

다양한 부하 특성에 따른 클러스터 시스템들의 성능 평가 및 비교

임유진, 최은미

한동대학교 정보통신학과

e-mail : inppo@hotmail.com, emchoi@handong.edu

Performance Evaluation and Comparison of Cluster Systems with Various Workload Characteristics

Yoojin Lim, Eummi Choi

Department of IT, Handong Global University

요 약

웹 클러스터 시스템은 인터넷에서 요청되는 부하를 여러 실제 서버들에게 분산시켜서 어플리케이션에 따른 처리와 응답을 하는 서버 단 시스템이다. 부하는 서비스를 제공하는 서비스에 따라 다른 성향을 띄게 되므로 서비스 제공자는 부하의 특성을 파악하여 알맞은 웹 클러스터 시스템을 사용하여야 효율적이고 좋은 성능을 보이게 된다. 이 논문에서는 7 개의 다른 성향의 부하 상황과 Alteon Web Switch, ALBM Traffic Manager 와 Linux Virtual Server 를 사용하여 클러스터를 구성하고 각 부하 상황에서 클러스터 시스템들이 어떤 성능을 보여주는지 비교 분석한다.

1. 서론

웹 서비스를 제공하는 서버들은 인터넷 사용량의 지속적인 증가로 인한 다양해지고 늘어나는 사용자와 부하(Workload) 상황에 놓이게 되었다. 서버가 제공하는 서비스의 종류나 사용자의 성격에 따라 요청하는 부하들의 성향이 달라진다. CPU, Network 나 Memory 와 같은 특정 자원에 종속되거나 여러 자원을 복합적으로 사용할 수 있고, 자원 사용량이나 사용 시간의 분포가 다르게 나타날 수도 있다. 이러한 부하의 과중을 서버 한대의 성능 확장으로 해결하는 것에는 한계가 있기 때문에, 동일한 서비스를 제공하는 여러 대의 실제 서버(Real Server)들을 하나의 시스템으로 묶는 클러스터(Cluster) 시스템을 사용하여 부하를 분산시켜 처리하게 되었다. 클러스터 시스템은 확장성, 고가용성, 경제성, 신뢰성을 높이는 고성능 시스템이라는 특성을 가진다. 또한 투명성을 가지고 있어 내부에서 실제 서버들이 추가되거나 제거되더라도 외부에서는 인식하지 못한 채 여전히 하나의 시스템으로 동작할 수 있게 된다. [1][2]

웹 클러스터 시스템에서는 외부에서 들어오는 요청

들을 시스템 앞에 존재하는 로드밸런서(Load Balancer)가 미리 정해져 있는 부하 분산 스케줄링 알고리즘에 따라 클러스터를 구성하는 서버들에게 전달시킨다. 일반적으로 사용되는 부하 분산 스케줄링 알고리즘에는, 들어오는 요청들을 서버들에게 순차적으로 전달하는 Round Robin(RR), 요청을 받을 때의 순간 접속수가 가장 적은 서버에게 전달하는 Least Connection(LC)를 들 수 있고, 서버 사이의 성능 불균형을 고려하여 요청을 전달하는 Weighted RR, Weighted LC 을 포함한 여러 알고리즘이 있다. 또한 서버의 상태를 고려하여 과부하 상태의 서버가 생기면 그 서버에게 과부하 상태가 벗어날 때까지 부하를 더 이상 주지 않는 Adaptive Algorithm 을 적용한 Adaptive RR(ARR)과 Adaptive LC(ALC)들이 있다.

이러한 웹 클러스터 시스템에 관련된 제품들은 이미 다양하게 개발, 사용되고 있다. 하드웨어 지향적이거나 소프트웨어 지향적인 구조적 특성, 가능한 부하 분산 스케줄링 알고리즘이나 모니터링과 같은 여러 면에 따라 성능에 큰 차이가 생기게 되고 또한 비용적인 면에서도 많은 차이가 나게 된다.

