

분산 시스템의 효과적인 웹 클러스터 성능 모니터링과 분석

김기, 최은미

한동대학교 전산전자공학부

e-mail : bart@seed.handong.edu, emchoi@handong.edu

An Effective Performance Monitoring and Analysis for a Web Cluster on a Distributed System

Ki Kim, Eunmi Choi

Dept. of Computer Science & Electronic Engineering, Handong Global University

요 약

인터넷 서비스 서버들의 가용성과 확장성, 부하분산의 특성들을 가지는 클러스터 시스템에서 성능 관리와 이상상황 관리를 위해서 본 논문에서는 성능 모니터링을 통하여 클러스터 시스템으로부터 필요한 자료를 수집할 수 있는 구조와 성능 분석을 위한 수집해야 하는 정보들의 분석과 수집된 데이터를 분석하기 위한 다양한 분석 방법론을 제시한다. 이러한 성능 분석을 통해 자원사용, 확장성, 가용성, 부하분산, 서비스의 질, 이상상황 추적 등을 고려하였다.

1. 서론

분산 상에서의 웹 클러스터를 사용함으로써, 인터넷 서비스를 제공하는 서버들을 하나의 형태 (SSI: Single System Image)로 이루어주어, 기하급수적으로 증가하는 인터넷 사용자들의 서비스 요청에 부합하게 처리하도록 서버 단의 성능 향상을 기대하게 되었다. 이 외에도 확장성(Scalability), 고가용성(High Availability), 경제성(Cost-Effectiveness), 이상상황으로부터 신뢰성 (Fault Tolerance)을 가진 고성능 시스템(High Performance System)으로 클러스터 시스템의 기대 효과는 높아지고 있다.[1,2] 이러한 클러스터 시스템을 관리하기 위해서는 설정(Configuration) 관리, 이상상황(Fault) 관리, 성능(Performance) 관리, 보안(Security) 관리, 사용자(User) 관리 측면이 고려되어야 한다.[6]

분산된 서버들의 성능 관리 측면과 이상상황 관리를 위해서 특별히 시스템 모니터링이 필요하다. 성능 모니터링은 서버 관리의 근본적인 부분이며 이상상황 발생의 추세 파악, 서버 추가 혹은 자원확장의 판단 근거, 성능 최적화의 판단 근거를 제공한다. 또한 작업 할당의 불균형으로 인한 전체 시스템의 성능 저하 현상을 파악할 수 있으며 장기간의 서버관리 정책을 수립하는 판단 근거를 제공할 수 있다. 이와 같이 클러스터 시스템의 성능 평가와 분석의 중요성이 대두되고 있다.[2,6,7]

본 논문에서는 클러스터 시스템에서의 효과적인 성능 모니터링과 분석 방법을 제시하며, 이를 제공하기 위한 클러스터 시스템 상에서의 구성 아키텍처와 분석 방법을 소개한다.

다음 절에서는 모니터링에 필요한 아키텍처 구조를 설명하며, 3 절에서는 분석을 위해 수집해야 하는 데이터, 4 절에서는 수집된 자료를 통해 분석방법과 5 절에서는 분석을 통해 고려할 수 있는 결정사항을 설명하고 6 절에서 결론을 내린다.

2. 클러스터 시스템에서 성능 모니터링을 위한 구조

클러스터 시스템의 성능 모니터링을 위한 구성요소는 크게 서버노드, 수집, 수집관리 및 보고서 생성, 데이터 베이스의 4 부분으로 나뉜다. 각각의 Sub-System 별의 컴포넌트 설계 구조도를 그림 1에서 볼 수가 있다.

2.1 Agent Domain Subsystem

각 서버 노드에 설치된 Agent 는 설정된 주기에 따라 성능수치를 읽어 Collector 에게 전달하며 또한 능동적으로 시스템 운영 중에 발생하는 상태변화 이벤트와 이상 상황을 Collector 와 다른 Agent 에게 전달한다.

2.2 Collector Domain Subsystem

각 서버 노드의 Agent 로부터 이벤트와 성능수치를 수집하고 수집된 정보를 Database 에 저장하며 주기적으로 Database 에 저장된 실시간 성능정보를 분, 시간, 일 단위로 Average, Maximum, Minimum 값을 계산하여 해당하는 Database Schema 에 저장한다.

