클러스터 시스템을 위한 단일 모니터링 에이전트에 대한 연구

박유찬, 홍태영
한국과학기술정보연구원 슈퍼컴퓨팅센터
e-mail:{yuchan, tyhong}@kisti.re.kr

A Study on Single Monitoring Agents for Cluster System

Yuchan Park, Taeyoung Hong
Supercomputing Center, Korea Institute of Science and Technology Information

요 약

고성능 컴퓨팅이 요구됨에 따라 유연성과 효율성을 가진 Beowulf형 클러스터 시스템의 활용이 증가되고 있다. Beowulf 형 클러스터는 각각이 독립적인 운영체제를 가진 노드로 구성되어 있기 때문에 관리 및 모니터링에 많은 어려움이 있다. 모니터링 프로그램은 각각의 노드를 관리함과 더불어 세부적인 시스템의 정보를 제공하도록 연구 개발되고 있다. 특히 단일 모니터링 에이전트는 좀 더 많은 정보의 요구로 인하여 시스템 자원을 적게 소모하면서 많은 데이터를 추출할 수 있도록 구성되어 있다. 본 논문에서는 단일 노드에 설치되어 직접 데이터를 수집하고 전송하는 단일 모니터링 에이전트에 대한 비교 및 분석을 수행하였다.

1. 서론

클러스터 시스템은 일반적으로 수십 대에서 수백 대 이상의 노드로 구성되어 있으며, 각각의 노드는 독립적인 운영체제로 구동된다[1]. 이러한 클러스터 시스템의 특성으로 인하여 단일 프로세서 시스템에 비하여 모니터링에 대한 어려움이 많다. 클러스터 시스템을 효율적으로 모니터링하기 위해서 supermon[2], ganglia[3], clumon[4], Tivoli[5], CSM[6] 등의 클러스터 모니터링 도구에 대한 연구 및 개발이 활발히 진행되고 있다. 이러한 모니터링 도구는 단일 모니터링 에이전트, 각 노드에 설치되어 있는 에이전트를 제어하는 메타 에이전트, 그리고 수집된 데이터를 출력하는 뷰어 프로그램으로 구성되어 있다. 본 논문에서는 이러한 모니터링 도구의 구성 요소 중에서 각각의 노드에 설치되어 시스템의 정보를 추출하는 기능을 담당하는 단일 모니터링 에이전트를 비교 분석하였다. 본 논문은 다음과 같이 구성되어 있다. 2장에서는 단일 모니터링 에이전트가 가져야 할 특징에 대해 간략히 기술하였으며, 3장에서는 대표적인 모니터링 도구에 포함된 단일 모니터링 에이전트를 비교 분석하였다. 마지막으로 4장에서는 결론 및 향후 연구 방향에 대해서 기술하였다.

2. 모니터링 에이전트

2.1 모니터링 에이전트의 특징

모니터링 에이전트의 가장 큰 기능은 시스템의 정상적인 동작 유무와 관리자에게 제공할 시스템의 정보 및 현재 동작중인 프로그램의 정보를 추출하는 것이다. 클러스터 시스템의 특성상 각각의 노드는 독립적인 운영체제로 구동되며, 각 노드에 설치되는 단일 모니터링 에이전트는 하나의 독립적인 프로그램으로 동작한다. 이렇듯 독립적인 프로그램으로 동작하기 때문에 시스템의 정보를 추출하는 과정에 CPU, 메모리자원, I/O 그리고 네트워크 트래픽이 발생하게 된다. 따라서 효율적인 에이전트의 요건으로는 시스템에 최소한의 부하로 최대한의 정보를 추출할 수 있어야 한다.

2.2 모니터링 에이전트의 동작 방식

대부분의 클러스터 시스템 모니터링에서는 단일 모니터링 에이전트를 데몬 형태로 구현하고 있다. 이러한 데몬 프로그램은 자체적으로 설정된 내역 혹은 중앙 모니터링에서 데이터 수집에 대한 요구가 발생하면 시스템에서 데이터 추출 작업을 수행한다. 요구하는 정보에 따라 약간의 차이를 보여주지만 추출하는 시스템 정보의 대부분은 /proc 파일 시스템을 이용한다. 이러한 모니터링 도구가 /proc 파일 시스템을 사용하는 이유는 메모리상에 존재하는 가상 파일 시스템이면서 커널의 정보 및 메모리, cpu 사용량, 각 프로세서의 정보를 제공하며, 데이터 수집에서의 안정성과 확장성을 보장해주기

때문이다. 이러한 모니터링 에이전트의 일반적인 동작 방식을 그림 1과 같다.