이 논문에서는 3 개의 특성이 다른 웹 클러스터 시

시스템을 가지고 각 제품의 특성에 따라 다양한 부하 상황에서 어떠한 성능을 보이는지 비교 분석하도록 하겠다.

2. 클러스터 시스템의 다양한 Load Balancer 소개

2.1. Alteon Web Switch – AD3 (AWS)

AWS 는 하드웨어 형태의 L4 스위치이다. 로드밸런싱(Load Balancing)을 사용하여 웹 트래픽 관리가 가능하고, 다양한 어플리케이션을 지원하며, 네트워크의 다운 타임이 없이 안정적인 네트워크 확장이 가능하다. 또한 헬스체크(Health-check), 다양한 모니터링과 글로벌 로드밸런싱을 사용하여 가용성을 높이며, 방화벽의 역할도 가능하기 때문에 보안성이 높다. 그리고 하드웨어 지향적이기 때문에 가격이 비싸다. [3]

2.2. ALBM Traffic Manager (TM)

ALBM (Adaptive Load Balancing and Management) TM 은 소프트웨어 형태의 L4 스위치이다. 확장성 있는 클러스터 시스템 구조를 가지고 있고, 다양한 형태의 멀티 클러스터를 지원한다. 다양한 플랫폼에서 사용이 가능하고 웹 콘솔로 실제 서버나 로드밸런서를 모니터링하고 조정하는 것을 포함하여 시스템 전체를 제어 할 수 있다. 또한 실제 서버들의 에이전트를 사용하여 ARR 이나 ALC 를 사용할 수 있다. [4]

2.3. Linux Virtual Server (LVS)

LVS 도 소프트웨어 형태의 L4 스위치로서, 리눅스에서 클러스터링 서비스를 제공하기 위해 만들어졌다. 리눅스 시스템에서 돌아가는 로드밸런서를 사용하여 확장성이 높은 클러스터 시스템을 쉽고 싸게 만들고, 모니터링과 헬스체크 등 여러 소프트웨어를 사용하여 높은 가용성을 얻을 수 있다. [5]

3. 시스템 성능 차이

웹 클러스터에서 시스템 성능에 크게 영향을 주는 것은 실제 서버의 성능, 로드밸런서의 성능, 부하 분산 스케줄링 알고리즘의 선택이다. 이 중 실제 서버는 동일한 것들을 사용하고 각 시스템의 부하 분산 스케줄링 알고리즘도 역시 거의 비슷한 성능을 가질 때 성능의 특징이 나타나는 요소는 L4 스위치, 곧 로드밸런서이다.

AWS 는 하드웨어적인 스위치 형태를 가지기 때문에 내부의 실제 서버들의 앞 단에 위치하여 모든 패킷 출입의 제어를 통해 로드밸런싱을 하게 된다. 과정이 한 단계 빠르고 하드웨어 단에서 처리를 하기 때문에 빠른 처리가 가능하다. 반면에 논문에서 사용한 TM 과 LVS 는 요청 패킷이 스위치를 통해 실제 서버들이 속한 세그먼트로 들어오고, 세그먼트에 있는 로드밸런서에서 그 패킷을 받아 정해진 알고리즘에 따라 정해진 실제 서버에게 전달하는 과정을 가진다. 이 시스템들은 소프트웨어적인 특성에 따라 설치되는 머신의 성능에 종속되게 된다.

4. 다양한 부하의 종류

서버에서 제공하는 웹 서비스에 따라 서버에 가중되는 부하의 특성이 달라지게 된다. 이 논문에서 사용된 부하 특성은 다음과 같다. [6]

4.1. Fine-Grain Request

내용이 없어 크기가 매우 작은 문서들을 요청하는 것을 말한다. 서버에서 요청을 처리하는데 사용되는 자원이 가장 적고 네트워크 대역폭을 적게 사용하기 때문에, 일반적으로 어떤 시스템의 최대 연결 개수를 측정하는데 사용된다.

