

클러스터링 웹 서버 모니터링 시스템의 설계 및 구현

이은⁰ 한인

(주)뱅크타운 부설기술연구소 시스템체계연구실

{elee⁰, ihan}@banktown.com

Design and Implementation of Clustering Web-Server Monitoring System

Eun Lee⁰ In Han

Dept. of R&D Center, Banktown Inc.

요 약

본 논문은 쌍 서버들을 클러스터로 구성한 서비스 서버들의 모니터링 항목 값을 인터넷으로 연결된 모니터링 서버에서 수집할 수 있는 모니터링 시스템을 구성한다. 본 시스템은 MS .NET 플랫폼에서 제공하는 성능 모니터 클래스를 사용하여 구현하였으며, 일반적인 모니터링 시스템과 달리 토픽기반 메시지 처리기를 사용하여 다양한 클라이언트의 요구에 적절히 대응할 수 있고, 지역 네트워크뿐만 아니라 인터넷으로 연결된 원격 네트워크의 클라이언트로 수집된 모니터링 항목 값을 전송할 수 있다.

1. 서 론

PC의 사양과 성능이 우수해짐에 따라 씬(Thin) 서버들을 클러스터[1][2]로 구성하여 사용하는 곳의 수가 크게 증가하고 있다. 또한, 운영체제도 Unix 중심이던 서버들에서 Linux나 마이크로소프트사의 윈도우즈 NT 또는 2000 서버로 전환하는 사이트들도 상당히 빠르게 증가하고 있는 추세이다. 또한, 이러한 클러스터링 웹 서버의 동시 접속자 수, 각 서버의 수신 요청 또는 애플리케이션 서버로의 송신 요청 큐 등의 각종 항목들을 모니터링 하고자 하는 요구도 꾸준히 증가하고 있다.

따라서, 본 논문은 마이크로소프트사의 클러스터링 서버를 구성하여 웹 서비스를 제공하는 시스템을 모니터링 하기 위한 시스템을 설계하고, 구현한다.

일반적으로 모니터링 시스템들은 모니터링 대상 시스템과 같은 네트워크로 구성된 모니터링 서버에서 다양한 방법으로 모니터링 항목 값을 수집하지만, 인터넷상으로 연결된 모니터링 서버를 사용할 수 없거나, 구성하더라도 매우 낮은 성능으로 인해 모니터링의 효과를 크게 얻지는 못한다. 하지만, 본 논문에서 제안하는 모니터링 시스템은 토픽기반의 메시지 처리기를 활용하여 인터넷으로 연결되어 있는 모니터링 서버로 각 서비스 서버들의 모니터링 항목 값을 전송하는 방식을 사용한다.

2장에서 토픽기반 메시지 처리기를 비롯한 관련 연구에 대해 기술하고, 3장에서 본 논문에서 제안하는 모니터링 시스템에 대해서 설명한다. 4장에서 간단한 성능 분석 및 활용 예를 기술하고, 5장에서 결론 및 향후 과제에 대해 논한다.

2. 관련 연구

여러 가지 모니터링 시스템 중에서 본 논문이 제안하는 모니터링 시스템과 가장 연관성이 깊은 마이크로소프트의 성능모니터와, 본 논문에서 사용한 토픽기반 메시지 처리기에 대해 간단히 기술한다.

마이크로소프트의 윈도우즈 서버 운영체제들은 마이크로소프트 “성능 모니터”[3]라는 프로그램이 내장되어 시스템의 상태를 모니터링 할 수 있다. “성능 모니터” 프로그램은 수집할 데이터를 선택하기 위해서 성능 개체, 성능 인스턴스, 그리고 성능 카운터를 지정하여야 한다. 다음 [그림 1]은 이들간의 관계를 도식화 한 것이다.

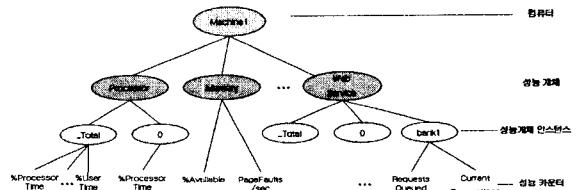


그림 1 성능 개체, 인스턴스, 그리고, 카운터의 상관도

[그림 1]의 성능 개체는 모니터링 할 수 있는 프로세서나 메모리와 같은 시스템 리소스나 모니터 대상 컴퓨터에서 실행되는 응용프로그램과 같은 서비스에 연결된 카운터의 논리집합이다. 성능 인스턴스는 시스템 모니터에서 컴퓨터에 있는 동일한 유형의 여러 성능개체를 구별하기 위한 그룹이다. 그러므로, 성능 인스턴스는 하나 이상이 될 수 있다. 예를 들어 하나의 시스템에서 두 개의 사이트를 구성할 경우 “Web Service” 성능 개체의 성능 인스턴스는 두 개가 존재하게 된다. 그리고, 성능 카운터는 시스템 모니터에서 성능 개체와 연결된 데이터 집합으로서 선택된 각 카운터에 대한 성능 모니터는 성능 개체에 대해 정의한 성능의 특정 측면에 해당하는 값을 제공한다.

