은 기존처럼 자체적인 컴퓨팅 자원을 보유하지 않아도 되기 때문에 초기 시스템 구입, 운영, 유지보수에 대한 비용을 절감할 수 있다. 게다가 단기간에 많은 컴퓨팅 자원을 운용해야 할 때에는 기존처럼 자원에 대한 추가적인 투자없이 자원을 유동성 있게 조절하여 사용할 수 있다. 이 같은 유용성에 비추어 볼 때 클라우드 컴퓨팅의 수요는 앞으로 크게 늘어날 것으로 전망된다.

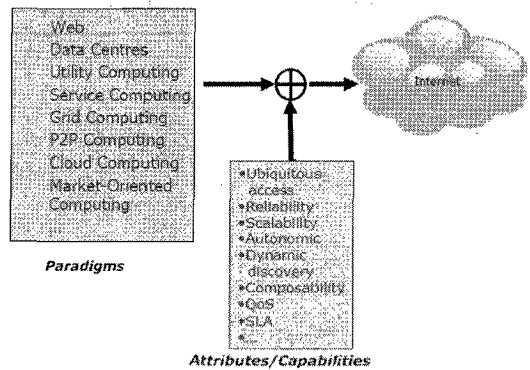


그림 1. 클라우드 컴퓨팅

클라우드 컴퓨팅은 보는 관점에 따라 다소 다를 수 있겠지만 어느 날 갑자기 생겨난 새로운 기술이라기 보다는 그림 1처럼 기존에 있던 여러 기술들이 목적에 맞게 구성되어, 새롭게 재조명 된 것이라 할 수 있다. 기존의 기술들로는 그리드 컴퓨팅, 유틸리티 컴퓨팅, 가상화 등이 있으며, 각 분야 별로 오랜 기간 연구가 진행되어 왔다. 그러나 이 기술들이 클라우드 컴퓨팅을 위해 구성되어 사용될 때 클라우드 컴퓨팅이라는 새로운 패러다임에 따른 다양한 요구가 생겨났으며, 산학 여러 기관에서 이러한 요구를 충족시키기 위하여 활발히 연구가 진행되고 있다.

II. 관련 연구

클라우드 컴퓨팅은 제공하는 서비스에 따라 공개 클라우드(public cloud)와 사설 클라우드(private cloud)로 나눌 수 있다. 공개 클라우드는 pay-as-you-go 방식으로 돈을 지불하고, 지불한 만큼 컴퓨팅 자원을 사용하는 것이다. 이는 유틸리티 컴퓨팅의 개념이며 누구든지 자원이 필요하면 사용할 수 있다. 아마존의 웹 서비스와 구글의 AppEngine, 마이크로소프트의 Azure서비스가 공개 클라우드 해당한다. 그에 반하여 사설 클라우드는 비즈니스 데이터센터와 같이 특수 목적으로 사용되는 것을 말한다[1]. 물론 사설 클라우드도 서비스 제공형태에 따라 유틸리티 컴퓨팅 방식이 사용될 수 있다. 클라우드 컴퓨팅에 대한 대부분의 연구가 공개 클라우드 기반으로 이루어지고 있기 때문에 앞으로 언급할 클라우드 컴퓨팅의 의미는 공개 클라우드로 제한한다.

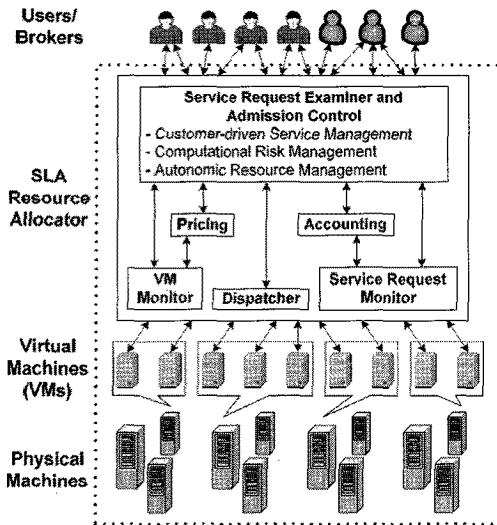


그림 2. 클라우드의 구조

클라우드 컴퓨팅은 그림 2처럼 물리적 메모리와 가상 머신, 리소스 할당자로 구성된다. 유저가 작업을 처리하기 위해 사용할 자원의 capacity를 정하고 이를 요청하면 서비스 제공자는 물리적인 자원을 유저가 원하는 capacity에 맞게 SLA(service level agreement)를 고려하여 가상화시켜 제공한다. 가상화된 자원은 물리적 자원처럼 고정적인 capacity