4.2. Fixed Size Fixed Duration (FSFD): 클라이언트에 의해 요청되는 각 작업의 양과 수행 시간이 일정한 경우이고 정적 웹 페이지의 요청하는 경우를 예로 들 수 있다.

4.3. Fixed Size Various Duration (FSVD): 클라이언트에 의해 요청되는 각 작업의 양은 일정하지만, 서버 측에서 수행되는 시간이 변하는 경우이다.

4.4. Various Size Fixed Duration (VSFD): 클라이언트에 의해 요청되는 각 작업의 양이 다르고, 서버 측에서 수행되는 시간은 일정한 경우이다. 수행되는 시간이 충분히 길게 하여 작업의 크기에 영향을 받지 않는 경우이다.

4.5. Various Size Various Duration (VSVD): 클라이언트에 의해 요청되는 각 작업의 양과 수행 시간이 변하는 경우이고, 동영상 서비스를 예로 들 수 있다.

4.6. Htm + Memory: Fine-Grain request 와 VSVD 를 60:40 의 비율로 랜덤 하게 요청하는 것이다. Fine-Grain request 를 사용함으로써 각 클러스터 시스템의 성능수치의 차이를 더 크게 볼 수 있게 된다.

4.7. Web-Tree Request: Web-bench 의 standard workload tree[7]는 보다 정확한 벤치 마크를 위하여 일반적인 웹 부하에 따라 서버에 저장해 놓은 미리 정의된 정적, 동적인 페이지들이다. 이 페이지들을 요청하는 것이다

위와 같은 성향에 따른 자원 사용량을 가지적으로 측정하기 위하여 메모리를 주된 부하로 정한다. 메모리는 컴퓨터의 성능에 직접적으로 영향을 주며, 또한 부하를 요청할 때 사용되는 메모리의 크기나 사용시간을 비교적 정확하게 조절할 수 있기 때문에 가상의 부하를 주기에 용이하기 때문이다.

5. 실험 환경

실험에 사용한 클러스터 시스템의 사양과 클라이언트 컴퓨터들의 주요 사양, 그리고 환경은 다음과 같다.

AWS 는 Network Address Translation (NAT)방식을 사용하고 100Mbits*8 포트 중 3 포트를 클라이언트들에게 할당한다. TM 은 Direct Routing 을 사용하고 실제 서버들에게 클러스터 시스템을 운용하기 위해 에이전트들을 설치해 놓는다. LVS 는 ipvs 0.8.0 을 사용하고 Direct Routing (DR)방식을 사용한다.

웹 클러스터 시스템을 구성한 실제 서버들은 4 대로 고정시켜 놓고 클라이언트 수는 부하에 따라 가변

적으로 한다. 서버들은 Dual CPU(pIII-900MHz * 2)와 Memory (512MB)를, 클라이언트들은 CPU (pIV-1.4GHz)와 Memory(256MB)를 사용한다. 네트워크 대역폭은 100MBit 이고, 테스트 툴은 Web-bench tool 을[8] 사용하고 모니터링 툴은 Perfmon 을[9] 사용한다.

각 클러스터 시스템의 성능을 보다 정확히 측정하기 위하여 클러스터를 구성하는 서버들을 동종의 것들로 구성하였고, 모든 실험에서의 성능 비교치는 Request/Second 로 한다. 또한 FSFD, FSVD, VSFD, KSVVD 에서는 ASP file 을 사용하여 요청되는 메모리의 크기나 사용시간을 비교적 정확하게 조절한다.

이 논문에서 부하 분산 스킴은 LC 와 ALC 만을 사용한다. 일반적으로 LC 가 RR 보다 나은 성능을 가지고 있고, 클러스터를 구성하는 서버들이 동종이기 때문에 WLC 를 사용할 필요가 없다.