마이크로소프트의 .NET 플랫폼[4]에서는 “성능 모니터”와 같은 애플리케이션을 작성할 수 있도록 PerformanceCounter 클래스와 PerformanceCounterCategory 클래스를 제공한다.

마이크로소프트의 “성능 모니터”는 지역 또는 원격지 시스템의 실시간 성능 데이터를 수집하여 도표, 보고서, 그리고 그래프로 표현할 수 있으며, 과거 값에 대한 로그 정보를 남기고, 임계값을 설정하여 임계값 이상의 값이 수집될 경우 관리자에게 경고를 하여 적절한 조치를 취할 수 있도록 한다.

하지만, WAN 환경에서 사용은 가능하지만 너무나 낮은 성능으로 인해 거의 사용이 불가능하고, 로그를 파일로만 기록하여 관리자에게 다양한 방법으로 시스템의 상태를 리포트할 수 없다. 또한, 하나의 히스토그램 표현 창만을 제공하여 여러 개의 데이터들이 섞이므로 실시간 모니터링 그래프로는 적절하지

못하다는 단점을 갖고 있다. 이와 같은 단점을 보완하기 위하여 본 논문에서 제안된 시스템에서는 토픽기반 메시지 처리기 (Topic-Based Message Processor)[5]를 사용한다.

토픽기반 메시지 처리기는 토픽이라는 가상 채널을 통해 메시지를 주고받음을 의미한다. 즉, 메시지 처리기는 다수의 토픽을 유지하고, 임의의 클라이언트는 자신이 수신하고자 하는 메시지와 관련이 있는 토픽에 등록(Subscribe)신청하여 해당 토픽에 발표(Publish)되는 모든 메시지들을 수신하게 된다. 하나의 컴퓨터는 하나 이상의 토픽에 등록과 발표를 할 수 있다. 특수한 경우로 하나의 노드는 토픽에 등록만을 하거나 발표만을 할 수 있도록 하는 경우도 있다. 다음 [그림 2]는 토픽기반 등록/발표 메시지 시스템의 개념도이다.

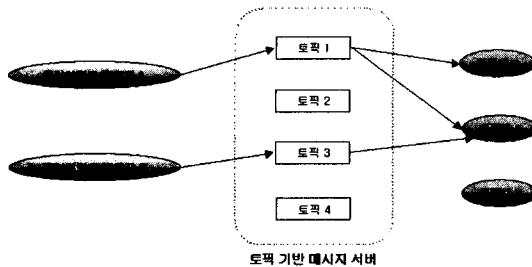


그림 2 토픽기반 메시지 시스템의 개념도

[그림 2]에서 발표자1이 메시지 서버로 메시지를 발표하면 토픽1에 등록한 등록자1과 등록자2가 메시지를 수신하게 된다. 만약 발표자2가 메시지를 전송하게 될 경우는 토픽3에 등록을 한 등록자2만이 메시지를 수신하게 된다. 등록자3은 어떠한 토픽에도 등록을 하지 않았기 때문에 메시지 서버로부터 전송되는 어떠한 메시지도 수신할 수 없다.

3. BtMonitor 시스템

본 논문에서 제안하는 클러스터링 웹 서버 모니터링 시스템의 이름은 BtMonitor(Banktown Monitoring System)라고 명명 한다. BtMonitor는 씬 서버들을 클러스터링으로 구성하여 웹서버를 운영할 때, 각 서버들의 성능 및 서비스를 모니터링하기 위해 사용될 수 있다. BtMonitor는 모니터링 대상 시스템들, 모니터링 항목 값을 수집하는 BtM 서버, 수집데이터를 다양한 형태로 표현하는 BtM 클라이언트 그리고, BtM 클라이언트와 BtM 서버의 메시지를 중개하는 BtM 메시지 서버로 구성된다. 다음 [그림 3]은 BtMonitor의 시스템 구조이다.

[그림 3]에서 BtM 메시지 서버와 BtM 서버가 하나의 노드에

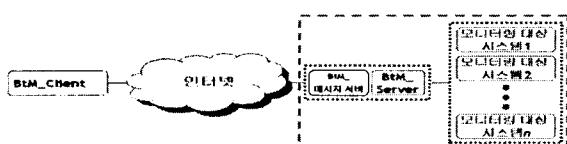


그림 3 BtMonitor 시스템 구성도
존재하지만, 반드시 같은 노드에 존재하여야 하는 것은 아니다.