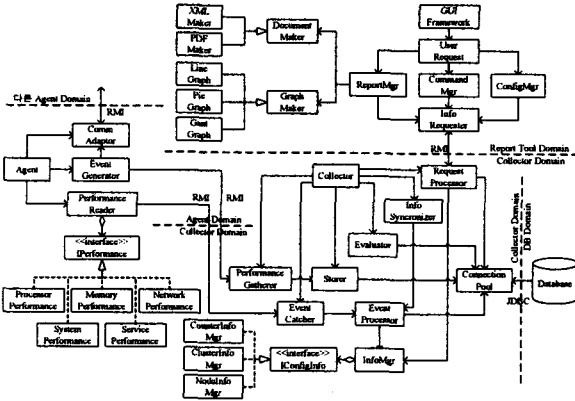


그림 1 모니터링을 위한 Sub-System 별 컴포넌트 설계 구조도

2.3 Report Tool Domain Subsystem

사용자에게 성능 모니터링의 관리를 위해서 어떤 Agent 가 어떤 성능수치를 어떤 수집주기로 수집 할 것인가를 결정하는 기능과 클러스터 시스템을 분석할 수 있는 보고서 생성 기능을 제공한다.

2.4 Database Domain Subsystem

각 서버 노드에 위치한 Agent 를 통하여 수집된 성능 수치와 발생된 이벤트를 저장하게 되며 클러스터 시스템 분석과 보고서 생성시 자료를 제공하게 된다

3. 성능 모니터링을 위해 수집하는 정보

성능 모니터링을 위하여 수집하는 정보는 성능에 관련된 정보, 이벤트 데이터 정보, 서비스의 질(Quality), 클러스터 시스템의 정적인 정보로 구성이 된다.

3.1 성능 데이터

1) 프로세서 성능 데이터

프로세서에 관련된 Processor Load (프로세서가 사용된 시간의 백분율), User Time (사용자 모드에서 소비한 프로세스 시간의 백분율), Privileged Time (특권 모드에서 명령 실행하면서 경과된 시간의 백분율), Interrupt / sec (프로세스가 받아 처리한 하드웨어 인터럽트의 초당 발생 횟수) 등의 성능 데이터를 수집한다.

2) 메모리 성능 데이터

메모리에 관련된 Available Memory (실행되는 프로세스에 사용할 수 있는 실제 메모리의 양), Page Fault / sec (초당 페이지 부재의 평균 수), Page Read / sec (하드 페이지 부재를 해결하기 위해 디스크를 읽은 비율), Page Write / sec (실제 메모리의 공간을 비우기 위해 페이지를 디스크에 쓴 비율) 등의 성능 데이터를 수집한다.

3) 네트워크 성능 데이터

네트워크에 관련된 TCP Connection (TCP 연결 개수), TCP Connection Failure (TCP 연결에 실패한 개수), Datagrams Received / sec (UDP 데이터그램을 받은 비율), Datagrams Sent / sec (UDP 데이터그램을 보낸 비율), Bytes

Received / sec (네트워크 어댑터에서 받는 바이트의 비율), Bytes Sent / sec (네트워크 어댑터에서 보내는 바이트의 비율) 등의 성능 데이터를 수집한다.

4) 시스템 성능 데이터

시스템에 관련된 Processor Queue Length (프로세서 대기열에 있는 스레드 수), Context Switches / sec (모든 프로세서가 한 스레드에서 다른 스레드로 전환한 전체 횟수), Processes (프로세스의 수), Threads (스레드의 수) 등의 성능 데이터를 수집한다.

5) 서비스 성능 데이터

Application 의 종류에 따라 수집하는 성능수치는 달라질 수 있으며 다음에서는 Window 2000 Server 의 IIS Web Server 와 관련된 성능수치의 예로서 Request / sec (초당 실행한 요청의 수), Web Bytes Recived / sec (서비스가 받은 데이터 바이트의 비율), Web Bytes Sent / sec (서비스가 보낸 데이터 바이트의 비율) 등의 성능 데이터를 수집한다.

3.2 이벤트 데이터

1) 상태변화 이벤트

클러스터 시스템의 상태변화 이벤트로서 Node Active(새로운 노드가 클러스터 시스템에 추가됨을 나타내는 이벤트), Node Fail (노드가 시스템이나 전원, 네트워크 등의 문제로 클러스터 시스템으로부터 이탈을 나타내는 이벤트), Node Overload (노드의 상태가 과도한 서비스 요청으로 overload 상태에 빠지게 됨을 나타내는 이벤트), Node Underload (노드의 상태가 overload 상태에서 정상상태로 돌아옴을 나타내는 이벤트)를 수집한다.