에 제한받지 않으므로 기존의 고정 자원 임대방식보다 보다 유연한 자원할당을 가능하게 한다. 그리고 사용자는 할당받은 capacity만큼의 비용을 지불한다.

이러한 구조와 특징에 기반하여 클라우드의 운용비용을 절감하거나 적은 비용으로 서비스의 성능을 향상시키기 위한 다양한 연구들이 진행되어 왔다. 본 장에서는 이들 중 몇 가지 연구에 대하여 간략하게 살펴본다.

1. Resource allocation

클라우드 컴퓨팅에서 유틸리티 컴퓨팅 개념은 서비스 제공자와 사용자 모두에게 중요한 고려사항이다. 사용자는 서론에서 언급한 바와 같이 시스템 구입, 운영, 유지보수에 대한 비용을 절감하고, 주어진 일을 수행하기 위해 최소한의 비용으로 자원을 임대하길 원할 것이다. 그리고 제공자는 클라우드의 운영에 최소한의 비용을 들이고, 더 많은 사용자에게 자원을 임대해 주기를 원할 것이다. 양측 모두 최소한의 비용으로 최대한의 효과를 목적으로 한다는 것에 공통점이 있다. 이는 리소스 할당의 문제와 관련이 깊다.

[2]에서는 사용자관점에서 가장 합리적인 자원 임대모델을 연구하였다. 클라우드에서 사용자는 자원 인스턴스를 돈을 주고 임대한다. 사용자는 할당받은 리소스의 Fully capacity를 사용하지 않더라도 리소스에 대한 비용을 지불해야 되므로 할당받은 자원이 Fully capacity를 초과하지 않는 한도에서 다른 사용자들과 공유할 수 있다. 자원의 공유는 사용자 측면에서 자원 임대비용을 낮출 수 있기 때문에 좋은 방법이다. 그러나 서비스 제공자측면에서는 자원의 임대수익이 줄어들기 때문에 제공자는 자원 공유에 대한 지침서 또는 최적화 방법을 제공하지 않을 가능성이 높다. 따라서 이 연구에서는 사용자의 자원 임대비용을 줄이기 위하여 자원을 공유하는 사용자들이 지불해야하는 Best/worst social cost를 내쉬 게임이론을 사용하여 모델링하였다.

[3]에서는 가상화된 자원 인스턴스의 재 할당에 대한 연구를 수행하였다. 가상화된 자원 인스턴스들은 물리적인 자원과 독립적이다. 다시 말해, 하나의 물리적 자원을 두 개 이상의 논리적 가상 자원으로 나눌 수도 있고, 두 개 이상의 물리적 자원을 하나의 가상 자원으로 묶을 수도 있다. 이러한 가상 자원의 형태는 클라우드 사용자의 요구에 따라 결정된다. 예를 들어 서버 컴퓨터 1대의 처리능력이 100이고, 20의 처리용량을 필요로 하는 사용자 5명이 있다면, 서버의

자원을 5개의 가상자원으로 나누어 각각 임대해줄 수 있다. 이때 물리적 자원과 가상머신의 자원은 1대 1로 매핑되지 않기 때문에 경우에 따라 서로 연관되어 동작하는 가상머신이 물리적으로 다른 서버에 할당 될 수 있다. 이 연구에서는 이러한 가상머신들이 서로 관련되는 작업을 할 때, 다시 말하여, 가상 머신들 사이에 통신이 필요할 때 발생할 수 있는 통신 오버헤드를 개선하였다. 서로 관련이 있는 가상 머신들이 물리적으로 다른 서버에 할당되었을 때 통신 오버헤드가 발생한다. 따라서 그림 3처럼 가상머신의 capacity(CPU, Network, Memory)를 고려하여 이들을 하나의 머신으로 이주시킨다면 통신 오버헤드는 감소할 것이다.

