소규모 SMP 클러스터 시스템 모니터링 개발

Development of Monitoring Tool for Small SMP Cluster System

성진우, 이영주, 최윤근, 박찬열
한국과학기술정보연구원 슈퍼컴퓨팅센터

JinWoo Sung, YoungJoo Lee, YounKeun Choi, ChanYeol Park
Supercomputing center, KISTI

요약

클러스터 시스템을 관리하기 위하여 모니터링 툴(S/W)이 필요하지만, 소규모 클러스터 시스템에 적합하고 관리자가 필요로 하는 기능을 갖춘 모니터링 툴을 결정하는 것은 쉽지가 않다. 본 문서는 인피니밴드 네트워크 스위치를 사용한 소규모 SMP 클러스터 시스템을 위하여 개발한 모니터링 툴의 설계와 구현 내용을 기술하였다. 모니터링 툴의 기능은 계산노드, 인피니밴드 스위치 그리고 작업관리 스케쥴러(PBS)의 작업을 모니터링한다.

Abstract

System manager needs monitoring tool(S/W) to manage cluster system. But, it is difficult to decide suitable monitoring tool for small SMP cluster system. This document described design of monitoring tool(mon) and development. Mon is monitoring tool for small SMP cluster system using InfiniBand network switch. Function of this tool is monitoring such as computing node(7 node), Infiniband network switch and monitoring of PBS job.

I. 서 론

컴퓨터 기술이 발달하면서 고성능 마이크로프로세서, 그리고 높은 대역폭과 낮은 지연 시간을 가지는 네트워크를 저렴하고 손쉽게 구할 수 있게 되었다. 컴퓨터 성능이 향상되고 가격이 하락하면서 PC 또는 워크스테이션들을 네트워크로 연결하여 대용량 컴퓨터의 성능을 가질 수 있는 클러스터 시스템에 많은 관심이 집중되고 있다. 클러스터 시스템은 두 대 이상의 컴퓨터를 서로 연결하여 확대된 단일 시스템 환경을 제공한다. 클러스터를 구성하는 컴퓨터 노드의 수는 수십 대에서 수백 대 이상으로 점점 규모가 커지고 있으며, 클러스터의 규모가 커질수록 관리의 어려움 또한 비례하여 증가한다[1]. 이러한 클러스터 시스템의 특성으로 단일 프로세서 시스템에 비하여 모니터링에 대한 어려움이 많다.

클러스터 시스템을 효율적으로 모니터링하기 위해서 supermon, ganglia, clumon, Tivoli, CSM 등의 클러스터 모니터링 도구들이 이용되고 있으며, 이에 대한 연구 및 개발이 활발히 진행되고 있다[2]. 그러나 이러한 모니터링 도구들은 job과 queue에 대한 모니터링을 완벽히 지원하지 못하거나 개발 초기단계라 안정성 및 확장성에 대한 검증이 되지 않은 상태이다.

KISTI는 2004년에 Myrinet 스위치 기반의 256노드(512CPU)규모의 클러스터 시스템을 위하여 모니터링 툴을 개발하였으며[3], 금년에는 InfiniBand 스위치 구성의 소규모(112CPU/7노드) SMP 클러스터 시스템의 모니터링을 위한 툴을 개발하였다.

본 논문에서는 금년에 개발한 소규모 SMP클러스터 시스템을 위한 툴의 설계 및 개발 내용을 설명한다.

논문의 구성은, 2장에서 관련 이론에 대해 간략히 기술하였으며, 3장에서 모니터링 툴 개발에 대하여 기술하고, 4장에서는 결론에 대하여 기술하였다.

II. 관련 이론

1. 모니터링 요소[4]

모니터링 요소(표 1)는 노드 정보와 서비스 상태 그리고 클러스터 정보 이렇게 3 가지 그룹으로 나눌 수 있다. 노드정보에는 현재 동작중인 각 노드별의 CPU, 메모리, 네트워크 등 각각의 자원에 대한 정보들이 포함되어 있다. 이 정보들을 이용하여 개별적인 노드의 상태를 모니터링 할 수 있다. 서비스 상태에는 각 노드에서 제공되는 HTTP, TELNET 등의 다양한 인터넷 서비스의 동작 및 이상유무를 알아낼 수 있는 정보가 포함되어 있다. 마지막으로 클러스터 정보그룹이 있는데 이 정보는 클러스터 전체가 적절히 유지되고 있는지를 감시하기 때문에 이는 어떻게 보면 클러스터 모니터링 요소들 중에서 가장 중요한 요인들이라고 할 수 있다. 이 클러스터 정보에는 노드 전체에서의 분산 처리 정도를 알 수 있는 로드(Load) 분

산율과 노드의 생존유무 그리고 노드 전체 자원의 소모율에 대한 통계자료들이 포함되어 있다.