2) 운영 이벤트

클러스터 시스템의 운영 이벤트로서 Cluster Created (새로운 클러스터가 생성됨을 나타내는 이벤트), Cluster Removed (클러스터가 제거됨을 나타내는 이벤트), Node Join to Cluster (노드가 클러스터에 Join 하여 서비스하게 됨을 나타내는 이벤트), Node Leave from Cluster (노드가 클러스터에서 빠지게 됨을 나타내는 이벤트)를 수집한다.

3) 서비스 이벤트

클러스터 시스템의 서비스 이벤트로서 Service Start (서비스가 시작되어 클라이언트로부터의 요청을 받아 들이게 됨을 나타내는 이벤트), Service Stop (서비스가 중지되어 클라이언트로부터의 요청을 받아 들일 수 없음을 나타내는 이벤트), Service Pause (서비스를 잠시 멈추어 기존 세션의 서비스는 계속 수행하지만 새로운 서비스를 받아들이지 않음을 나타내는 이벤트), Service Resume (서비스가 Pause 상태에서 복귀를 나타내는 이벤트)를 수집한다.

3.3 서비스의 질

클러스터 시스템의 서비스 질을 측정하기 위해 Service Response Time (클라이언트가 클러스터 시스템에게 서비스를 요청하여 응답을 받는 시간을 표현하기 위해 각 서버 노드에게 주기적으로 서비스를 요청을 보내어 응답시간) 을 측정한다.

3.4 서버 노드 정보

클러스터 시스템을 구성하는 서버 노드의 정적인 시스템 정보로서 OS Name (서버 노드의 운영체제), Number of Processor (서버 노드가 가지고 있는 CPU의 개수), Processor Type (CPU의 종류), Processor Clock Speed (CPU의 Clock Speed), Physical Memory (서버 노드가 가지고 있는 Memory의 크기), File System Type (파일 system의 종류), Host Name (서버 노드의 Host Name), Domain Name (서버 노드가 속한 Domain)을 수집한다.

4. 성능 분석 방법

4.1 성능수치 vs. 시간

시간의 변화에 따른 성능수치의 변화를 분석한다. 여기서 성능수치는 자원사용, 서비스 요청, 서비스의 질이 될 수 있으며, 이 분석을 통하여 특정 자원의 사용이 시간에 따라 어떻게 변화 하는가, 어떤 자원의 부족함이 서비스의 제공에 문제가 되는가, 서비스의 요청이 시간에 따라 어떻게 변화 하는가, 서비스의 요청이 집중되는 시간대, 서비스의 응답시간의 변화와 또한 여러 서버 노드의 성능수치를 같이 비교함으로써 부하분산이 잘 이루어 졌는가를 분석 할 수 있다.

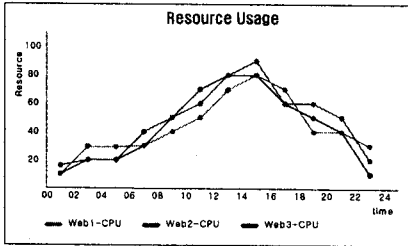


그림 2 성능수치 vs. 시간 분석을 위한 그래프

4.2 이벤트 vs. 시간

상태변경, 이상상황 발생의 이벤트를 발생 시간 순서와 종류별로 분석한다. 이 분석을 통하여 클러스터 시스템이 이상상황에 빠지기까지의 상태가 어떻게 변했는지의 과정을 Error Trace이나 Status Trace를 함으로써 문제의 원인을 찾을 수 있으며, 이상상황 발생의 추세를 파악하여 클러스터 시스템의 운용에 대한 도움을 얻을 수 있다.

4.3 성능수치 vs. 성능수치

성능수치간의 상관관계를 분석한다. 이 분석을 통하여 자원과 자원의 관계, 서비스 요청과 자원의 관계, 서비스 요청과 서비스의 질과 관계, 서비스의 질과 자원의 관계를 분석하여 파악 할 수 있다

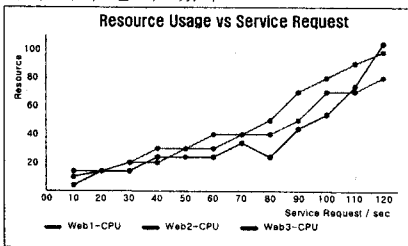


그림 3 성능수치 vs 성능수치 분석을 위한 그래프

4.4 성능수치 vs. 이벤트

성능수치와 이벤트간의 상관관계를 분석할 수 있다. 성능수치와 이벤트를 종합해서 살펴 봄으로써 이상상황에 빠진 시스템의 상태와 원인을 더욱 정확히 분석 할 수 있다