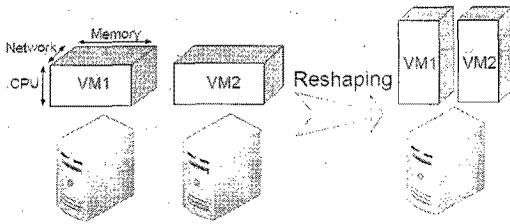


그림 3. 가상머신의 Reshaping

그림 3처럼 가상머신을 다시 재 할당하는 과정을 **reshaping**이라고 한다. **reshaping**은 목적에 부합하는 형태로 가상머신을 재 할당 하는 절차를 모두 포함하며 여러 가지 재 할당 정책에 따라 서로 다른 형태로 가상머신을 구성할 수 있다. 이 연구에서는 가상머신 간의 통신 오버헤드를 감소시키고 클라우드 운영비용을 감소시키기 위하여 관련 가상머신들을 물리적으로 동일한 서버에 위치시키기 위해 **reshaping**을 수행하였다.

그림 4는 **reshaping** 전후의 통신오버헤드를 비교한 것이며 **reshaping** 전 보다 통신 오버헤드가 감소하였음을 확인할 수 있다. 또한 **reshaping** 후에는 그 전보다 자원을 사용하지 않는 서버의 수가 늘어나고, 이러한 서버의 운영을 중지할 수 있다. 따라서 서비스 제공자 측면에서는 사용자들에게 동일한 서비스를 제공함과 동시에 클라우드 운영에 소모되는 비용을 절감할 수 있다.

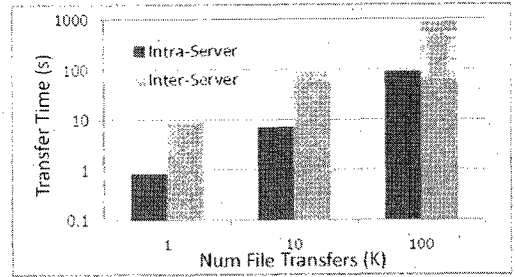


그림 4. 통신 오버헤드 비교

Reshaping은 가상머신의 이주성능에 많은 영향을 받는다. 가장 머신 이주는 현재 활발히 연구되고 있는 주제이며 클라우드에 많은 장점을 제공한다. 유지 보수가 필요할 서버에 대해 가상 머신을 다른 곳으로 이주 시킬 수 있고, 앞서 살펴 본 바와 같이 가상 머신의 성능을 향상 시킬 수 있으며 운영비용을 절감 할 수 있다. 또한 특정 가상머신에서 에러가 발생하여 물리적으로 같은 서버에 위치한 다른 가상머신에 영향을 줄 수 있을 경우 머신들을 다른 곳으로 이주 시킬 수 있으며, 가상 머신들을 적절히 재 할당하여 부하 밸런스를 조절 할 수도 있다.

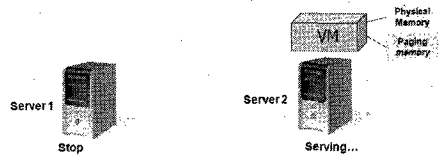
가상 머신 이주는 이러한 많은 장점을 제공하지만 클라우드에 적용을 위해서는 아직 해결해야 할 과제들이 많다. 그 중 이주 시 발생하는 콜 타임은 특히 중요한 해결 과제이다. 가상머신이 이주할 때 가상머신은 서비스를 중지하고, 이주 완료 후 다시 서비스를 재개하기 때문에 이때는 서비스가 제공되지 않으며 이 시간을 콜 타임이라고 한다. 따라서 콜 타임 동안에는 사용자의 작업을 처리 할 수 없으므로 클라우드의 QoS(Quality of Service)에 부정적인 영향을 끼칠 수 있다. 사용자는 클라우드에서 자원을 임대할 때 처리할 작업의 규모를 고려하여 가장 먼저 QoS를 고려한다. 따라서 사용자의 QoS를 보장할 수 없는 클라우드는 제대로 된 서비스를 제공하지 못한다고 할 수 있다.