BtM 서버와 BtM 메시지 서버의 부하량을 고려하여 관리자에 의해서 프로그램의 어떠한 수정이 없이도 다른 노드로 분리할 수 있다.

BtM 서버는 관리자가 설정한 수집 간격과 수집 항목에 따라 주기적으로 원격지 모니터링 대상 시스템의 모니터링 항목을 수집하여 BtM 클라이언트로 전송하기 위하여 BtM 메시지 서버로 전송한다. 메시지는 다음 [그림 4]와 같은 XML 형태를 갖는다.

```
<?XML version="1.0" encoding="EUC-KR"?>
<BANKTOWN-XML type="post" sender="server" receiver="MsgServer" SvcName="BankTown"
<CounterVal MachName="XXX" CounterName="YYY" GetTimes="... "##</CounterVal>
.
.
.
</BANKTOWN-XML>
```

그림 4 수집 서버에서 메시지 서버로 전송하는 메시지 형식

[그림 4]에서 BANKTOWN-XML 엘리먼트의 type 속성 값은 토픽기반 메시지 처리기에서 사용하는 연산의 종류가 표시되고, SvcName 속성 값은 토픽의 이름이 기록된다. CounterVal 엘리먼트는 수집된 항목 값을 나타내고, 사용되는 속성으로 호스트 이름, 항목 값의 이름, 그리고, 수집 시간이 기록된다. CounterVal 엘리먼트의 수는 토픽의 종류에 따라 결정된다.

BtM 메시지 서버는 BtM 서버로부터 수신된 발표 메시지를 해당 토픽에 등록한 모든 등록자(본 시스템에서는 BtM 클라이언트)로 전송한다. BtM 메시지 서버를 중계자로 하여 BtM 서버와 BtM 클라이언트의 메시지 교환을 간략히 도식화하면 다음과 같다.

[그림 5]에서 BtM 서버와 BtM 클라이언트가 메시지 교환을 위해 같은 토픽에 등록함을 알 수 있다. 또한, BtM 서버와 BtM 클라이언트의 등록이 완료되면 BtM 서버에서 발표(Post)되는 모든 수집 데이터 메시지는 BtM 메시지 서버를 통해 전송(Cast)됨을 알 수 있다. BtM 메시지 서버에서 BtM 클라이언트

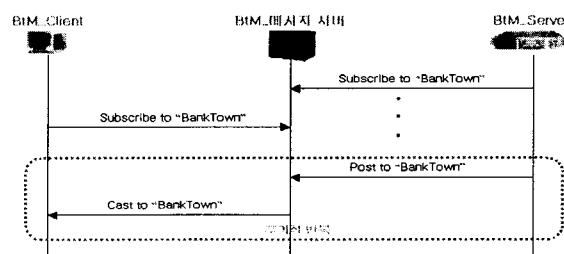


그림 5 BtM 서버와 BtM 클라이언트의 메시지 서버 이용
트들로 전송되는 메시지는 [그림 4]의 메시지와 매우 유사한 형식을 갖는다.

BtM 클라이언트는 수집된 메시지를 분석하여 텍스트 정보와 실시간 그래프를 제공한다. 또한 MySQL을 이용하여 수집된 데이터를 저장하여 일자별 또는 월별 통계자료를 보고서로 출력한다. 보고서는 텍스트 보고서와 그래픽 보고서로 제공한다.

4. BtMonitor의 구현 및 활용 예

BtMonitor는 마이크로소프트의 윈도우즈 플랫폼에서 동작된다. BtM 서버는 .NET의 C#으로 구현되었고, BtM 클라이언트는 .NET의 VB.NET 언어로 구현되었다.

BtM 서버에서 사용한 중요 클래스는 System.Diagnostics 네임스페이스의 클래스들을 사용하였고, 모니터링 항목을 얻기 위해서 사용하는 과정은 다음과 같다.

```
private System.Diagnostics.PerformanceCounter MyPC ;
<중략>
MyPC = new System.Diagnostics.PerformanceCounter();
MyPC.MachineName="#,#,#,#,#";
MyPC.CategoryName = "Web Service";
MyPC.InstanceName = "Psb";
MyPC.CounterName = "Current Anonymous Users";
<중략>
CounterVal = MyPC.NextValue();
```

그림 6 성능 값 얻어오기

BtM 메시지 서버는 실행 연구였던 토픽기반 메시지 처리 시스템 연구의 산출물을 사용하였고, BtM 클라이언트는 BtM 메시지 서버로부터 수신한 메시지를 XML 파일을 이용하여 분석하고, 이를 텍스트 기반과 실시간 그래픽 기반으로 표현하였다. 그래프는 ChartFx for .Net을 사용하였다. 또한, MySQL을 이용하여 로그 데이터를 저장하였다. 그리고, 관리자에 의한 임계값 설정 기능으로 수신된 항목 값이 임계값을 초과할 경우 관리자에게 알람을 할 수 있도록 하였으며, 저장된 데이터는 일자별 통계자료와 월별 통계자료를 텍스트 기반의 보고서와 그래프 보고서로 출력할 수 있도록 하였다.

