

클러스터 시스템의 성능 레포트 툴의 아키텍처 모델링

김 기* · 최은미*

Architecture Modeling of a Performance Report Tool for a Cluster System

Ki Kim · Eunmi Choi

Abstract

In order to manage a cluster system that consists of a number servers, management aspects such as configuration management, fault management, performance management, and user management should be considered. Especially, it is necessary to monitor performances for performance and fault management. An agent in each server collects performance counters, status changes, and events occurred in normal or abnormal states. The data collected are delivered to a collector server and processed in a report tool for performance analysis and management decision in the cluster system point of view, by detecting fault state and tracing out resource usage, service request and response, and states until failed. In this paper, we propose an architecture modeling of a performance report tool for proactive cluster system management. Some results on a cluster system are presented.

Key Words: performance report tool, cluster system, cluster system management, modeling

1. 서론

분산 시스템 상에서, 동종의 어플리케이션 서비스를 갖는 여러 서버를 묶어서 하나의 단일 서비스를 제공하는 클러스터 시스템이 인터넷의 서버단에 안전성과 24시간 365일 가용성을 제공하는 인프라 구조로 자리잡게 되었다.[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]

특히 성능 관리 측면과 이상상황 관리를 위해서 성능 모니터링이 필요하다. 성능 모니터링은 서버관리의 기본적인 부분이며 이상상황 발생의 원인과 추세 파악, 성능 최적화, 서버추가 혹은 자원확장의 판단근거를 제공한다.[2,4,6]

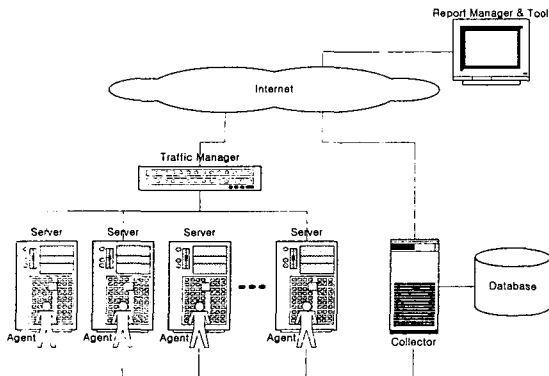
본 논문에서는 클러스터 시스템의 성능 모니터링을 통하여 보고서를 생성하는 레포트 툴의 아키텍처를 domain별로 나누어 클래스 구조를 설명하도록 하겠다.

2장에서는 클러스터 시스템의 성능 레포트 툴의 개요를 설명하며, 3장에서는 각 domain별 구

* 한동대학교 전산전자공학부

조도를 설명한다. 4장에서는 레포트 툴을 이용한 결과를 설명하며 5장에서 본 논문의 결론을 내린다.

2. 클러스터 시스템의 성능 레포트 툴의 개요



〈그림 1〉 클러스터 시스템의 성능 레포트를 위한 구조도

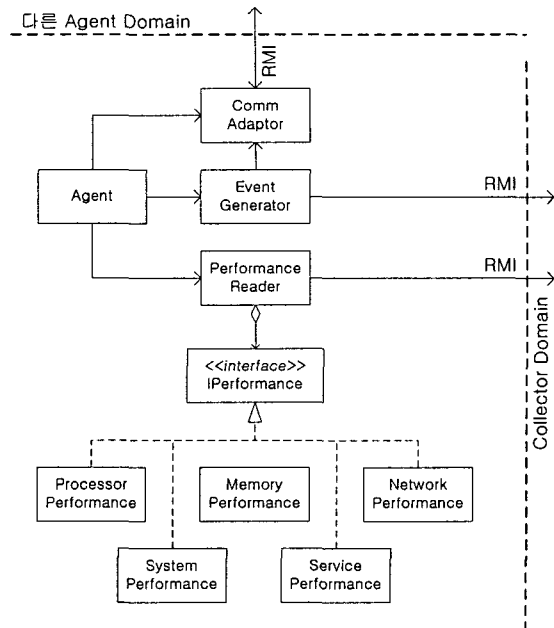
클러스터 시스템은 인터넷에서 들어오는 서비스 요청을 앞단의 Traffic Manager를 통하여 각 서버들에게 분산시킴으로 서비스를 제공한다. 클러스터 시스템을 위한 성능 모니터링은 서비스를 제공하는 서버노드 domain과 각 서버 노드에 설치된 agent로부터 정보를 수집하는 Collector domain, 수집된 정보를 저장하는 데이터베이스 domain, 인터넷을 통하여 수집을 관리하고 보고서를 생성하는 domain으로 나누어진다.

