

분산 파일 시스템을 위한 웹기반 워크로드 생성기 개발¹⁾

김동오⁰, 차명훈, 김홍연, 김영균
 한국전자통신연구원 스토리지시스템연구실
 {dokim, mhcha, kimhy, kimyoung}@etri.re.kr

김성엽
 테스트마이다스
 sykim@testmidas.com

The Development of Web-based Workload Generator for a Distributed File System

Dong-Oh Kim⁰, Myung-Hoon Cha, Hong-Yeon Kim, Young-Kyun Kim
 Storage System Research Laboratory, ETRI
 Sung-Yub Kim
 Convergence Testing Business Department, TestMidas

요 약

다양한 응용 분야에서 대량의 데이터 저장이 빈번해 짐에 따라 분산 파일 시스템에 대한 효율성이 커지고 있으며, 이로 인해 수많은 분산 파일 시스템이 개발되고 있다. 이러한, 분산 파일 시스템의 효율성을 검증할 수 있는 방법에 대한 필요성이 커지고 있다. 또한, SSD와 같은 우수한 성능을 가지는 장치를 사용하는 분산 파일 시스템에서는 SSD와 같은 장치의 효율성을 보기 위한 복잡한 워크로드를 생성 하여 시험할 수 있어야 한다. 하지만 기존의 입출력 시험 도구에서는 복잡한 상황을 시험하는 것이 매우 어렵다. 따라서, 본 논문에서는 필요에 따라 파일의 분포에 따라 다른 입출력 패턴을 가지는 여러 부하 상황을 생성이 가능하며, 웹을 통해 손쉽게 시험할 수 있는 분산 파일 시스템을 위한 웹기반 워크로드 생성기를 개발하였다.

1. 서론

정보화 사회가 발전함에 따라 발생하는 대량의 데이터를 저장하기 위해 스토리지 시스템이 대형화 되고 있다. 이러한 환경에서 거대한 데이터를 효율적으로 저장하기 위해 여러 분산 파일 시스템이 개발되었으며[1], 안정화 및 성능 개선이 급속히 이루어지고 있다. 또한, SSD와 같은 다양한 저장 장치가 발전함에 따라 이러한 장치를 포함하여 구성하는 파일 시스템에 대한 연구가 이루어지고 있다[2].

이러한 대용량 분산 파일 시스템을 효율적으로 시험하기 위해서는 다양한 패턴의 대용량 데이터를 효율적으로 발생 시키고 결과를 확인할 수 있어야 한다. 또한, SSD와 같은 장치가 특수한 목적으로 결합되어 있는 경우[3]의 효율성을 보기 위해서는 복잡한 형태의 데이터를 좀 더 손쉽게 생성할 수 있어야 한다.

따라서, 본 논문에서는 비중이 다른 여러 부하에 따른 다양한 입출력 다양한 입출력 패턴 따라서 부하를 발생할 수 있으며, 웹을 통해 손쉽게 시험할 수 있는 분산 파일 시스템을 위한 웹기반 워크로드 생성기를 개발하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 본 논문에서 개발한 웹기반 워크로드 생성기를 연동하는데 사용한 분산 파일 시스템인 마하 파일 시스템(이하 MAHA-FS)에 대해 간략히 소개한다. 3 장에서는 본 논문에서 제시하는 분산 파일 시스템을 위한 웹기반 워크로드 생성기의 설계에 대해서 언급한다. 4 장에서는 워크로드 생성기의 구현 모습에 대해서 기술한다. 마지막으로 5 장에서는

결론에 대해 간략히 언급한다.

2. MAHA-FS

MAHA-FS는 다수의 저비용 서버들을 이용하는 대규모 파일 시스템으로써, ETRI에서 GLORY-FS[1]의 후속으로 개발 중인 고성능 분산 파일 시스템이다. MAHA-FS는 수많은 서버들의 저장 공간을 거대한 하나의 가상 공간으로 구성해 제공한다. 또한 데이터를 복제함으로써 장애에 대한 통제 능력과 높은 입출력 처리 성능을 제공한다. 그림 1은 MAHA-FS의 구조를 보여준다.