다음 [그림 7]은 BtM 클라이언트의 한 예이다. 상단에 위치한 그림이 메인 화면으로 실시간 수집 데이터의 텍스트와 그래프를 나타내고, 왼쪽 아래 화면은 텍스트 기반의 통계 보고서, 오른쪽 아래 화면은 그래프 기반의 통계 보고서이다.

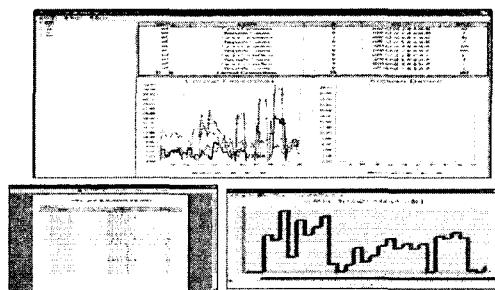


그림 7 BtM 클라이언트의 한 예

(위:메인, 좌하:텍스트보고서, 우하:그래프보고서)

본 논문에서 제안하는 모니터링 시스템을 이용하여, 현재 클러스터링 서버들 중에서 부산은행, 기업은행, 사이버드림타운, 제주은행의 웹 서비스를 모니터하고 있다. 이중 부산은행의 경우는 실제적으로 부산은행 본점에서도 같은 BtM 클라이언트로 서비스를 모니터하고 있다.

본 연구에서 제안하는 모니터링 시스템의 성능을 측정하기 위하여 [표 1]과 같은 환경을 구성하고, 메시지 서버와 수집 서버를 하나의 노드로 구성하여 실험을 하였다.

표 1 성능 측정을 위한 실험환경 값

파라미터	값
모니터링 대상 서버의 수	5대
BtM 클라이언트의 수	10대, 30대
수집 시간 간격	1초
수집 시간	10분
측정항목	프로세서 이용률(%)
네트워크 대역폭	10Mbps

다음 [그림 8]은 실험 결과 그래프이다.

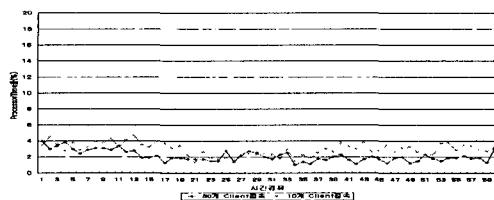


그림 8 클라이언트 접속 수에 따른 프로세서 이용률

[그림 8]의 실험 결과 BtMonitor는 클라이언트 접속 수에 영향을 받지 않으면서, 0~4%의 프로세서 이용률을 보이고 있다. 그래프를 통하여 메시지 서버와 수집 서버를 갖은 노드로 구성하더라도 큰 지장이 없음을 알 수 있다.

5. 결론 및 향후 과제

본 논문에서 토픽기반 메시지 처리기를 활용하여 인터넷으로 연결된 노드에서도 서비스 시스템들을 모니터할 수 있는 시스템을 제안하고 구현하였다. 토픽기반 메시지 처리기를 활용함으로써 최소한의 메시지로 다양한 클라이언트의 요구에 대응할 수 있다는 장점과 지역 네트워크가 아닌 원격 네트워크상의 서비스 서버들을 모니터링 할 수 있다는 장점을 갖는다.

향후 과제는 현재 마이크로소프트 운영체제의 서비스 서버들만이 모니터링 대상 시스템이 될 수 있으나, Unix나 Linux를 운영체제로 하는 서비스 서버들까지 모니터링 대상으로 할 수 있도록 구현을 완료하는 것이다.

6. 참고 문헌

- [1] Windows NT Microsoft Cluster Server, by Richard R. Lee, Osborne McGraw-Hill, 1999.
- [2] Windows 2000 Clustering Technologies: Cluster Service Architecture
- [3] <http://www.microsoft.com/korea/technet/iis/prfrelmn.asp>
- [4] Introducing .NET, James Conard, Patrick Dengler, Brian Francis 외 7명, Wrox Press Ltd.
- [5] <http://developer.knownow.com/devguide/>