3. Domain 별 구조도

3.1 서버노드 단에서의 모니터링

성능 모니터링은 각 서버 노드에 설치된 agent를 통하여 이루어지며 다음의 클래스들로 구성되어서 일을 처리하게 된다.

Agent는 클러스터 시스템의 각 노드에 설치되어 동작하며, 능동적으로 운영 중에 발생하는 이벤트와 이상상황을 EventGenerator에게 전달한다. 또한 Agent는 주기적으로 성능수치를 노드



〈그림 2〉 서버노드 단에서의 모니터링 관련 클래스 구조

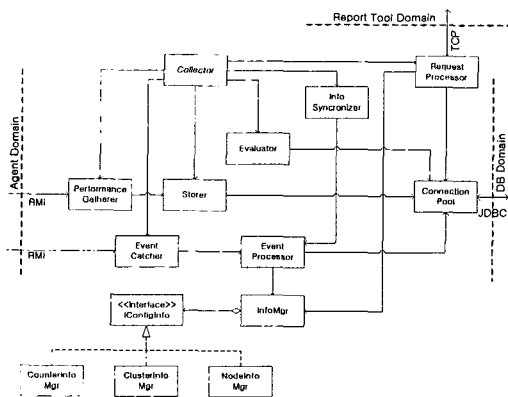
로부터 읽어서 PerformanceReader에게 전달한다. EventGenerator는 상태변경, 이상상황과 같은 운영에 필요한 사건에 대해 이벤트를 발생함으로써 RMI를 통하여 Collector에게 전달한다. CommunicationAdaptor는 노드간의 정보와 이벤트를 주고 받는다. 한편, PerformanceReader는 각 성능수치를 담당하는 성능 객체를 IPerformance를 통하여 호출하여 부분별 성능수치를 수집하여 RMI를 통하여 Collector에게 전달한다.

ProcessorPerformance는 프로세서에 관련된 Processor Load (프로세서가 사용된 시간비율), User Time (사용자 모드에서 소비한 프로세스 시간비율), Interrupt/sec (프로세서가 처리한 하드웨어 인터럽트의 초당 발생수) 등의 성능수치를 수집한다. MemoryPerformance는 메모리에 관련된 Available Memory (실행되는 프로세스에 사용가능한 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 공간

을 비우기 위해 페이지를 디스크에 쓴 비율) 등의 성능 수치를 수집한다. Network Performance는 네트워크에 관련된 TCP Connection Established (TCP 연결 개수), TCP Connection Failure (TCP 연결에 실패한 개수), Bytes Received/sec (네트워크 어댑터에서 받는 바이트의 비율), Bytes Sent/sec (보내는 바이트의 비율) 등의 성능수치를 수집한다. SystemPerformance는 시스템에 관련된 Processor Queue Length (프로세서 대기열의 쓰레드 수), Context Switches/sec (모든 프로세서가 한 쓰레드에서 다른 쓰레드로 전환한 전체 횟수), Processes (프로세스의 수), Threads (쓰레드의 수)와 같은 성능 수치를 수집한다. ServicePerformance는 어플리케이션 서비스 관련된 성능수치를 수집하며 어플리케이션 도메인에 따라 수집하는 성능수치는 달라질 수 있다. 웹서비스의 경우 Request/sec (IIS 웹서비스에서 초당 실행한 요청수), Bytes Recived/sec (웹서비스가 받은 데이터 바이트의 비율), Bytes Sent/sec (보낸 데이터 바이트의 비율) 등의 성능수치를 수집한다.[3]

3.2 데이터 수집

Collection부분에서 하는 일은 각 서버노드의 agent로부터 이벤트와 성능수치를 수집하고, 수집 내용을 데이터베이스에 저장하여 보고서 작성에 쓰이도록 한다.



