대규모 통신 서버 환경하에서의 성능시험 자동화를 위한 환경 설정 및 수집 방법

전재규° 석승학 유재형
{jkchun°, suksh, styoo}@kt.co.kr

Method of Collecting Information and Setting Environment to Automate Performance Test in a Large Communication Server Platform

Jae-Kyu Chun, Seung-Hak Suk, Jae-Hyoung Yoo
KT Network Technology Laboratory

요 약

해당 논문에서는 많은 수의 서버로 구성된 환경에서의 자동화된 성능카운터 설정 및 수집 방안에 대해서 제시하고 대규모 시스템일 경우 많은 시간이 소요되고 번거로운 작업에 대한 자동화 및 카운터 설정, 제어를 위한 스크립트 제시, 리포트 자동 생성 방안을 제시하여 효율적인 시험 및 분석방안에 대해서 설명한다.

1. 서 론

일반적으로, 성능시험의 분석과정은 클라이언트 단에서의 응답시간과 서버 단에서의 성능로그 분석을 통하여 결과 및 원인분석, 시스템의 병목점을 찾는 작업으로 등으로 구성된다. 또한 전체적인 시스템 개발 일정상, 성능시험 후에, 성능시험의 분석결과를 토대로 향후, 수정 및 개선점을 찾고 조속히 시스템에 반영을 해야 하기 때문에 성능시험의 분석 작업은 수작업에 의존하기 보다는 자동화된 방법을 도입함으로써 작업 시간을 단축 시키는 것이 무엇보다도 중요한 요소인 것으로 판단된다.

특히, 20~30대 이상의 서버군으로 구성된 대규모의 서버 환경하에서 성능시험을 진행할 경우에는 성능시험을 위한 환경 설정 및 데이터 수집 등의 작업 또한 상당한 시간을 요하며, 분석 또한 많은 수작업을 필요로 한다 [1][2].

Windows 기반의 서버 환경에서, 성능시험 시 서버에서 수집되는 성능카운터 분석 작업은 설정, 작동(시작 및 중지), 수집, 분석 등 4단계로 구성된다.[1][2]

첫 번째 설정작업은 일단, 특정 서버에 터미널 클라이언트를 통해 접속한 후 필요한 성능카운터를 지정함으로써 작업이 시작된다. 지정 작업이 완료되면, 다른 서버에도 적용하기 위해 현재의 지정 내역을 파일로 저장한다. 이 때, 저장 내용은 XML 형태로 저장되며, 저장된 파일을 나머지 서버의 동일 폴더로 복사한다. 복사가 완료된 후, 각 서버에 접속하여, 각 서버의 성능관리자에 기존에 XML 형태로 생성해 놓은 성능개체 항목 파일을 이용하여 첫번째 서버와 동일한 형태로 수집해야 할 성능카운터들을 등록해 놓는다.

두 번째 작업은 시작 및 중지 작업으로서, 성능시험이 진행되면, 각 서버에 접속하여, 각 서버의 성능카운터 수집 서비스를 시작 상태로 변경하고, 성능시험이 완료되면, 성능카운터 수집 서비스를 중지시킨다.

세 번째 작업은 수집 단계로서, 여러 대의 서버에 분산되어 있는 성능 카운터 로그 파일들을 한 서버로 복사해서 모아 놓는 작업이다.

네 번째 작업은 각 성능카운터 로그 파일의 내용을 성능관리자 프로그램을 이용하여 분석하는 작업으로서, 성능수집을 해야 하는 서비스와 서버의 숫자가 많을 수록 많은 많은 작업시간이 소요된다.

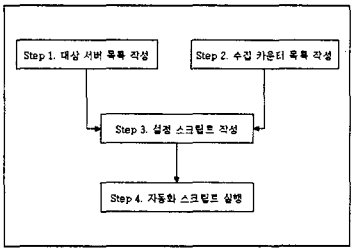
만약, 6대의 서버 환경하에서, 100개의 서비스들에 대한 분석작업을 진행할 경우, 최소 3~4일의 작업 시간이 소요된다. 따라서 20대 이상의 서버군으로 구성된 대규모 환경하에서 위와 같은 수작업 기반의 성능카운터 설정 및 수집, 분석 작업을 거친다면, 최소 2주일 정도가 소요되거나, 작업인력을 3~4배로 증가를 시켜야 하는 문제가 발생한다.

본 논문에서는 이러한 문제점을 해결하기 위하여, 기존의 4가지 형태의 작업과정을 자동화한 과정을 소개하며, 이러한 자동화된 과정을 통하여 작업시간을 3일에서 30분으로 단축시킨 결과를 얻을 수 있었다.

2. 일괄 설정 방법

첫 번째 단계인 일괄설정 단계에서는 4가지의 Step으로 이루어 진다. 즉, 서버들의 목록을 만드는 작업, 설정해야 할 성능카운터 목록을 만드는 작업, 그리고 실행해야 할 스크립트를 만드는 작업이다.