그림 1 MAHA-FS 구조

그림 1에서 보듯이 MAHA-FS는 모든 시스템 및 파일의 메타데이터를 저장하고 관리하기 위한 메타데이터 서버, 데이터를 저장 및 관리하기 위한 여러 데이터 서버, Fuse 라이브러리를 기반으로 여러 응용에게 POSIX API를 제공하기 위한 클라이언트로 구성된다.

3. 웹기반 워크로드 생성기 설계

현재 웹기반 워크로드 생성기는 분산 파일시스템인 MAHA-FS를 기반으로 구성되어 있다. 이는 편의를 위해 워크로드 생성기에서 볼륨 생성 및 관리를 지원하기 위한 것이며, 이를 제외한 다른 기능은 다양한 분산 파일

1) 본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였음. [10038768, 유전체 분석용 슈퍼컴퓨팅 시스템 개발]

시스템과 연동 가능하다. 그림 2는 웹기반 워크로드 생성기의 구조를 보여준다.

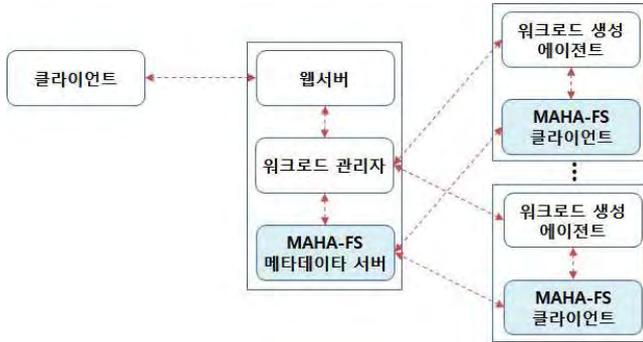


그림 2 웹기반 워크로드 생성기 구조

그림 2에서 보듯이 웹기반 워크로드 생성기는 크게 웹서버, 워크로드 관리자, 워크로드 생성 에이전트로 구성된다. 웹서버는 클라이언트에게 웹 인터페이스를 제공하며 워크로드 관리자에게 명령을 전달하는 역할을 담당한다. 워크로드 관리자는 볼륨, 워크로드, 맵 정보를 저장하고 관리하며 MAHA-FS에 볼륨을 생성하거나 워크로드 생성 에이전트에 워크로드 수행 명령을 내리고 결과를 취합하는 역할을 담당한다.

워크로드 생성 에이전트는 사용자가 설정한대로 워크로드를 발생시키고 결과를 취합하여 워크로드 관리자에게 전달하는 역할을 담당한다. 이때, 워크로드 생성 에이전트에서 워크로드를 발생하는 부분은 파일 시스템 성능 측정 도구로 널리 사용되는 iozone[4], bonnie++[5], fio[6], xdd[7] 중에서 순차(sequential)/랜덤(random) 접근과 READ/WRITE를 독립적으로 각각 수행할 수 있고, 파일 접근 주기를 설정할 수 있으며, 실시간 결과 출력이 가능한 xdd를 사용하였다.

또한, 본 워크로드 생성기는 하나의 워크로드 내에서 파일의 비중 별로 접근 형태 및 접근 주기를 구분이 가능하다. 그림 3은 접근 비중에 따른 접근 방법 및 주기를 다르게 설정한 예를 보여준다. 사용자는 그림 3과 같이 특정 데이터를 Hot 데이터 형태의 접근 패턴을 주는 것은 물론, Cold 데이터 형태의 데이터 접근 패턴 및 순차와 랜덤이 섞여서 들어오는 복합 패턴 등을 주는 것도 가능하다.

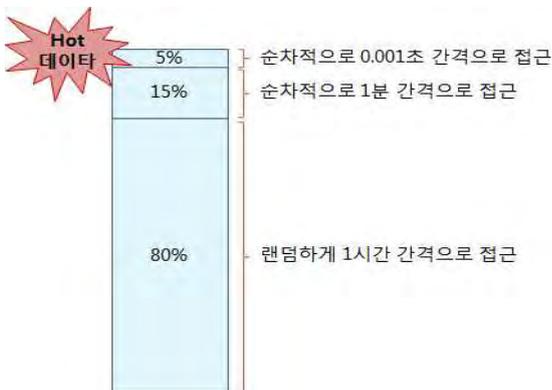


그림 3 접근 비중 설정의 예

그림 3에서 보듯이 전체 파일에서 5% 파일에 대해서는 모든 파일을 하나씩 0.001초 간격으로 접근하며, 다음 15% 파일은 모든 파일을 1분 간격으로 접근한다. 마지막으로 나머지 파일은 랜덤하게 선택하여 1시간 간격으로 접근하게 된다.