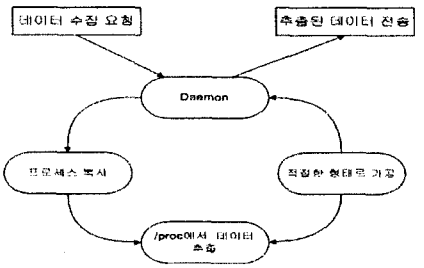

그림 1 모니터링 에이전트의 일반적 동작 방식

3. 단일 시스템 모니터링 에이전트 비교 및 분석

클러스터 시스템을 위한 모니터링 도구들은 각각 고유한 특징을 가지고 있다. 여기에서는 2장에서 언급한 것과 같은 기본 동작과 그 기능을 분석하고, 실제 모니터링 에이전트가 사용하는 시스템 자원을 비교 분석하였다.

3.1 gmond

gmond는 ganglia에서 사용하는 단일 시스템 모니터링 에이전트이다[7]. gmond는 multi-thread를 이용하는 에이전트로써 그림 2와 같이 크게 4부분으로 구성되어 있다. metric scheduler thread는 시스템에서 gmond가 올바르게 동작하는지 상태를 주기적으로 점검하며 어떠한 변경사항이 감지되었거나 일정 시간이 지나면 시스템의 상태를 멀티캐스팅으로 알리게 된다. multicast listen thread는 멀티 캐스팅으로 전송된 모든 데이터와 자신의 데이터를 fast in-memory cluster hash로 저장한다. 따라서 hash에는 모든 gmond와 gmetric에서 전송된 모든 데이터가 존재하게 된다. XML output thread는 요청된 데이터를 처리하는 부분이다. XML output thread에서는 요청된 데이터를 hash에서 검색하여 XML 형태로 전송한다[8].

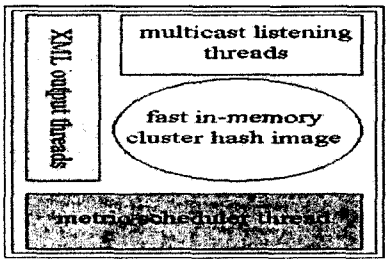

그림 2 gmond 구조

3.2 pcp

pcp(performance co-profile)는 SGI에서 IRIX 시스템을 이용하는 모니터 및 관리 도구로 사용하기 위해 만든 상용 소프트웨어였으나 2002년도에 공개 소프트웨어로 전환되었다[9]. clumon에서는 모니터링 에이전트로써 pcp를 이용한다. pcp는 그림 3과 같다.

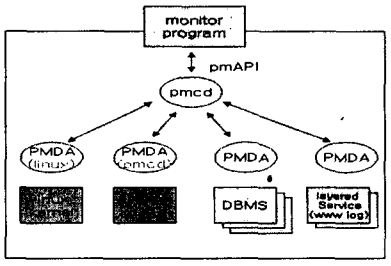

그림 3 pcp 구조

Monitoring program이 pmAPI를 통해 metric value를 요청하면, 시스템상의 pmcd(performance metric collector daemon) 가 이를 대신하여 PMDA(performance metric domain agent) 에 요청을 한다. PMDA는 pmcd의 요청에 대해 시스템상의 필요한 performance metrics, instance domains 그리고 instantiated values를 얻은 후 이를 pmcd에 응답한다. pmda_linux의 경우 대부분의 필요한 metric의 value를 /proc을 통해 얻는다.

3.3 mon

mon은 supermon에서 사용하는 단일 시스템 모니터링 에이전트이다[10]. mon 역시 /proc에서 필요한 데이터를 추출하지만 기본적으로 supermon의 kernel 모듈에서 정보를 추출하도록 되어 있다. mon의 구성 도는 그림 4와 같다. mon에는 동적으로 사용할 수 있는 커널 모듈(supermon)을 사용한다. 커널 모듈은 /proc/sys에 데이터를 삽입하여, 클라이언트가 데이터 요구 시 /proc/sys를 읽어 전송한다. 이러한 데이터는 s-expressions 형식을 사용한다.

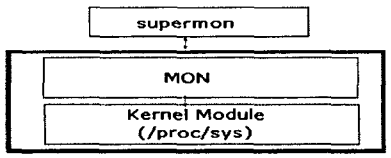

그림 4 mon 구조

3.4 비교 및 분석

2장에서 언급했듯이 단일 시스템의 모니터링 에이전트는 독립적인 프로그램이기 때문에 CPU, Memory 자원 그리고 네트워크 자원을 소모한다. 이 절에서는 각각의

모니터링 에이전트가 각 노드에 미치는 영향을 분석하기 위해 CPU와 Memory 자원 소모율만을 비교하였다.