가상 머신은 이주 레벨에 따라 **Pure stop-and-copy**(또는 **cold**)기법과 **live**(또는 **hot**)으로 나눌 수 있다.

pure-stop-and-copy기법은 가상 머신을 종료하고 이주시킨후 다시 시작하여 서비스를 재개하는 방식이며 대체적으로 머신 종료와 시작에 긴 콜 타임이 발생한다. 반면에 **live**기법은 가상머신을 종료하지 않고 그대로 이주시키는 방

식으로 가상머신이 사용하는 메모리를 추가적으로 복사하여야 하는 부담이 발생하지만 서비스를 계속 제공할 수 있으므로 클 타임이 짧다[4].

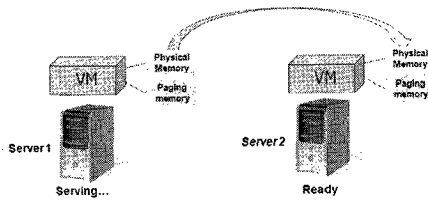
live방식은 그림 5와 같은 절차를 따라 가상머신을 이주시킨다. 첫째, 이주 시킬 가상머신의 이미지를 이주 대상의 물리적 서버로 전송한다. 대체적으로 클라우드는 일반적인 목적의 서비스를 수행하기 위한 자원을 제공하므로 가상머신의 이미지는 동일한 경우가 많다. 예를 들어 웹서비스의 경우 서비스를 위한 페이지 소스만 다를 뿐 웹서버가 설치된 가상머신의 OS이미지 자체는 동일하다. 따라서 이 절차에서 다른 서버로의 이미지 전송이 필요 없을 수 있다. 둘째, 이주 대상 서버에서 가상머신이 준비된 경우, 이주 준비 중인 가상머신의 작업을 그대로 수행하기 위해서 물리적 메모리와 페이징 메모리를 이주 대상 서버에 준비된 가상머신으로 복사하여야 한다. 이 과정에서 이주 준비 중인 가상머신은 서비스를 계속 제공한다. 셋째, 메모리의 복사가 진행될 동안 이주 준비 중인 가상머신은 서비스를 제공하고 있는 상태기 때문에 메모리의 수정이 이루어 질 수 있다. 따라서 수정이 이루어진 메모리는 다시 준비된 가상머신에게 전송하여 메모리를 갱신 시킨다. 이러한 과정을 반복하여 서로의 메모리를 동기화 시킨다. 앞서 말한 세 가지 절차가 완료되면 이주 준비 중인 가상머신은 종료됨과 동시에 이주 대상 서버에 준비된 가상머신이 서비스를 이어 제공한다.



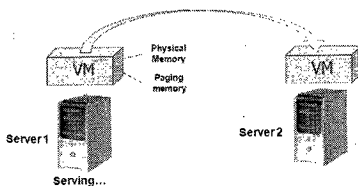
(c) 서비스 제공 주체 이전

그림 5. 가상 머신의 live이주 기법

가상 머신이 이주할 때 클 타임을 살펴보기 위하여 [5]에서는 다음과 같은 실험을 수행하였다. 테스트 클러스터의 구성은 6개의 서버로 구성되어 있으며 이중 1개의 서버는 헤드노드 역할을 수행하고 5개는 가상머신 호스트로 사용된다. 각 노드는 Intel Xeon E5410(2.33 GHz Quad-core processor, 2x6MB L2 cache, Intel VT technology), 4 GB 메모리, 7200 rpm 하드 디스크로 구성되어 있다. 이들 노드들은 기가비트 랜으로 연결되어 있으며 호스트 노드는 Citrix XenServer Enterprise Edition 5.0.0.을 사용하였고 가상 이미지는 Ubuntu 8.04 server edition OS와 apache 2.28, mysql 5.2.4-2가 설치되어 있다. 실제와 비슷한 워크로드를 생성하기 위하여 Web 2.0 어플리케이션인 Olio와 Faban load generator를 사용하였다. 클러스터에서 가상 머신 이주는 그림 6과 같이 이루어진다.