4. 웹기반 워크로드 생성기 구현 결과

웹기반 워크로드 생성기의 메뉴는 다음과 같이 4가지로 구성된다.

- Volume : 시험에 사용하기 위한 볼륨을 생성 및 관리하기 위한 메뉴
- Workload : 시험에 사용하기 위한 워크로드를 생성 및 관리하기 위한 메뉴
- Map : 클라이언트, 볼륨, 워크로드를 연결하여 시험을 수행하기 위한 맵을 생성 및 관리하기 위한 메뉴
- Execute : 시험의 수행 옵션을 지정하고 시작 명령을 내리기 위한 메뉴
- Result : 시험 수행 결과를 확인하기 위한 메뉴

그림 4는 웹기반 워크로드 생성기의 Volume 메뉴를 이용한 볼륨 관리 화면을 보여준다.

Volume Name	Chunk Size	Replica	Quota Limit	Edit	Delete
default	64.0M	2	10.0T	Edit	Delete
HDD	64.0M	4	10.0G	Edit	Delete
SSD	128.0M	2	1.0T	Edit	Delete
VOL	64.0M	2	10.0G	Edit	Delete

그림 4 볼륨 관리 화면

그림 4에서 보듯이 볼륨 정보는 볼륨 이름을 설정하기 위한 Volume Name, 청크 크기를 설정하기 위한 Chunk Size, 복제본 수를 설정하기 위한 Replica, 볼륨의 용량 한계를 설정하기 위한 Quota Limit 항목과 볼륨을 생성/편집/삭제하기 위한 Add New/Edit/Delete 항목으로 구성된다. 웹기반 워크로드 생성기의 볼륨 관리 메뉴는 MAHA-FS를 기준으로 구현되었으며, MAHA-FS의 유틸리티를 이용하여 볼륨 생성/삭제/편집/정보 확인을 수행한다.

그림 5는 웹기반 워크로드 생성기의 Workload 메뉴를 이용한 워크로드 관리 화면을 보여준다.

Name	File Size	File Num	Record	Pattern	Access	Freq.	File Select	Option	Edit	Delete
random_test	10M	10	4K	Seq:0%, Rnd:100%	R:100%, W:0%	1sec	random		Edit	Delete
seq_test	100M	10	64K	Seq:100%, Rnd:0%	R:0%, W:100%	1sec	seq		Edit	Delete

그림 5 워크로드 관리 화면

그림 5에서 보듯이 워크로드 정보는 워크로드의 이름 설정하기 위한 Name, 생성되는 파일의 크기를 설정하기 위한 File Size, 생성되는 파일의 개수를 설정하기 위한 File Num, 파일 접근 시 사용할 레코드의 크기를 설정하기 위한 Record, 파일 접근 시 순차/랜덤 접근의 비중을 설정하기 위한 Pattern, 파일 접근 시 READ/WRITE의 비중을 설정하기 위한 Access, 기본 파일 접근 시기를 결정하기 위한 Freq, 파일의 일정 부분에 대한 추가적인 접근 시기를 결정하기 위한 Option 항목과 워크로드를 생성/편집/삭제하기 위한 Add New/Edit/Delete 항목으로 구성된다.

그림 6은 웹기반 워크로드 생성기의 Map 메뉴를 이용한 맵 관리 화면을 보여준다.

Client IP	Workload Name	Volume Name	Edit	Delete
192.168.182.128	random_test	VOL	Edit	Delete
192.168.182.130	random_test	HDD	Edit	Delete
192.168.182.131	seq_test	SSD	Edit	Delete

그림 6 맵 관리 화면

그림 6에서 보듯이 맵 정보는 워크로드 생성 에이전트를 구동시킬 클라이언트의 IP를 지정하기 위한 Client IP, 해당 클라이언트에서 구동시킬 워크로드의 이름을 지정하기 위한 Workload Name, 워크로드를 수행하며 실제 파일을 생성할 볼륨 이름을 지정하기 위한 Volume Name 항목과 워크로드를 생성/편집/삭제하기 위한 Add New/Edit/Delete 항목으로 구성된다. Save As와 Load 항목은 현재 설정된 맵 항목을 저장하거나 기존에 설정해 놓은 맵 내용을 불러오고자 할 경우에 사용한다.