3.4.1 테스트 환경
각각의 단일 시스템 모니터링 에이전트는 표 1 의 시스템에서 테스트를 수행하였다.

표 1 테스트 환경

CPU	메모리	OS	커널	모니터링 도구		
				ganglia	clumon	supermon
2.4GHz	512MB	CentOS 4.0	2.4.31	3.0.0	2.4.0	1.4

3.4.2 테스트 프로그램
각 모니터링 에이전트의 성능을 측정하기 위한 테스트 프로그램은 자체 제작하였다. 측정할 단일 모니터링 에이전트를 입력하면, 이 프로그램은 /proc에서 단일 모니터링 에이전트의 프로세스 ID를 검색한다. 각 모니터링 에이전트의 프로세스에 대해서만 정보를 취합 분석하여 출력하도록 하였다.

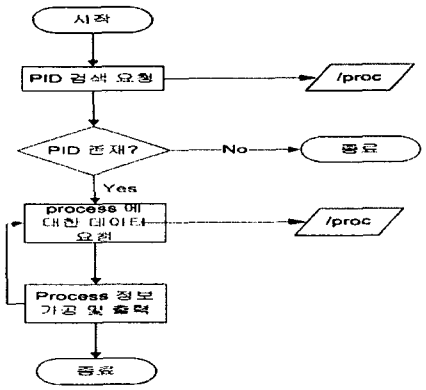

그림 5 테스트 프로그램 동작

3.4.3 비교 분석
테스트 프로그램을 이용하여 각 단일 시스템 모니터링 에이전트의 자원 소모율을 확인하였다. 각각의 에이전트는 모두 안정적인 형태로 동작하였으나 mon의 경우에는 초기 변동 비율이 매우 컸지만 시간이 지날수록 그 변동 비율이 적어지는 것을 볼 수 있다. 이러한 이유는 mon에서는 타 에이전트와는 달리 자체적으로 데이터를 전송하기 때문이다. 메모리 사용률은 세 개의 모니터링 에이전트가 정보 수집의 요청에 따라 큰 변동이 없음을 알 수 있다.

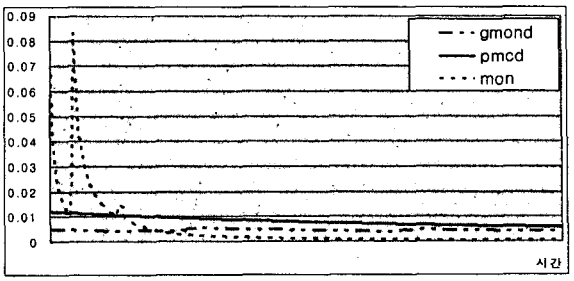

그림 6 CPU 사용률

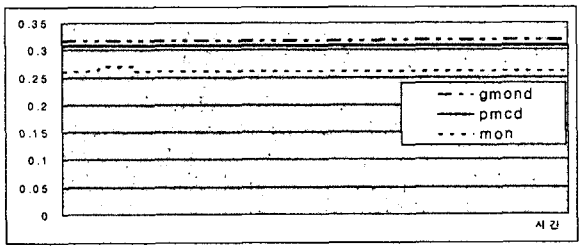

그림 7 메모리 사용률

4. 결론 및 향후 연구 방향
클러스터 시스템이 점점 더 대형화되고, 구성 형태가 다양해짐에 따라 시스템의 모니터링에 대한 연구 및 개발이 빠르게 이뤄지고 있다. 본 논문에서는 오픈소스 기반의 단일 노드 시스템 모니터링 에이전트에 대한 소프트웨어적인 분석을 수행하였다. 향후에는 본 분석을 통하여 다양한 기능을 수행할 수 있는 모니터링 에이전트에 대한 연구를 수행할 계획이다.

참고문헌
[1] 조혜영, 홍태영, 홍정우, 클러스터 시스템 관리 도구에 관한 연구, 한국정보처리학회, 제 11권, 2호, 1043, 2004
[2] Matthew J. Sottile Ronald G. Minnich, Supermon: A high-speed cluster monitoring system, Cluster '02, 2002
[3] ganglia homepage, http://ganglia.sourceforge.net/
[4] clumon homepage, http://clumon.ncsa.uiuc.edu
[5] tivoli homepage, http://www-306.ibm.com/software/tivoli/
[6] csm homepage, http://www-03.ibm.com/servers/eserver/clusters/software/csm.html
[7]Matthew L. Massie, Brent N. Chun, and David E. Culler. Parallel Computing, Vol. 30, Issue 7, July 2004.
[8] Performance Co-Pilot Open Source Release, http://oss.sgi.com/projects/pcp/
[9] supermon homepage, http://supermon.sf.net