(a) 이미지 복사



(b) 메모리 복사

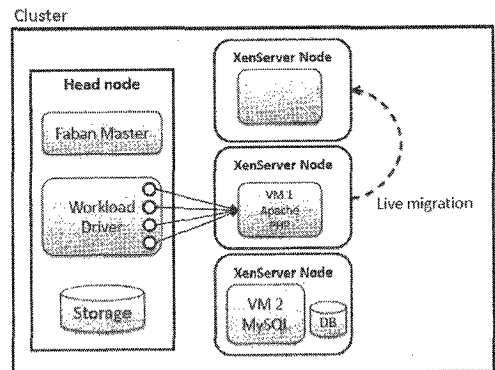


그림 6. 가상머신 이주

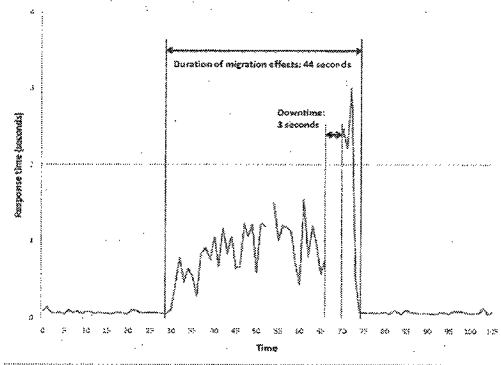


그림 7. 가상 머신 이주 시 콜 타임

그림 7에서 30초부터 74초까지 가상 머신이 이주가 이루어졌으며 총 44초가 소요되었다. 이 구간동안 서비스는 실제 발생한 콜 타임 3초를 제외하고는 모두 정상적으로 제공되었다. 가상머신이 이주를 시작하는 30초부터는 서비스 제공과 메모리복사가 동시에 이루어지기 때문에 응답시간이 대체적으로 길어졌고, 서비스 제공 주체가 이전되는 67초에서 70초까지는 서비스가 중단되는 콜 타임이 발생했다. 콜 타임 발생 이후 서비스를 처리하기 시작한 70초부터 응답시간의 지연이 발생하는데 이는 콜 타임동안 처리하지 못한 응답을 한꺼번에 처리 하기위한 오버헤드가 발생하였기 때문이다.

[4]에서는 Quake 게임 서버역할을 수행하는 가상머신을 이주시키는데 60밀리초에서 최대 3초까지 소요되었다고 한다. 지금까지 결과로 비추어볼 때 가상머신 이주는 비록 콜 타임이 발생하지만 여러 가지 장점을 가지므로 클라우드 서비스 제공자에게 매력적인 기법임에는 틀림없다.

2. MapReduce

MapReduce는 구글에서 제안한 데이터의 병렬처리 프로그래밍 모델이다. MapReduce를 사용하면 많은 노드에게 처리할 데이터를 분할하여 처리하게 함으로써 분산 환경 시스템에서 효율적인 작업 수행이 가능해진다. 오픈 소스 프로젝트인 Hadoop[6]은 구현된 MapReduce를 제공하고 있으며 주로 Amazon, Baidu, Facebook, Yahoo 등 다양한 클라우드 기반 시스템에서 사용하고 있다.

MapReduce는 슈퍼컴퓨팅 분야에서 연구된 MPI와 비교

될 수 있다. MPI는 고속 네트워크 인프라상에서 분산된 노드 사이에서의 계산을 위한 데이터 분배 및 취합을 수행하는 모델이다.