<그림 3> 데이터 수집관련 클래스 구조

Collector는 데이터를 수집하기 위한 Collection 부분에서 최초로 초기화되어 모듈들을 생성하고 관리하며, RequestProcessor는 사용자에게 보고서 생성 요청과 수집관리에 관한 요청을 받아 처리할 수 있는 TCP통신을 제공하여 해당하는 정보 제공과 명령을 처리한다.

PerformanceGatherer는 RMI를 통하여 각 서버의 agent들로부터 성능수치를 수집하며, EventCather는 각 서버 agent에서 발생한 이벤트를 수집하여 EventProcessor에게 넘겨주고 EventProcessor는 수집 내용을 데이터베이스에 저장하고 클러스터와 노드에 관련된 변경된 정보를 InfoMgr에게 보낸다.

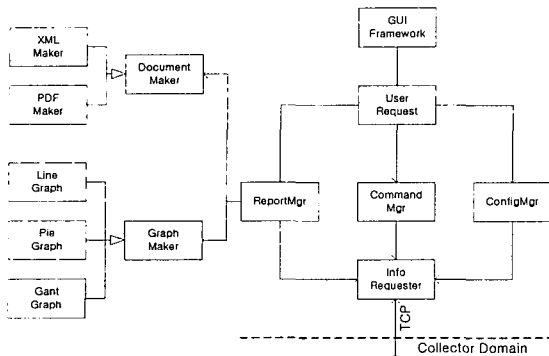
Storer는 가공되지 않은 수집된 그대로의 성능수치를 ConnectionPool을 통하여 데이터베이스에 저장하도록 하며, Evaluator는 가공되지 않은 성능수치를 분, 시간, 일 단위의 시간에 따라서 분석데이터로 만들어 별도의 데이터베이스 스키마에 저장한다. InfoSynchronizer는 실제로 운용되고 있는 클러스터 환경과 Collector에 저장되어 있는 환경을 비교하고, 수집하고 있는 시스템의 정지, 네트워크 문제로 인하여 발생할 수 있는 불일치를 주기적으로 확인하여 정보를 동기화한다.

InfoMgr는 EventProcessor에게 받은 정보를 바탕으로, 가지고 있는 클러스터, 노드 등의 환경 정보를 갱신하고 관리한다. CounterInfoMgr은 성능수치의 목록과 단위 등의 정보를 관리하며, ClusterInfoMgr는 클러스터의 목록과 서비스 종류, 클러스터에 속한 노드들의 목록 등의 정보를 관리한다. NodeInfoMgr은 노드의 목록과 시스템 자원, 운영체제 등의 정보를 관리한다.

ConnectionPool은 자료처리 속도향상을 위하여 데이터베이스에 연결하는 connection들을 pool로 만들어 제공한다.

3.3 수집관리 및 보고서 생성

사용자에게 노드별로 수집할 성능수치와 수집 주기를 설정하는 기능을 제공하며 분석을 위한 보고서를 만든다.



〈그림 4〉 수집관리 및 보고서 생성 관련 클래스 구조

GUI Framework는 사용자에게 보고서 생성과 수집관리를 할 수 있도록 인터페이스를 제공하며, UserRequest는 사용자의 request와 option을 분석하여 command와 parameter를 만들고 유지하여 사용자의 request에 따라서 ReportMgr, CommandMgr, ConfigMgr에게 일을 전달하여 준다.

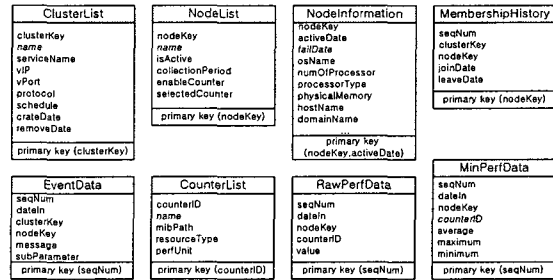
CommandMgr은 사용자에게 수집되기 원하는 성능수치 선택과 수집주기를 설정할 수 있는 command를 만들어서 InfoRequester에게 전달한다. ConfigMgr은 사용자에게 클러스터, 노드목록 등 클러스터 시스템의 기본적인 정보와 설정을 제공하며 현재의 클러스터 시스템의 전체 configuration을 저장하며, ReportMgr은 사용자에게 받은 request와 옵션을 토대로 데이터를 요구하며 보고서를 생성하도록 DocumentMaker와 GraphMaker에게 정보를 전달한다.

