클라우드 컴퓨팅을 위한 특정한 상황에서의 병목현상과 데이터 분산 처리

오성준, 황명진, 윤희용
성균관대학교 컴퓨터 공학과
e-mail: okwehs@hotmail.com

Hot Spot and Data Distribution Process in Specific Situation for Cloud Computing

Sung-Jun Oh, Myung-Jin Hwang, Hee-Young Youn
Dept of Computer Engineering, Sung Kyun Kwan University

요 약

클라우드 컴퓨팅은 대규모의 분산 시스템을 기반으로 페타바이트급 이상의 처리가 가능해진 최근에 각광받고 있는 기술이다. 대부분 비 대칭 분산 구조로 객체기반 클러스터 파일 시스템으로 이루어져있다. 이런 환경에서 효율성과 신뢰성 등을 고려하여 많은 연구가 진행 중이다. 본 논문에서는 특정한 상황에서 병목현상과 메타 데이터 서버간의 부하분산을 처리함으로써, 한 특징인 효율적인 측면에서 향상이 가능한 기법을 제안하였다.

1. 서 론

최근 클라우드 컴퓨팅은 서로 다른 물리적인 위치에 존재하는 하드웨어와 소프트웨어들을 가상화 기술으로 통합하여 사용자들에게 네트워크를 통해 서비스를 제공한다.

클라우드 컴퓨팅 시스템들의 구조는 크게 대칭형 구조(symmetry)와 비 대칭형(asymmetry) 구조로 구분할 수 있다. 비대칭형 구조에서는 메타데이터를 요구하는 다수의 클라이언트들이 한 메타 데이터 서버(MetaData Server, 이하 MDS)에 집중되어 그 MDS에서 정상적으로 처리 할 수 있는 한계를 초과하는 경우, 이런 병목현상(hot spot)을[6] 해결 할 수 있는 구조를 설계할 필요가 있다. MDS의 병목 현상을 해결하기 위한 시도로 MDS들의 클러스터를 고려할 수 있다. MDS 클러스터를 구성하여 효율적으로 운영하기 위해서는 하나의 MDS에게 과부하가 걸리지 않도록 클러스터를 구성하고 있는 MDS들 간에 부하를 균등하게 분산시키는 방법을 찾는 것이 중요하다.[2]

본 논문의 목적은 각 클러스터 기법들에서 병목현상이 일어나고, 새로운 MDS를 추가했을 때에 기존 클러스터 기법에서 처리할 때 생기는 절차적 문제를 해결하기 위해 병목 현상과 메타 데이터 분산, 두 가지 문제를 해결하는데 있어 불필요한 단계를 제거하는 것이다. 이로써 MDS간에 데이터를 이동하기 위한 Network bandwidth이나 I/O처리 부분에 있어서 개선을 보여준다.

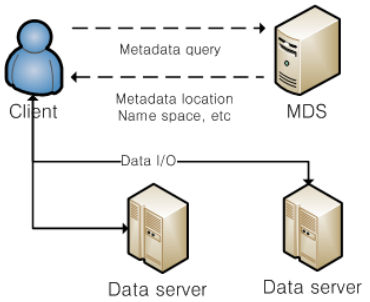
본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 기술하고, 3장에서는 특정한 상황에서 병목현상과 메타 데이터 분산을 처리하는 새로운 방법에 대해 소개한다. 4장에서는 제안된 기법의 결론과 향후 연구를 제시한다.

2. 관련 연구

2. 1. 비 대칭 분산 파일 시스템 구조

비대칭형 분산 파일 시스템 구조란, MDS가 별도로 독립되어 메타 데이터와 데이터의 처리 경로가 독립됨으로써 클라이언트, 데이터 서버 등을 대규모로 운영하는 경우에 입출력 성능 및 확장성이 증가하는 장점이 있는, 즉 MDS가 별도로 독립되어 있는 구조를 말한다.[2]

이 구조에서는 클라이언트가 MDS에 질의를 하며, 실제 데이터 입출력 작업은 클라이언트와 데이터서버 사이에서 직접 처리 되게 된다.(그림 1)



(그림 1) 비 대칭 분산 파일 시스템

2. 2 클러스터 파일시스템

클러스터 파일 시스템은 네트워크 상의 분산된 대량의 서버들을 클러스터로 구성함으로써 대용량의 저장 공간을 확보하고 병렬 처리를 통한 빠른 입출력 속도를 제공한다. 이를 통해 파일 입출력의 성능을 높여주고 독립적인 확장과 안전한 파일 서비스를 제공한다. 비 대칭형 구조를 갖는 대

표적인 파일시스템인 객체기반 클러스터 파일 시스템은 기본의 파일 시스템과는 달리 객체처리를 위한 객체기반 인터페이스로 논리적인 객체 단위로써 관리한다.

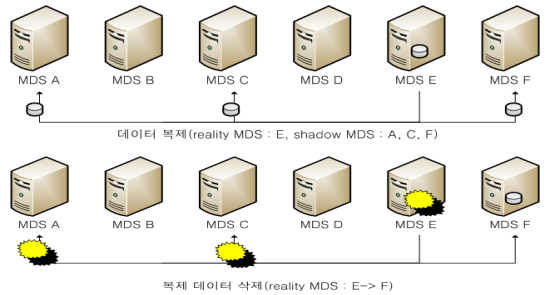
현재 나온 MDS 클러스트링 기법들은 정적 서브트리 분할(static subtree partitioing), 해싱(hashing), Lazy Hybrid, 동적 서브트리 분할(dytnamic subtree partitioning)[1] 등의 기법들이 나옴으로써 MDS들의 클러스터를 운영하는데 있어서 성능의 저하를 방지하고 효율적인 부하 분산(load balancing)을 유지하기 위해 많은 연구들이 활발히 진행되고 있다.