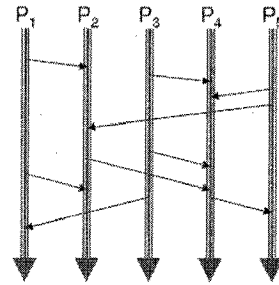


그림 8. MPI의 작업 처리

MPI의 프로그래밍 모델은 위 그림과 같이 각 노드에게 분배되는 데이터가 주로 메모리에 저장 되어 전달되며, 노드 사이에 데이터를 교환함으로써 주어진 문제를 해결하는 과정이 빈번하게 발생된다. 각 프로세스(P)가 수행하는 작업은 해당 컴퓨팅 자원을 모두 소비하는 연산 작업이 대부분이며, 특정 프로세스가 해당 컴퓨팅 자원을 모두 소비하는 특징이 있다. 또한 노드사이에 데이터 이동이 빈번하게 발생하므로 고속 네트워크 인프라가 반드시 구축되어야 한다. 그에 반하여 MapReduce는 연산능력보다 대규모의 데이터 처리에 주안점을 둔 모델로 대용량 데이터를 분산하고 이를 각 노드가 수행하여 중간결과를 다시 디스크에 저장하는 방식이다. 노드 상의 데이터 통신은 대부분 파일 I/O위주로 이루어진다. 그림 9은 MapReduce의 동작 방식을 보여준다.

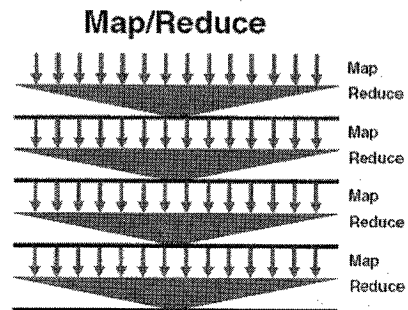


그림 9. MapReduce

결국, MPI 와 MapReduce 프로그래밍 모델은 주어진 문제를 해결하는 방식에 대한 상반된 입장을 보인다. 전자는 고속 네트워크를 이용한 대규모 컴퓨팅 계산에 적합한 모델로써 기초과학 연구에 주로 사용되는 반면, 후자는 데이터의 분석, 가공과 같은 대규모 데이터 처리에 적합하다. 이러한 이유로 최근 클라우드 시스템에서 MapReduce를 사용하여 데이터의 분석, 가공을 위한 여러 가지 기법들이 연구 되고 있다. 그중 [7]에서는 데이터 분석을 위한 효율적인 프로비저닝(Provisioning) 기법을 제안하고 있다. 프로비저닝은 주어진 작업을 처리하기위하여 작업할당 및 처리 순서 등의 계획을 준비하는 것으로 MapReduce에서는 작업을 할당할 노드의 개수, 데이터 분할 개수 등의 결정을 의미한다.

클라우드 컴퓨팅은 유틸리티 컴퓨팅을 요금모델로 사용하기 때문에 작은 비용으로 최적의 성능을 내야한다. 그러나 Hadoop에서 기본적으로 제공되는 MapReduce를 위한 프로비저닝은 최소 비용, 최적 성능을 보장하지 않는다. 따라서 이 연구에서는 수행할 작업에 따른 최소 비용, 최적 성능을 위한 프로비저닝에 대해 연구하였다. 그리고 [8]에서는 Mochi라는 로그분석 틀을 제안하였다. 이 연구는 로그를 통하여 작업에 대한 설정, 결과를 분석하고 이를 통하여 효율적인 프로비저닝이 가능하도록 가이드 라인을 제공한다. 이외에도 MapReduce는 클라우드 컴퓨팅을 이용한 데이터분석에 효율적이기 때문에 이와 관련된 연구가 활발히 진행되고 있다.

3. 기타 연구

자원 할당과 MapReduce이외에도 클라우드의 성능을 향상시키고 유틸리티 컴퓨팅을 고려한 여러 가지 연구가 진행되고 있다. 본 절에서는 관련 연구를 간략하게 소개한다.

[9]에서는 데이터 공유에 대한 연구를 진행하였다. 클라우드에서 독립적인 웹 서비스가 수행된다면 웹 서비스들은 공동의 데이터 베이스를 통하여 데이터 공유를 빠르고 간단하게 수행할 수 있다. 이 연구에서는 공동의 데이터 베이스를 사용함으로써 발생할 수 있는 여러 가지 문제점들을 설명한다.