이러한 클러스터 모니터링 통계 정보들을 이용하여 클러스터 서버 관리자는 클러스터의 이상유무를 빠르게 파악하여 유지보수 할 수 있다.

[표 1] 클러스터 성능 요소

그룹	성능 메트릭
노드 정보	• CPU 사용량
	• 메모리 사용량
	• 네트워크 사용량
	• 현재의 세션(사용자)수
	• 프로세스 수/크기
	• 스레드 수
서비스 상태	• HTTP
	• SMTP
	• TELNET 등
클러스터 정보	• 각 노드의 로드 분산률
	• 각 노드의 생존 유뮤
	• 각 노드의 자원 사용률

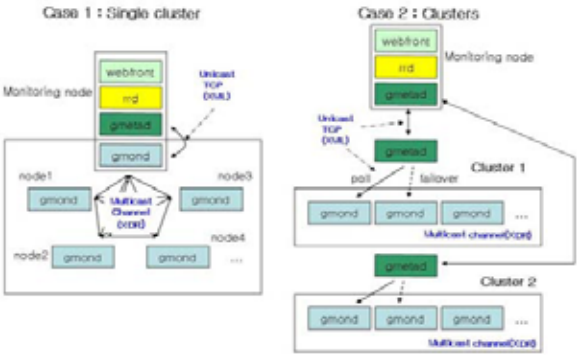
2. 모니터링 도구[5]

1.1 Ganglia

Ganglia는 UC Berkeley의 Millenium 프로젝트의 일환으로 개발하기 시작한 소프트웨어 중 하나로 클러스터, 그리드 등의 HPC 시스템의 모니터링 도구이다. SDSC(San Diego Supercomputer center)의 Rocks팀을 비롯한 여러 그룹에서 이 project에 참여하고 있다.

Ganglia는 분산된 서비스 구조와 multicast를 통해 안정적인 scalability를 보장하며, 기본적으로 그리드 환경에서 사용할 수 있는 구조를 가지고 있다. 그리고 RRD를 database로 사용하여, 시스템의 상태와 성능에 대한 정보를 주기적으로 저장하고, 웹을 통한 편리한 관리 기능을 지원한다. 그러나 다음과 같은 몇 가지 보완되어야 할 점 들을 지적할 수 있다. 첫째 사용자가 모니터링하고자 하는 노드 정보를 추가하는 기능이 매우 제한적이다. 둘째 클러스터 시스템에서 가장 중요한 모니터링 대상인 job과 queue 정보에 대한 모니터링을 아직 완벽히 지원하지 못하고 있다. 셋째, 각 노드의 프로세스 트리 정보를 Clumon 같은 다른 모니터링 도구처럼 직접 웹상에서 보여주기 어려운 구조를 가지고 있다.

[그림 1]에 Ganglia의 프로그램 구성도를 자세히 나타내었다. Ganglia는 크게 gmond와 gmetad의 두 가지의 데몬으로 구성된다. Gmond는 멀티 쓰레드 데몬으로 각 노드에 실행되며, [그림 2]와 같이 monitor thread, Listening thread, XML Export thread로 구성된다.



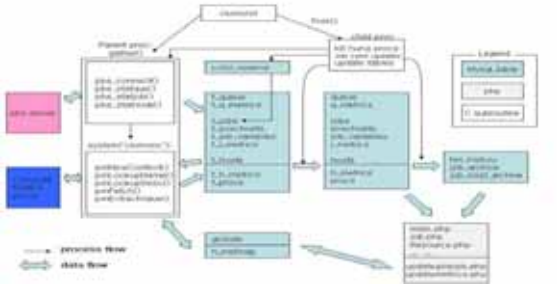
▶▶ 그림 1. Ganglia 프로그램 구성도

Gmond는 monitor thread를 통해 노드 자신에 대한 metric 정보를 수집하고, 이 정보를 XDR(eXternal Data Representation) 데이터 포맷으로 multicast 채널로 보낸다. metric 정보에는 호스트네임, IP 등 시스템의 기본 정보와 CPU, 메모리, 디스크에 관련된 시스템의 현재 성능 등 일반적으로 시스템의 /proc에서 얻을 수 있는 정보를 포함한다. Ganglia는 사용자 인터페이스인 gmetric을 이용하여, 기본적인 metric 정보외의 다른 metric 정보의 추가 기능도 제공한다.