3. 제안 기법

한 클러스터 내의 여러 MDS 중 한 MDS에 병목현상이 나타나고 그 클러스터에 새로운 MDS를 연결되었을 경우, 채택된 클러스터 기법에 따라 파일이 적절한 MDS에 분산되어야 한다. 이 때 MDS간 I/O의 부담이 크다. 기존 시스템은 앞으로 제안할 시스템에 비해서 절차적인 부분으로 인해 I/O 처리손실이 생길 수 있다.

예를 들면, 한 클러스터 내의 A, B, C, D, E의 5개의 MDS가 있다. 그 중 E MDS가 병목현상이 일어났다 하자. 그리고 F라는 MDS가 그 클러스터에 추가가 되었다. 그럴 때 E MDS는 병목현상을 해결하기 위해서 랜덤으로[5] A, B, C의 MDS에 병목현상을 일으킨 파일들을 복사하여 옮긴다.[3](보통 이 다른 MDS로의 복제 상수는 3이다.) 이 때, E MDS는 reality metadata server라 하고, A, B, C MDS는 shadow metadata server라 한다.[4] 그 후 병목 현상이 풀리고 나면, A, B, C의 MDS에 있는 복제된 파일을 지우게 된다. 그러고 나서 새롭게 추가된 F MDS에 할당 되어야 할 파일을 각 MDS에서 찾아내서 F MDS로 옮겨지게 된다. 설사 처음에 F MSD를 포함해서 복제된다고 해도 병목현상이 풀린 후에 복제된 메타 데이터는 지워지게 되고, 다시 F MDS에 파일을 할당하는 비효율적인 절차가 이루어지게 될 것이다.

이것의 문제는 현재 병목현상이 일어난 파일이 새롭게 추가된 MDS에 할당될 파일인지 아닌지 고려를 해보지도 않고 다른 MDS에 할당한다는 것이다.



(그림 2) 새로 제안된 기법의 예

새로 제안된 기법(그림 2)은 가장 먼저 병목현상을 해결하기 위해서 병목 현상이 일어난 파일을 새로 추가된 MDS에 무조건 복제할당을 하는 것이다. 그 이유는 현재 새롭게 추가된 MDS는 현재 다른 MDS로부터 받은 파일이 아무 것도 없는 상태이기 때문에 클라이언트로부터 아무런 쿼리를 받지 않는다. 그렇기 때문에 이 새로 추가된 MDS는 다른 MDS로의 파일 할당을 하는 것보다 2차적인 병목현상이 일어날 확률이 가장 적고, network bandwidth도 다른 MDS보다 높기 때문에 hot spot처리에 가장 용이한 노드이기 때문이다.

그 후, 병목현상이 일어난 파일이 원래 이 새롭게 추가된 MDS에 추가되어야 하는 파일이라면 병목현상이 풀어졌을 때, 이 MDS에 있는 복제된 파일을 제외하고 다른 MDS에 있는 파일들을 지우면 되는 것이다. 그러므로써 병목현상은 병목현상대로 처리되고, 병목현상 후에 새롭게 추가된 MDS에 파일을 다시 할당해야 하는 단계를 뛰어넘게 되는 것이다.

4. 결론 및 향후 연구

기존 기법에서 필연적으로 한번 반복해야했던 불필요한 단계를 제안 된 기법으로써 제거함으로써 MDS 사이의 파일을 옮기는 I/O처리 시간을 단축시킬 수 있으며, 추가된 MDS를 이용하기 때문에 병목현상 처리 시에 다른 MDS에 복제된 파일을 옮겨 놓은 것 보다 network bandwidth 측면에서 효율적이게 된다.

추후에 더 나은 성능을 위해서 병목현상이 일어날 수 있는 MDS를 미리 예측하거나 병목현상과 MDS간의 부하분산 처리를 따로 떼지 않고 한번에 처리 할 수 있는 기법을 제안할 것이다.

참고문헌

[1] S.A. Brandt, L. Xue, E.L. Miller, and D.D.E. Long, “Efficient Metadata Management in Large Distributed File Systems,” In Proc. of IEEE/11th NASAGoddard Conf. on Mass Storage Systems and Technologies, Apr. 2003, p.290-298.

[2] 차명훈, 이상민, 김준, 김영균, 김명준, “대규모 분산 파일 시스템 환경의 메타 데이터 관리”, 전자통신동향분석 제 22권 제 3호 2007년 6월, p154-165

[3] S.A. Weil, K.T. Pollack, S.A. Brandt, and E.L.Miller, “Dynamic Metadata Management for Petabytescale File Systems,” In Proc. of the 2004 ACM/IEEE Conf. on Supercomputing, Nov. 2004.

[4] Wiojia Li, Wei Xue, Jiwu Shu, Weimin Zheg "Dynamic Hashing : Adaptive Metadata Management for Petabyte-scale File", Proceedings of the 23 st IEEE/14 th NASA Goddard, , Apr. 2006

[5] Gongye Zhou, Qiuju Lan, Jincai Chen, "A Dynamic Metadata Equipotent Subtree Partition Policy for Mass Storage System," fcst, pp.29-34, 2007 Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST 2007), 2007

[6] SA Weil, SA Brandt, EL Miller, KT Pollack, "Intelligent Metadata Management for a Petabyte-cale File System", 2nd Intelligent Storage Workshop