[10]에서는 클라우드를 위한 프록시 네트워크를 연구하였다. 클라우드에서는 유동적인 서비스를 위하여 서비스 클라우드와 데이터 클라우드를 따로 운영할 수 있다. 이 경우 서비스 클라우드와 데이터 클라우드 간에는 유사한 데이터 교환이 많아지는데, 이때 발생할 수 있는 네트워크 트래픽을

줄이기 위해 두 클라우드 사이에 사용될 수 있는 프록시 네트워크를 제안하였다.

[11]에서는 자원의 임대비용과 어플리케이션 성능을 고려하여 어플리케이션 수행환경(work bench)을 자동으로 구성해주는 프레임워크를 제안하였다. 이 연구에서는 어플리케이션 수행에 필요한 자원의 capacity와 처리 결과(처리 속도, 정확도)를 실험을 통하여 제시하였으며 이러한 결과를 바탕으로 예상 비용과 필요한 속도, 정확도를 만족 시킬 수 있는 어플리케이션을 위한 자원 분배 기준을 제시하였다.

[12]에서는 클라우드 컴퓨팅을 위한 기반기술을 구현하였다. 연구에서 소개된 NC state cloud는 VCL(Virtual computing laboratory)을 기반으로 구현된 오픈 소스 기반의 클라우드 솔루션이며 IBM의 blade center와 Hadoop기반의 기술을 사용하여 아마존 EC2와 많은 부분에서 비슷한 기능을 제공한다. NC state cloud는 현재 3만명 이상의 North Carolina대학의 학생과 교직원을 대상으로 EC2와 같은 인프라와 구글앱스와 유사한 소프트웨어를 제공해주고 있다. NC cloud의 향후 연구 주제로는 이기종의 하드웨어 간의 표준화, 보안, 시간대에 따른 자원 활용률에 따른 유연한 대처방안을 제시하고 있다.

이밖에도 클라우드를 위한 보안 매커니즘, 병렬 알고리즘 구현, 자원 공유기법, 성능 최적화기법 등 많은 연구가 진행되고 있다.

III. 결론

가트너는 최근 그룹보고서에서 클라우드 컴퓨팅에 대한 기업들의 전환 시기를 세 단계로 나누고 있다. 1단계는 2010년부터 2013년에 걸쳐 일관성 있는 개발도구와 다양한 기능 등이 클라우드 기술로 통합하는 단계, 2단계는 2012년부터 2015년까지 클라우드 컴퓨팅의 주류시장이 형성되면서 일상화된 솔루션이 사용되는 단계, 3단계는 2013년 이후 소수의 대형 클라우드 기반 플랫폼 공급업자들이 시장을 지배하면서 기술적인 표준을 제공하는 단계에 이를 것으로 보고 있다. 이러한 보고처럼 클라우드 컴퓨팅은 멀지 않은 시간 안에 활발하게 운영될 것으로 보이고, 그에 따라 많은 서비스들이 클라우드 기반으로 옮겨갈 것으로 예상된다.

클라우드 컴퓨팅이 활성화 되고 정상적인 서비스를 제공

하기 위해서는 안정성, 개인 정보 보호, QoS제공, 경쟁력있는 가격 등 여러 가지 관점에서 타당성을 제시하여야 한다. 그러나 지금의 클라우드 컴퓨팅 기술은 미성숙 단계로 앞서 언급한 타당성을 제공하기에는 부족하다. 일례로, 2008년 2월 클라우드 컴퓨팅 서비스인 아마존 S3서비스가 3시간 이상 중단되는 사고가 발생했다. 아마존은 상황을 신속히 수습하고 해명 자료를 발표했으나 이 사고로 인해 클라우드 컴퓨팅 서비스에 대한 불신을 증폭시킨 것은 사실이다. 아마존의 사태에서도 알 수 있듯이 클라우드 컴퓨팅에서 서비스를 제공하기 위해서는 서비스 안정성에 대해 확실하게 보장해 주어야 하며, 이밖에 서비스의 질을 저해할 수 있는 요소들을 감지하고 통제할 수 있는 높은 수준의 관리 능력이 필요하다. 현재 클라우드 컴퓨팅은 시작단계라고 할 수 있으며 많은 산학기관에서 서비스의 질을 향상시키기 위한 다양한 연구가 진행되고 있다. 그러나 아직 실질적인 서비스를 제공하기 위해선 여러 가지 측면에서 해결해야 할 과제들이 많다.