Web-bench tool 은 클라이언트들에 있는 에이전트들을 사용하여 웹 트래픽을 일으키는데 이 때 컨트롤러에서 여러 파라미터(Parameter)들을 조절하여 한 순간에 클라이언트들이 일으키는 트래픽의 양을 결정한다. 그리고 부하요청은 여러 파일들을 랜덤(Random)하게 요청한다.

6. 실험 결과

여러 대의 클라이언트를 사용하여 다양한 부하 상황을 만들고 AWS, TM, LVS 를 같은 상황에 놓이게 하여 성능을 측정하였다.

Fine-Grain request 를 사용하면 실제 서버들에 부하를 주지 않고 로드밸런서에서 병목현상이 생기는 트래픽의 양을 알 수 있기 때문에 AWS, TM, LVS 의 성능최대치를 측정할 수 있다. 그림 1 에서 보면 AWS 가 Fine-Grain request 부하 상황에서는 다른 제품에 비해 2~3 배 많은 양의 트래픽을 처리할 수 있는 것을 확인할 수 있다. 이는 AWS 가 하드웨어적인 우수성을 가지고 있기 때문이다.

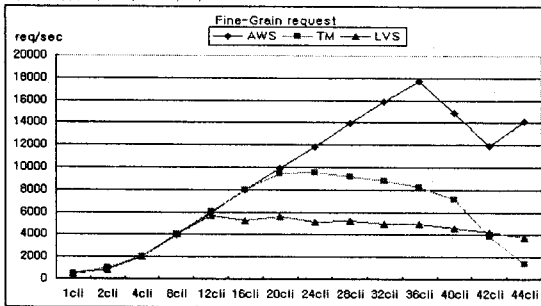


그림 1. Fine-Grain request 에서의 성능그래프

그림 2 에서 보면 FSFD 부하 상황일 때 AWS, TM, LVS 의 최대 성능치들이 거의 비슷하게 측정된다. 부하 특성이 고정적이기에 부하 분산 서버에서 나뉜 부하가 각 서버에서 처리될 때, 서버간의 자원 불균형이 일어나지 않기 때문이다. 또한 병목 현상이 로드밸런서가 아닌 실제 서버들이기 때문이기도 하다. AWS 는 Fine-Grain request 때와는 다르게 오히려 성능이 약

간 떨어진다. 이 현상의 한가지 이유로는 IP 로드밸런싱을 할 때 NAT 방식과 DR 방식의 성능 차이에서 오는 것이다. 일반적으로 같은 상황일 때 NAT 가 DR 에 비해 성능이 떨어진다.

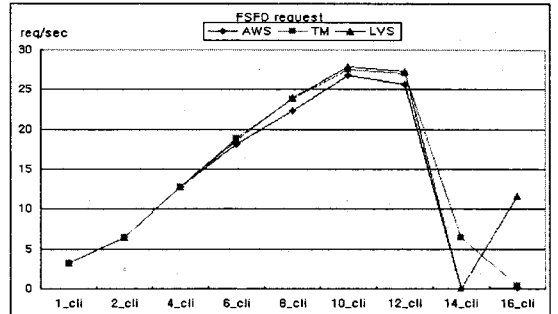


그림 2. FSFD request 에서의 성능그래프

그림 3 은 FSVD 부하 상황에서의 성능들을 나타낸다. 부하의 특징이 같은 양을 다른 시간 동안 요청하는 것인데, 각 제품의 LC 들이 비슷한 성능을 내주지 않는다. TM 과 LVS 가 안정적인 성능을 보여주는 것에 비해 AWS 는 상대적으로 좋지 않은 성능을 보여준다.