각 작업의 결과는 각각의 파일로서 만들어 지며, 각 과정의 결과를 만드는 작업은 다음과 같다.



[그림1] 일괄 설정 방법

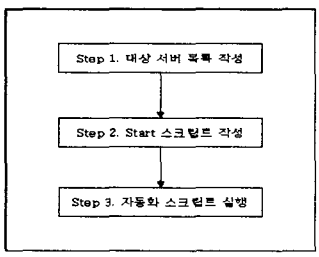
Step 1의 서버 목록을 만들기 위해서는 텍스트 파일을 생성하고 서버 이름을 순차적으로 등록하면 되는데, 이 때, 각 서버명은 각각의 라인에 기입을 한다. 그리고 만들어진 목록 파일을 MyServer.txt로 저장을 한다.

그 다음 작업으로서, Step 2의 작업으로서, 설치할 성능카운터를 텍스트 파일에 등록해 놓는다. 이 때에도 등록해야 할 성능카운터들을 각 라인 별로 등록을 한다. 그리고 만들어진 목록 파일을 MyPerf.txt로 저장을 한다.

이제, Step 3의 작업으로서, 이미 만들어진 서버목록 파일과 카운터 항목 파일을 이용하여 일괄적으로 명령을 실행시키는 스크립트를 작성한다. 이 과정은 윈도우의 logman 유틸리티 명령어를 수행되며, 아래의 명령어를 새로운 텍스트 파일에 저장을 한다. 만들어진 명령어 파일을 Create_Counters.bat로 저장을 한다. Step 4로서 이 명령어를 Command 창에서 실행시키게 되면 지정된 성능카운터 항목들이 미리 설정해 놓은 서버에 모두 일괄적으로 설치되게 된다.

3 .일괄 Start 방법

두 번째 단계인 일괄 Start하는 단계에서는 앞 단계에서 작성된 대상 서버 목록 파일을 재사용하며, 3가지의 Step으로 이루어 진다. 즉, 서버들의 목록을 만드는 작업(재사용), 실행해야 할 스크립트를 만드는 작업, 그리고 스크립트를 실행시키는 작업이다.



[그림2] 일괄 Start 방법

첫 번째 Step 1 은 일괄설정 단계의 Step 1 에서 작성된 대상 서버 목록을 그대로 재사용한다. 두 번째 Step 2 는 logman 명령어를 이용하여 일괄적으로 Start 시키는 명령어 라인을 작성한 후 스크립트로 저장하는 것이다.

```
FOR /F %%i IN (MyServers.txt) DO logman start
My_Perf_Log(You_Should_Change) -s %%i
```

위의 내용을 Start_Counters.bat 파일로 저장한다. 이제 Step 3으로서 이 명령어를 Command 창에서 실행시키게 되면 지정된 성능카운터 항목들이 미리 설정해 놓은 서버에서 모두 일괄적으로 시작되게 된다

4. 일괄 Stop 방법

logman 명령어를 이용하여 일괄적으로 Stop 시키는 명령어 라인을 작성한 후 스크립트로 저장하는 것이다.

```
FOR /F %%i IN (MyServers.txt) DO logman stop
My_Perf_Log(You_Should_Change) -s %%i
```

위의 내용을 Stop_Counters.bat 파일로 저장한다. 이 명령어를 Command 창에서 실행시키게 되면 지정된 성능카운터 항목들이 미리 설정해 놓은 서버에서 모두 일괄적으로 중지되게 된다.

5. 일괄 수집 방법

각 서버들에 저장되어 있는 성능 로그 파일을 한 곳에 수집하여 저장을 하기 위해 서버별 디렉토리를 생성하는 단계이다. 이 Step에서는 아래의 스크립트를 작성 후 실행한다.

```
FOR /F %%i IN (MyServers.txt) DO MKDIR
C:\Temp\PerformanceTest\%%i
```

스크립트를 실행시키고 나면, 실행시킨 서버에 각 서버별 디렉토리가 생성된다. 그리고 각 서버에 저장되어 있는 로그 파일을 특정 서버로 복사하는 스크립트를 작성하는 단계이다. 이 스텝에서는 아래의 스크립트를 작성한다.

```
FOR /F %%i IN (MyServers.txt) DO COPY
\\%%i\C$\Temp\*.csv C:\Temp\PerformanceTest\%%i\*.*.%1
```

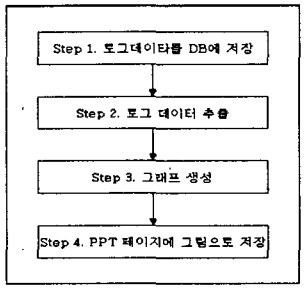
위의 내용을 Copy_Counters.bat 파일로 저장한다. 이제 이 명령어를 Command 창에서 실행시키게 되면 각 서버에 저장되어 있는 성능로그 파일들이 모두 한 곳으로 복사되어 수집되게 된다. 단, 이때 스크립트를 실행시킬 경우, 반복되는 시험을 고려하여, 각 수집단계 마다 구별되는 이름을 붙일 수 있으며, 식별하고자 하는 이름을 명령어 뒤에 붙여서 실행시킨다.

```
Copy_Counters.bat <test_description>
```

6. 성능 리포트 자동 생성방법

제안된 방법 중 마지막 단계인 성능리포트 자동 생성단계에서는 성능시험 결과 분석 시 가장 많이 소용되는 성능 그래프 및 리포트 작업을 자동화 하는 과정이다. 이 방법에서는 엑셀을 이용하여, MS 파워포인트 파일에 관련된 모든 로그 데이터를 그래프로 생성해 놓는다.