참고문헌

- [1] M. Armbrust, et al., "Above the Clouds: A Berkeley View of Cloud Computing," UCB/EECS-2009-28.
- [2] J. Londono, A. Bestavros and S. Teng, "Collocation games and their application to distributed resource management," Tech. Rep. BUCS-TR-2009-002, Boston University, 2009.
- [3] J. Sonnek and A. Chandra, "Virtual Putty: Reshaping the Physical Footprint of Virtual Machines," in Proceedings of the Workshop on Hot Topics in Cloud Computing (HotCloud'09), San Diego, CA, June 2009.
- [4] C. Clark, K. Fraser, S. Hand, J.G. Hanse, E. Jul, C. Limpach, I. Pratt, A. Wared, "Live migration of virtual machines," in Proc. of NSDI'05: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, Berkeley, CA, USA, USENIX Association (2005), pp. 273-286.
- [5] W. Voorsluys, J. Broberg, S. Venugopal and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in Proc. of the 1st International Conference on Cloud Computing, pp. 254-265, 2009.
- [6] Hadoop, <http://hadoop.apache.org>
- [7] K. Kambatla, A. Pathak, and H. Pucha, "Towards Optimizing Hadoop Provisioning in the Cloud," in Proc. of the First Workshop on Hot Topics in Cloud Computing, June 2009.
- [8] J. Tan, X. Pan, S. Kavulya, R. Gandhi, P. Narasimhan, "Mochi: Visual Log-Analysis Based Tools for Debugging Hadoop," in Proc. of HotCloud '09 Workshop on Hot Topics in Cloud Computing, 2009.
- [9] R. Geambasu, S. D. Gribble, and H. M. Levy, "Cloud Views: communal data sharing in public clouds," in Proc. of HotCloud '09 Workshop on Hot Topics in Cloud Computing, 2009.
- [10] J. Weissman and S. ramakrishnan, "Using Proxies to accelerate Cloud Applications," in Proc. of HotCloud '09 Workshop on Hot Topics in Cloud Computing, 2009.
- [11] A. Demberel, J. Chase, S. babu, "Reflective Control for an Elastic Cloud Application: An automated Experiment Workbench," in Proc. of HotCloud '09 Workshop on Hot Topics in Cloud Computing, 2009.
- [12] M. A. Vouk, "Cloud computing - Issues, research and implementations" in Proc. of the International Conference on Information Technology Interfaces (ITI'08), pp. 31-40, June 2008.

저자 소개



김창현

2008: 경일대학교
컴퓨터공학과 공학사.
2010: 한양대학교
컴퓨터공학과 공학석사.
현재: 한양대학교
컴퓨터공학과 박사과정.
관심분야: 병렬처리시스템,
센서 네트워크,
그리드컴퓨팅,
클라우드컴퓨팅.



전창호

1977: 한양대학교
전자공학과 학사.
1982: Cornell University
컴퓨터공학과 석사.
1986: Cornell University
컴퓨터공학과 석사.
1977-1979: 전자통신연구소
연구원.
현재: 한양대학교
전자컴퓨터공학부 교수.
관심분야: 병렬처리시스템,
성능분석, Grid컴퓨팅,
클라우드컴퓨팅.



이원주

1989: 한양대학교
전자계산학과 공학사.
1991: 한양대학교
컴퓨터공학과 공학석사.
2004: 한양대학교
컴퓨터공학과 공학석사.
현재: 인하공업전문대학
컴퓨터정보과 부교수.
관심분야: 병렬처리시스템,
성능분석, Grid컴퓨팅,
클라우드컴퓨팅.