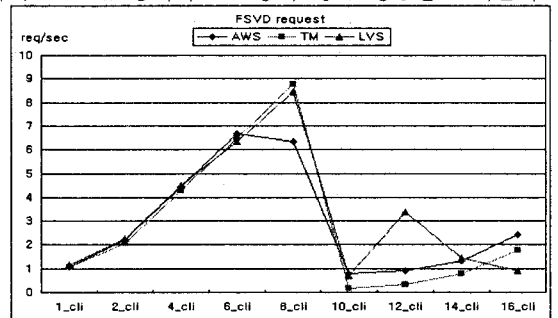


그림 3. FSVD request 에서의 성능그래프

VSFD 부하 상황에서는 그림 4 에서 보는 것처럼 비슷한 형태의 성능 그래프들이 나타나나, 실제 서버에서 병목현상이 나타나는 부하 상황에서의 다른 실험들처럼 AWS 의 성능이 상대적으로 조금 떨어지는 현상이 나타난다.

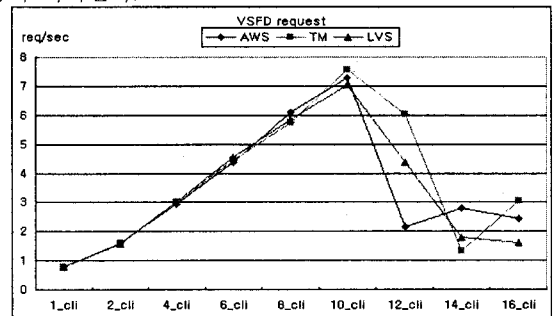


그림 4. VSFD request 에서의 성능그래프

가장 변화가 심한 VSVD 부하 상황에서는 기본적

인 LC 는 비슷한 최고 성능치를 보인다. 하지만 VSVD 와 같이 서버간의 불균형을 심하게 야기시키는 부하 상황에서는 TM 의 ALC 가 더욱 효과적인 것으로 나타났다. 그림 5 에서 X 축의 8 클라이언트일 때 TM 의 ALC 만이 안정적인 성능을 나타내는 것을 볼 수 있다.

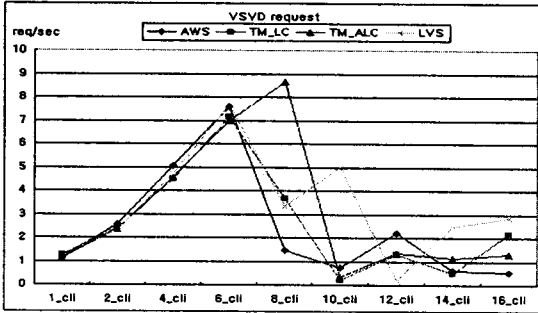


그림 5. VSVD request 에서의 성능그래프

VSVD 와 같이 순수하게 메모리에 대한 부하만 요청했을 경우 성능의 차이가 있더라도 수치적으로 너무 적어 VSVD 와 htm, 즉 Fine Grain request 의 부하를 적절한 비율로 섞어 부하를 만들어 보았다. 예상하기로는 어느 정도 Fine Grain request 가 있기 때문에 그 부하 상황에서 매우 좋은 성능을 보인 AWS 가 좋은 성능을 보일 것이라고 예상하였으나 결과는 그림 6 에서처럼 그렇지 않다. 여전히 TM 이 가장 좋은 성능을 보이며 오히려 AWS 는 더욱 나쁜 성능을 보인다.