그리고 MS 파워포인트로 생성된 결과를 토대로, 메모리 누수와 서비스 중지, 서비스 과다 사용 등에 대한 분석을 진행하게 된다. 성능 리포트 자동 생성 방법은 아래와 같이 4가지 Step으로 구성된다.



[그림3] 성능리포트자동생성단계

6.1. 성능카운터 데이터 취합을 위한 ACCESS 테이블 생성 및 입력

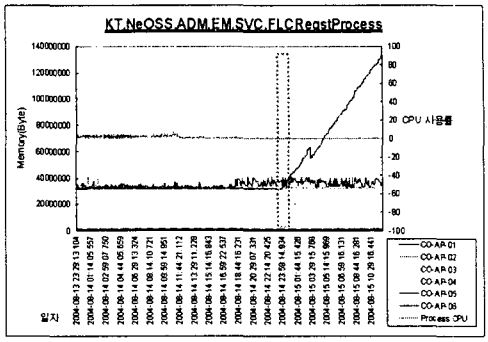
파일로 저장되어 있는 성능 데이터 파일을 DB로 저장하기 위해서는 성능 로그파일을 읽어서 각각 Counter 항목과 데이터 항목을 분리하는 과정을 거친다. 각각의 성능로그 파일을 Parsing 하여 생성된 테이블에 입력을 한다. 이를 수행하기 위해서는 별도의 프로그램을 개발하여 수행하였다

6.2 차트 및 PPT 파일을 생성시키기 위한 엑셀 스크립트

로그 데이터를 DB에 저장한 이후에 진행되는 스텝은 DB에 저장되어 있는 데이터를 읽어서 그래프로 표시하는 작업이다. 본 검토에서는 이러한 작업을 자동적으로 수행하고 별도의 개발 작업을 거치지 않도록 하기 위하여 MS Excel을 추출 및 그래프 표시 도구로서 사용하였다. 또한 작성된 그래프의 양이 많기 때문에 효율적으로 분석할 수 있도록 MS 파워포인트를 통해서 페이지 단위로 그래프를 표시하였다.

6.3 성능수집 결과의 분석

파워포인트의 각 페이지에서 차트로 표시된 성능수집 결과를 이용하여, 여러 가지 형태로 분석될 수가 있고, 성능시험에서의 안정성 결과로 메모리 누수, 서비스 다운, 서비스 CPU 과다사용 등을 분석할 수 있다. 본 논문에서는 메모리 누수의 유형만 분석하였고, [그림4]와 같다.



[그림 4] 메모리 누수 예

7. 결론

20~30대 이상의 서버군으로 구성된 대규모의 서버 환경하에서 성능시험을 진행할 경우에는 성능시험을 위한 환경 설정 및 데이터 수집 등의 작업 또한 상당한 시간을 요하며, 분석 또한 많은 수작업을 필요로 한다.

만약, 6대의 서버 환경하에서, 100개의 서비스들에 대한 분석작업을 진행할 경우, 기존 수작업 방식에서는 최소 3~4일의 작업 시간이 소요된다. 따라서 20대 이상의 서버군으로 구성된 대규모 환경하에서 위와 같은 수작업 기반의 성능카운터 설정 및 수집, 분석 작업을 거친다면, 최소 2주일 정도가 소요되거나, 작업인력을 3~4배로 증가를 시켜야 하는 문제가 발생한다.

본 논문에서는 이러한 문제점을 해결하기 위하여, 기존의 4가지 형태의 수작업과정을 자동화한 과정을 소개하며, 이러한 자동화된 과정을 통하여 작업시간을 3일에서 30분으로 단축시킨 결과를 얻을 수 있었다.

참고 문헌

[1] Edward Whalen, " Microsoft SQL Server 2000 Performance Tuning Technical Reference," Microsoft Press 2002.

[2] Rico Mariani and Brandon Bohling, " Improving .NET Application Performance and Scalability," patterns & practices, 2004.

[3] Mohammad Al-sabt, " Testing Software Patterns," patterns & practices, 2003.

[4] Microsoft, " Caching Architecture Guide for .NET Framework Applications, " patterns & practices, 2003.

[5] Mark Boulter, " Smart Client Architecture and Design Guide," patterns & practices, 2004.