▶▶ 그림 2. gmond 구성도

1.2 Clumon

Clumon은 공개 클러스터 시스템 모니터링 도구로서, NCSA(The National Center for Supercomputing Applications)에 의해 독자적으로 개발되었다. Clumon은 각 노드 상에서 metric을 수집하기 위한 프로그램으로 SGI에서 만든 Performance Co-Pilot(PCP)을 사용한다. PCP는 원래 SGI의 IRIX에서 모니터 및 관리 툴로 사용하기 위해서 만든 상용 소프트웨어였으나 2002년 2월에 Open source로 공개되었다.



▶▶ 그림 3. Clumon의 프로그램 구성도

Clumon은 클러스터 시스템의 성능에 대한 정보를 얻기 위해 PCP 및 PBS[6] 등을 활용하기 때문에 단순한 구조에도 불구하고 탁월한 장점을 많이 가지고 있다. 첫째, PBS가 제공하는 클러스터 시스템의 job, queue, node정보를 제공하며 PCP에서 기본적으로 지원하는 500여개가 넘는 metric pool 내에서 사용자가 모니터링하고자 하는 metric을 쉽게 추가 할 수 있다. 둘째, 각 노드의 process tree를 웹을 통해 쉽게 확인 할 수 있으며 이를 통해 각 노드의 현재 상태를 쉽게 파악 할 수 있다. 그러나 Clumon은 아직 개발 초기단계이라 안정성 및 확장성에 대한 검증이 되지 않은 상태이며 설계 자체부터 중형 이하의 리눅스 클러스터 환경에 최적화 되어 있다는 점에서 한계를 갖는다. 또한 Clumon은 Ganglia처럼 그리드를 지원하지 않으며 PBS 이외의 다른 작업 스케줄러와는 연동 될 수 없는 단점을 가지고 있다. [그림 3]에 Clumon의 프로그램 구성도를 자세히 나타내었다.

Ⅲ. SMP 클러스터 모니터링 툴 개발

클러스터 시스템을 위한 모니터링 도구를 개발하고자 할 때 표 1과 같은 사항이 고려되었다.

[표 1] 개발 요구사항

a. 각 노드의 생사(生死) 정보가 중요하다.
b. 장애발생시 알림기능이 있어야 한다.
c. 사용자 작업에 대한 정보가 나타나야 한다.
d. 정보가 자동으로 갱신되어야 한다.
e. 이식성과 사용이 쉬워야 한다.

개발 환경은 아래와 같다.

-H/W : 7노드(16CPU/node)클러스터 시스템

-O/S : Linux 2.4.20(RedHat 7.3)

-언어 : shell script

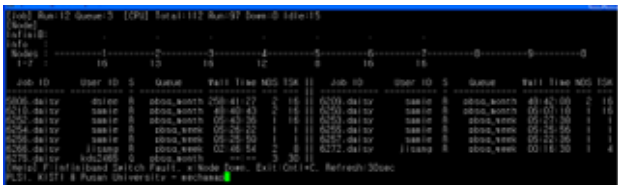
-필요 프로그램 : PBS(Portable Batch System)

그리고, 노드가 정상상태인지 혹은 문제가 있는지는 노드의 생사와 계산용 네트워크인 InfiniBand 스위치의 상태를 점검하여 결정하였다. 노드의 생사를 확인하는 방법은 ping 명령어를 이용하였으며, 다음과 같으며, 점검 결과는 파일로 보관하도록 하였다.