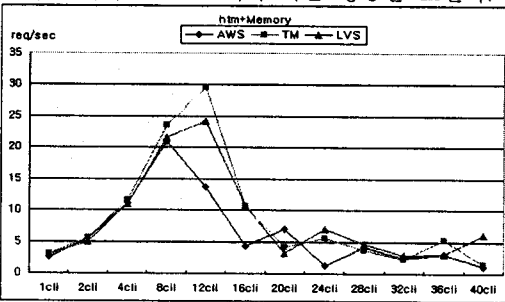


그림 6. Htm+Memory request 에서의 성능그래프

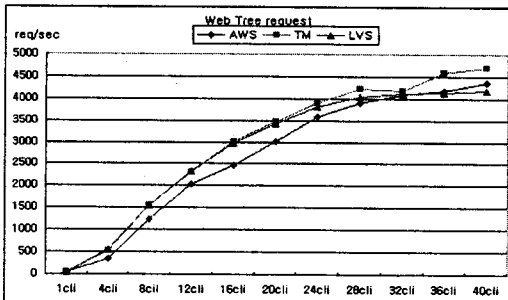


그림 7. Web Traffic request 에서의 성능그래프

일반적인 웹 트래픽의 특성을 가지고 있는 Web Tree request 의 부하 상황에서는 그림 7 과 같이 약간

의 차이가 있지만 AWS, TM, LVS 모두 안정적인 성능을 보여준다.

클러스터가 받고 있는 부하 상황이 어떠한 지에 따라 요청을 처리하는 실제 서버들의 상태가 변하게 되고, 이것은 실험 결과를 통해 클러스터 시스템 성능에 많은 영향을 끼치는 것을 확인할 수 있다. AWS 는 실제 서버들에게 부하가 많이 가지 않는 Fine-Grain 과 같은 경우에 매우 좋은 성능을 보이고, TM 은 실제 서버들이 부하로 인해 과부하 상태로 변하면 상대적으로 좋은 성능을 보이게 된다. 즉, 여러 부하 상황에서 각각의 클러스터 시스템들을 같은 상태로 세팅을 해도 제품 특성에 따라 다른 성능을 보이는 것을 확인할 수 있다.

7. 결론

이 논문에서는 Fine-Grain request, Various Memory request 4 개, Web-Tree request 와 같은 다양한 부하 상황과 Alteon Web Switch, ALBM TM, LVS 들로 구성된 웹 클러스터 시스템들에 대하여 알아보고 여러 부하상황에서 각 클러스터들의 성능을 측정하고 분석하였다. 같은 부하 상황이라도 사용한 웹 클러스터 시스템이 어떠한 특성을 가지고 있느냐에 따라 성능이 달라지는 것을 확인하였다.

이 결과들로 웹 클러스터 시스템의 부하 상황을 분석하는 것에 대한 중요성을 강조할 수 있다. 부하 상황이 정확하게 분석되었을 때 현재 제공하는 서비스에 알맞은 클러스터 시스템을 선택하고 구성할 수 있기 때문이다. 특히 비용을 고려하면, 부하 분석 없이 비싼 제품을 사용하는 것은 오히려 성능뿐만 아니라 비용에까지 효과적이지 않을 수 있기 때문에 반드시 필요하다.

참고문헌

- [1] Rajkumar Buyya, "High Performance Cluster Computing vol. 1," Prentice Hall, 1999
- [2] George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems-Concepts and Design," 3rd Edition, Addison Wesley, 2001
- [3] "Alteon Web Switches," <http://www.nortelnetworks.com/products/01/alteon/webswitch/index.html>
- [4] 임유진, 이원규, 한영태, 이동훈, 최은미, "Linux LVS 와 ALBM 클러스터 시스템의 성능평가 및 비교분석," 18 회 추계학술발표논문집, 한국정보처리학회, 2002
- [5] "LVS documents", <http://www.linuxvirtualserver.org/Documents.html>
- [6] 임유진, 이원규, 최은미, "부하 특성에 따른 분산 스케줄링 알고리즘의 성능 평가 및 비교," 19 회 추계학술발표 논문집, 한국정보처리학회, 2003
- [7] "WebBench's standard workload tree", <http://www.etestinglabs.com/benchmarks/webbench/3w/about/3w/wrktree.asp?visitor=X>
- [8] "WebBench", <http://www.veritest.com/benchmarks/webbench/webbench.asp?visitor=X>
- [9] "Monitoring and Tuning Your Server," <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/reskit/iis50rg/iischp5.asp>