그림 7은 웹기반 워크로드 생성기의 Execute 메뉴를 이용한 실행 화면을 보여준다.

Map Execution

Cycling Option: Normal

of Unit: 1 # of Process: 1

Run Time: 00:00:10 File Creation:

Execute

그림 7 실행 화면

그림 7에서 보듯이 맵에서 설정한 워크로드의 실행 옵션을 지정하기 위한 Cycling Option, 맵에서 설정한 워크로드를 실행할 때의 그룹 단위를 지정하기 위한 # of Unit, 각 워크로드의 프로세스 테스크 개수를 설정하기 위한 # of Process, 워크로드를 실행할 시간을 설정하기 위한 Run Time, 파일 생성 여부를 선택하기 위한 File Creation, 실행하기 위한 Execute 버튼으로 구성된다.

워크로드 실행 옵션은 Normal, Sequential, Incremental 세 가지를 지원하며, Normal 옵션은 맵에서 설정한 워크로드 전체를 동시에 실행하라는 옵션이며, Sequential 옵션은 맵에서 설정한 워크로드 # of Unit에서 설정한 개수 단위로 순차적으로 실행하라는 옵션이

다. 마지막으로 Incremental 옵션은 맵에서 설정한 워크로드 # of Unit에서 설정한 개수 단위로 늘려가며 실행하라는 옵션이다.

그림 8은 웹기반 워크로드 생성기의 Result 메뉴를 이용한 결과 화면과 함께 전체 웹기반 워크로드 생성기의 모습을 보여준다. 그림 8에서 보듯이 실행한 결과를 취합하여 그래프 형태로 보여준다.

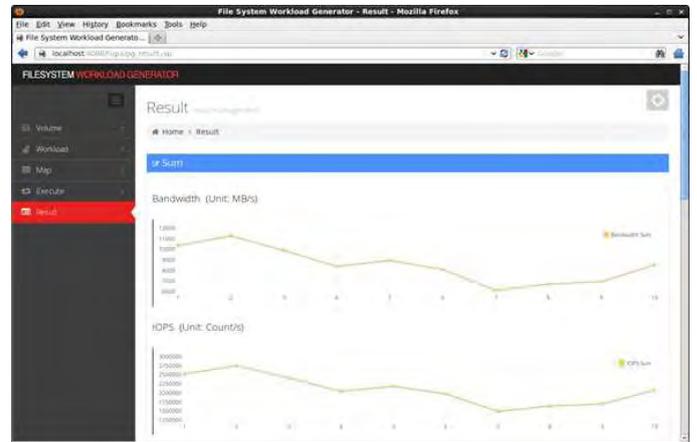


그림 8 결과 화면

5. 결론

본 논문에서는 비중이 다른 여러 부하에 따른 다양한 입출력 다양한 입출력 패턴 따라서 부하를 발생시킬 수 있으며, 웹을 통해 손쉽게 시험할 수 있는 분산 파일 시스템을 위한 웹기반 워크로드 생성기를 개발에 대한 설계 및 구현에 대해서 설명하였다. 향후에는 추가로 계속해서 신규 파일을 생성하며 부하를 발생시킬 수 있는 기능과 상시 발생하는 기본 부하 발생 기능을 추가에 대한 고려가 필요하다.

6. 참고문헌

- [1] 민영수, 진기성, 김홍연, 김영균, “클라우드 컴퓨팅을 위한 분산 파일 시스템 기술 동향,” 전자통신동향 분석 24권 4호, pp.55-68 2009.
- [2] EMC, Next Generation Storage Tiering with Isilon SmartPools, 2013.
- [3] Sheng Qiu and A. L. Narasimha Reddy, “NVMS: A Hybrid File System for Improving Random Write in NAND-flash SSD,” 2013.
- [4] iozone, <http://www.iozone.org/>
- [5] bonnie++, <http://sourceforge.net/projects/bonnie/>
- [6] fio, <http://freecode.com/projects/fio>
- [7] xdd, <http://www.ioperformance.com/>