[표 2] mon 소스 일부

```
NODELIST="node1 node2 node3 node4 node5 node6 node7"
for i in `echo $NODELIST`
do
  if ping $i -c 1 -w 1 > /dev/null 2>&1
  then
      echo > /dev/null
  else
      echo "$i x" >> $Giga_State_File
  fi
done
```

InfiniBand 스위치의 상태는 스위치 관리 명령어인 vstat 명령어를 이용하였다. 이상의 2가지의 점검 결과 파일을 이용하여 모니터링 화면을 구성한다. 완성된 도구의 실행 모습은 [그림 4]와 같다. 프로그램을 실행하면 크게 2종류의 정보를 보여주며, 노드에 대한 정보(그림 4에서 상단 부분)와 사용자 작업(그림 2의 하단 부분)에 대한 부분으로 구분된다. 노드에 대한 정보는 표 3과 같다. 각 노드에 장착되어 있는 InfiniBand 스위치에 어떠한 이상이 발생하였다면 이 프로그램에서는 붉은 글씨의 'F'가 표시가 된다.(그림 5의 'F') 관리자들은 모니터링 화면을 통하여 시스템에 이상이 발생하 였음을 쉽게 알 수 있다. 그리고 '.'과 '13', '16' 등과 같이 숫자들로 나타나 있는 정보가 화면에 이어서 나타나 있으며, 이들의 각각은 하나의 노드를 의미한다. 그리고, '.'은 작업을 수행하지 않고 있는 노드이며, 숫자의 표시는 그 노드에서 수행되는 사용자 작업의 수를 나타낸다. 노드에 장애가 발생하였다면 화면에는 붉은 색의 'x'가 표시되며 알람도 울린다(그림 5). 'x'와 'F' 표시와 알람은 설정된 시간 간격(30초)으로 그 노드가 복구될 때까지 계속 된다. 이 모니터링 도구에서 보여주는 노드에 대한 정보 외에 사용자 작업에 대한 정보가 또한 제공되며, 그 정보는 PBS 명령어인 qstat 명령어의 정보와 유사하다.



▶▶ 그림 4. 화면 구성

[표 3] 프로그램 구성 설명

[job]: 사용자의 작업
Run: 현재 실행되는 작업의 수
Queue: 대기중인 작업의 수
[CPU]: CPU 현황
Total: 전체 CPU의 수
Run: 사용자 작업이 실행되고 있는 CPU의 수
Down: down된 CPU의 수
Idle: 대기중인 CPU의 수
[Node]: 노드 상태
InfiniB info: InfiniBand 네트워크 상태
.: 정상 상태
F: 비정상 상태
Nodes: 노드 상태
.: 정상 상태(Idle 상태)
숫자: 노드에서 수행되는 작업의 수
x: 노드가 비정상 상태

▶▶ 그림 5. 장애 발생시 모니터링 화면 예

Ⅳ. 결 론

본 논문에서는 7노드(112 CPU)의 소규모 SMP 클러스터 시스템을 모니터링하기 위하여 개발한 도구에 대하여 기술하였다. 이 도구에서는 클러스터 시스템에서 각 노드의 생사, Interconnect 스위치인 InfiniBand 스위치의 상태 그리고 PBS 사용자 작업에 대한 정보를 제공한다. 그리고 shell 스크립터로 개발되어 설치하여 사용하는데 어렵지 않다.

이 도구를 통하여 클러스터 시스템과 사용자 작업의 모니터링이 가능하며, 클러스터 시스템의 환경(규모 등)이 바뀌더라도 쉽게 수정하여 모니터링이 가능하다.

향후의 작업으로는 성능을 측정하여 다른 툴들과 비교를 하여 기능성뿐만 아니라 성능적인 측면도 알아볼 예정이다.

▌참 고 문 헌▐

[1] 최재영, 이준호, 황석찬, “클러스터를 위한 소프트웨어 도구”, 정보과학회지, 제18권 3호, 2000년 3월.
[2] 박유찬, 홍태영, “클러스터 시스템을 위한 단일 모니터링 에이전트에 대한 연구”, 한국정보과학회, Vol.32, No2(1), 2005년
[3] 성진우, 이영주, 이상동, 김중권, “클러스터 시스템의 모니터링 도구 설계 및 구현”, 한국정보처리학회 논문집 11권 제2호, 2004년
[4] 심형용, 선동국, 김성조, “리눅스 클러스터 모니터링 시스템 설계”, 정보과학회지, vol. 29. No 1, 2002년
[5] 조혜영, 홍태영, 홍정우, 클러스터 시스템 관리 도구에 관한 연구, 한국정보처리학회, 제11권, 2호, 1043, 2004년
[6] http://www.openpbs.org/main.html