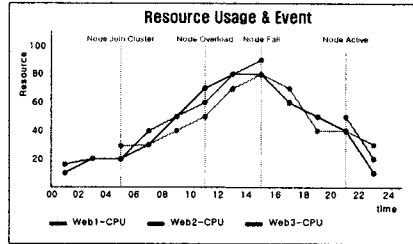


그림 4 성능수치 vs 이벤트 분석을 위한 그래프

4.5 이벤트 vs. 이벤트

이벤트간의 상관관계를 분석한다. 이 분석을 통하여 특정한 이벤트 발생 전후 일정 시간 내에 발생한 다른 이벤트의 빈도조사, 특정 두 이벤트가 일정 시간 내에 발생한 빈도 조사, 특정 시간 내에 발생한 이벤트의 빈도 조사, 특정 두 이벤트가 발생한 시간 간격 조사, 여러 이벤트가 순차적으로 발생한 패턴의 빈도를 조사하여 분석 할 수 있다

5. 성능 분석을 통해 고려할 수 있는 결정사항

5.1 자원사용

서버 노드의 자원사용을 나타내는 대표적인 성능수치로는 Processor Load, Available Memory, Network In Byte, Network Out Byte 등이 있다. 위에서 다루었던 성능수치 vs 시간 분석을 통하여 자원사용의 추세와 최대 사용량을 파악하고, 어떤 자원의 부족으로 인한 서비스의 제약이 있는지 확인하고 scale-up 혹은 scale-out의 판단 기준으로 삼을 수 있다.[3]

5.2 확장성

서비스와 밀접한 관계를 가지는 자원의 사용량이 임계값을 넘는 경우는 서버성능을 scale-up (각 서버 노드의 자원을 upgrade 함으로 확장하는 것) 하거나 혹은 서버의 일부 자원의 확장만으로는 서비스를 증가 할 수 없을 경우와 클러스터 시스템의 가용성을 높이고자 하는 경우엔 서버 개수를 scale-out (클러스터에 서버 노드를 추가함으로 확장하는 것)하는 결정 기준으로 삼을 수 있다.[5]

5.3 가용성

가용성에 관한 결정은 이벤트 vs 시간 분석을 통한 클러스터 시스템의 서비스의 가용성 자료와 노드들의 가용성 자료를 통하여 분석이 가능하다. 장기적인 가용성 정책 결정 시 고려 사항은, 전체 클러스터 시스템의 가용률(A_C)이 높아 서비스가 정지하지 않았더라도 노드들의 가용률의 평균값(Avg_{AN})이 낮다면 클러스터 시스템에 노드를 더 추가하여 클러스터 시스템의 잠재적 서비스 실패 가능성을 낮출 필요성이 있다.[4]

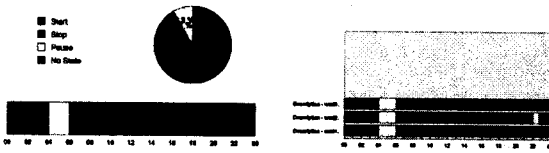


그림 5 클러스터와 노드의 가용성 그래프

$$A_N = \frac{D_{NS}}{D_{NS} + D_{NF}}$$

A_N : 노드의 가용성
 D_{NS} : 노드의 서비스 시간
 D_{NF} : 노드의 장애 시간

$$AVG_{AN} = \frac{A_{N1} + A_{N2} + \dots + A_{Ni}}{i}$$

AVG_{AN} : 노드들의 평균 가용성

$$A_C = \frac{D_{CS}}{D_{CS} + D_{CF}}$$

A_C : 클러스터의 가용성
 D_{CS} : 클러스터의 서비스 시간
 D_{CF} : 클러스터의 장애 시간

5.4 부하분산

클러스터 시스템에서는 Round Robin, Least Connection, Weighted Round Robin, Weighted Least Connection 등의 부하분산 방식을 사용하여 클라이언트로부터 들어온 Service Request 를 서버 노드들에게 분산시키게 된다. 부하분산 방식의 결정과 부하분산이 어떻게 이루어졌는지 분석하기 위해서는 Service Request 를 표현하는 성능수치가 무엇인지 찾아야 하는데 Web Service 를 하는 클러스터 시스템의 예에서는 TCP Connection 이 해당하는 성능수치가 된다. 한 클러스터에 속한 노드들의 TCP Connection 수치를 같이 비교함으로써 분석이 가능하다.

