

클라우드 스토리지 성능 분석

이재원 채동주 안재형 김장우
포항공과대학교 고성능컴퓨팅연구실

연구 목적

클라우드 컴퓨팅을 위한 스토리지 시스템의 안정성, 기능성, 편리성 (Reliability, Availability, and Serviceability, RAS) 및 확장성 (Scalability) 을 조사하고, 컴퓨터 아키텍처 기술로 시스템 개선책을 제시한다

요약

- 클라우드 스토리지 시스템 중 하나인 OpenStack Swift [1] 를 이용해 다양한 사용자 요청에 대한 스토리지 시스템의 성능을 분석함
- 현재 OpenStack Swift의 구조에서 병목현상이 발생하는 지점들을 밝혀냄
- 서버의 가상화를 통해 실시간으로 자원 분배를 조정하여 성능을 향상시키는 기법을 제시함

연구 배경

- OpenStack Swift의 구조

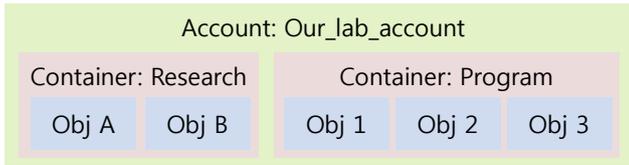


그림 1. Swift의 저장소 개념

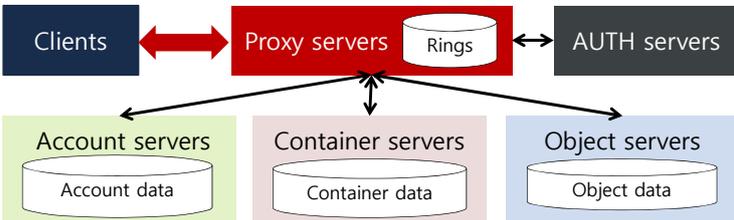


그림 2. Swift의 논리적 서버 구조

Ring			
Partition #	Zone 1	Zone 2	Zone 3
0	Dev0 (N0, D1)	Dev5 (N1, D3)	Dev10
1	Dev1	Dev6	Dev11
...

그림 3. Swift Ring의 구조

OpenStack Swift의 특징

- 데이터와 메타데이터를 물리적으로 구분된 장치에 일정 개수만큼 복사, 저장함으로써 단일 고장점 (Single Point of Failure) 을 제거하고 안정성과 기능성 (Reliability and Availability) 을 보장함
- 저장소의 일시적, 영구적 변화에 대해 항상 일정한 개수의 복사본을 유지함으로써 편리성(Serviceability)을 강화함
- Ring이라는 특수한 구조를 이용하여 데이터/메타데이터의 위치를 판별 (Amazon의 Dynamo [2] 와 유사)

문제점

- 안정성, 기능성, 편리성(RAS)을 얼마나 보장할 수 있는가
- 성능 저하 없이 대규모로 확장이 가능한가
- 병목현상이 발생하는 지점은 어디인가
- 컴퓨터 아키텍처 기술을 이용하여 어떻게 시스템을 개선할 수 있는가

연구 결과 및 분석

- 실험 설정

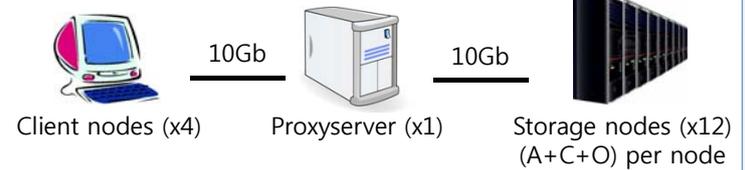


그림 5. OpenStack Swift 실험 환경

- 실험 결과

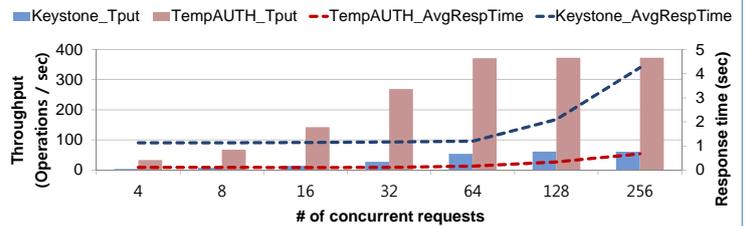


그림 6. 1MB Upload 성능 측정

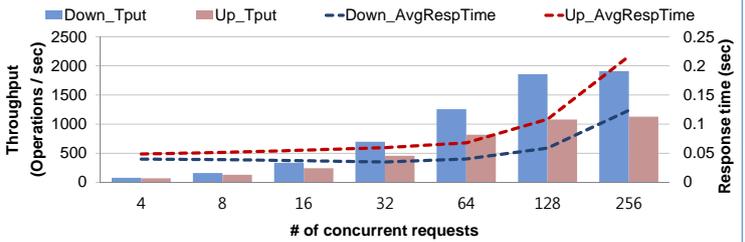


그림 7. 128KB Upload / Download 성능 측정

결과 분석

- 파일 크기가 작을수록 시간당 더 많은 요청이 처리됨
- 업로드의 경우 여러 개 사본이 동시에 스토리지로 전송되므로 다운로드보다 네트워크 사용량이 큼
- 본 실험에서는 스토리지에서의 병목현상이 발견되지 않음
- Openstack Keystone 인증 서버 사용시 느린 인증 속도와 네트워크 홉 증가로 인증 단계에서 병목현상이 발생함
- 큰 파일 (1MB) 전송시 프록시와 스토리지 간 네트워크에서 대역폭에 의한 병목현상이 발생함 (전송률 약 1GB/s)
- 작은 파일 (128KB) 전송시 프록시 CPU 사용량이 급증하며 프록시와 스토리지 간 네트워크에서 연결 개수로 인한 병목현상이 발생함 (전송률 약 700MB/s)

결론

- Keystone 수정, 확장성 있는 데이터베이스 사용, 네트워크 최적화를 통해 인증 시스템의 병목현상을 해결해야 함
- 큰 파일 전송의 경우 네트워크 대역폭을 효율적으로 활용하므로 네트워크 용량 증설을 통해 확장성을 확보해야 함
- 프록시에서 CPU 혹은 연결 개수로 인해 병목현상 발생 시 가상화된 프록시를 실시간으로 스토리지 노드 중 일부에 배치함으로써 성능 향상을 도모할 수 있음

참조 문헌

- [1] Openstack Swift, <http://swift.openstack.org>
[2] G. DeCandia et al., "Dynamo: amazon's highly available key-value store," in Proceedings of the twenty-first ACM symposium on Operating systems principles (SOSP), 2007.