DocumentMaker는 보고서를 문서로 생성하는 기능을 제공하며, XmlMaker는 보고서를 XML format, PdfMaker는 PDF format의 문서로 생성한다.

GraphMaker는 보고서에 필요한 그래프를 생성하며, LineGraph는 라인 그래프, PieGraph는 파

이 그래프, GantGraph는 간트 그래프를 제공한다.

3.4 데이터베이스 스키마



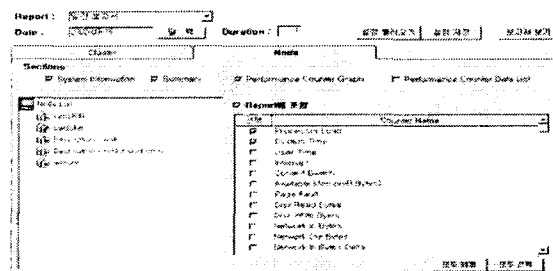
〈그림 5〉 데이터베이스 스키마

ClusterList와 NodeList는 클러스터, 노드의 목록과 정보를 저장하며, CounterList는 성능수치의 목록과 정보를 저장한다.

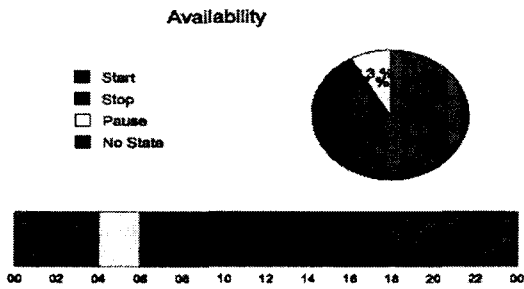
Raw PerfData는 실시간 성능수치를 저장하며, MinPerfData, HourPerfData, DayPerfData는 성능수치를 분, 시간, 일 간격으로 평균값, 최대값, 최소값을 계산하여 저장한다. EventData는 서버에서 발생한 이상상황, 상태변화의 이벤트를 저장하며, 클러스터와 노드의 소속 정보는 MembershipHistory가 저장한다.

4. 레포트 틀을 이용한 결과

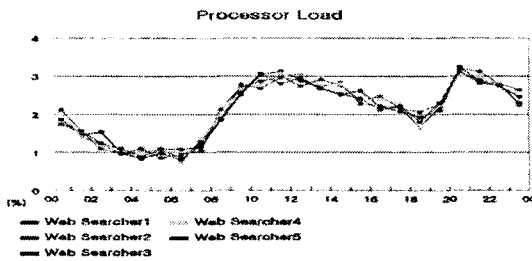
다음 그림들은, 클러스터 시스템의 레포트 틀에서 제공하는 보고서생성에 필요한 GUI와 성능 평가 결과로 클러스터 시스템의 가용성과 자원사용 그래프를 보이고 있다.



〈그림 5〉 보고서 생성을 위한 GUI



<그림 6> 성능보고서의 가용성 그래프 예제



<그림 7> 성능 보고서의 자원사용그래프 예제

5. 결론

클러스터 시스템의 효과적인 성능 분석을 위한 수집과 관리 및 레포팅의 아키텍처 모델링을 제안하였다. 서버 단에서의 모니터링, 수집, 데이터베이스, 정보 관리 및 보고서 생성으로 domain을 나누어 클래스 구조를 소개하였으며, Java로

구현한 레포트툴을 이용한 클러스터 시스템의 성능 결과를 보였다.

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 <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechol/iis/reskit/iis50rg/iischp5.asp>
- [4] 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
- [5] Mark Grand, Patterns in Java, Volume1, A Catalog of Reusable Design Patterns Illustrated with UML, John Wiley & Sons, INC., 1998.
- [6] Henz-Gerd, Sebastian Abeck, Bernhard Neumair, Integrated Management of Network Systems, Morgan Kaufmann, 1998.