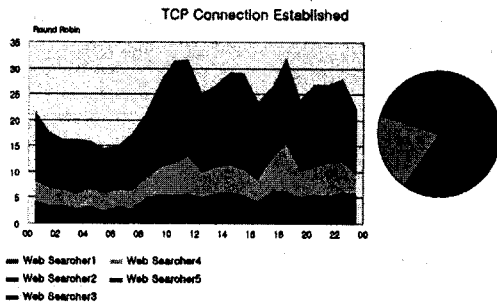


그림 6 클러스터 시스템의 부하분산 분석 그래프

5.5 서비스의 질

클러스터 시스템의 서비스의 질을 나타내는 한 방법으로써 여기서는 응답시간을 사용하고자 한다. 응답시간은 클라이언트가 서비스 요청을 보내어 클러스터 시스템으로부터 응답을 받기까지의 시간을 의미하며, 이것을 측정하기 위해서는 각 서버 노드의 서비스를 제공하는 application 에 주기적으로 서비스 요청을 보내어 응답을 받기까지의 시간을 측정하는 것이 필요하다. 이 데이터를 앞에서 언급한 성능수치 vs 시간 분석, 성능수치 vs 성능수치 분석을 통해 응답시간이 시간에 따라 어떻게 변하며 어떤 자원의 사용과 연관이 있는가를 파악하여 응답시간을 줄이기 위한 노력이 필요하다.

5.6 이상상황 추적

클러스터 시스템이 이상상황에 빠지는 이유를 파악하기 위해서는 각 컴포넌트들이 어떤 동작, 상태 혹은 설정의 변경, 어떤 문제 봉착 시에 이벤트를 발생시키고 그것들을 수집함으로써 그 단계를 되짚어 볼 수 있다. 또한 그 각 단계마다 자원사용의 상황을 같이 고려하여 시스템의 상태와 원인을 자세히 파악해야 한다. 앞에서 언급한 이벤트 vs 시간 분석을 통해 발생한 이벤트들의 history 를 살펴 볼 수 있으며, 이벤트 vs 이벤트 분석을 통해 이벤트간의 연관성(Event Correlation)과 근본 원인(Root Cause)을 파악할 수 있으며, 이벤트 vs 성능수치 분석으로 이벤트가 발생한 시점에서의 시스템의 상황을 자세히 파악 할 수 있다

6. 결론

클러스터 시스템의 성능 관리와 이상상황 관리를 위해서 성능 모니터링을 통하여 분산된 시스템으로부터 필요한 자료를 수집할 수 있는 구조와 수집된 자료를 바탕으로 클러스터 시스템의 성능을 분석할 수 있는 방법을 제시하였다. 클러스터 시스템의 성능을 분석하기 위해 수집해야 하는 요소로는 성능 데이터, 이벤트 데이터, 서비스의 질, 서버 노드의 정보가 있으며, 수집된 데이터를 분석하기 위해 성능수치 vs 시간 분석, 이벤트 vs 시간 분석, 성능수치 vs 성능수치 분석, 성능수치 vs 이벤트 분석, 이벤트 vs 이벤트 분석이 필요하다. 수집된 데이터와 여러 분석 기능을 통해 자원사용, 확장성, 가용성, 부하분산, 서비스의 질, 이상상황 추적 등의 결정사항을 고려할 수 있다

향후 연구로는 서버의 개수나 자원확장과 같은 관리가 요구되는 것을 명시해 줄 수 있는 기능과 클러스터 시스템이 이상상황에 빠지는 패턴을 습득하여 이상상황 발생 가능성을 경고하거나 회피하는 방법들을 연구를 할 계획이다.

참고문헌

- [1] Andrew S. Tanenbarum, Maarten van Steen, "Distributed Systems principles and Paradigms", Prentice Hall, 2002
- [2] Rajkumar Buyya, "High Performance Cluster Computing vol. 1", Prentice Hall, 1999
- [3] "Monitoring and Tuning Your Server", Microsoft Technews <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnolog/iis/reskit/iis50Org/iischp5.asp>
- [4] Wensong Zhang, Shiyao Jin, Quanyuan Wu, "Creating Linux Virtual Server", LinuxExpo 1999 Conference, <http://www.linuxvirtualserver.org/ols/lvs.ps.gz>
- [5] Wensong Zhang, "Linux Virtual Server for Scalable Network Services", Ottawa Linux Symposium 2000, <http://www.linuxvirtualserver.org/ols/clvs.ps.gz>
- [6] Henz-Gerd, Sebastian Abeck, Bernhard Neumair, Integrated Management of Network Systems, Morgan Kaufmann, 1998.
- [7] Zeisler, Varma, Wallace, Kalich, Xion, "TMN CORBA Matures: A Service Provider Gateway for Measuring Service Availability", IEEE Network Operations and Management Symposium, Vol.2 1998, pp374-380