소규모 SMP 클러스터 시스템을 위한 모니터링 툴의 성능 시험

성진우, 이영주, 최윤근, 박찬열
한국과학기술정보연구원 슈퍼컴퓨팅센터
e-mail: jwsung@kisti.re.kr

Performance Test of Monitoring Tool for Small SMP Cluster System

Jin-Woo Sung, Young-Joo Lee, Youn-Keun Choi,
Chan-Yeol Park
Supercomputing center,
Korea Institute of Science and Technology Information(KISTI)

요 약

다수의 노드로 구성된 클러스터 시스템을 관리하기 위하여 모니터링 툴(S/W)이 필요하지만, 자신의 시스템에 적합한 툴을 확보한다는 것은 쉽지가 않다. 본 문서는 소규모 SMP 클러스터 시스템을 위하여 개발한 모니터링 툴(my-mon)과 성능 시험 내용을 기술하였다. ganglia와 같이 웹기반의 툴들도 있지만 필요한 기능들로 구성된 my-mon은 다양한 구조의 클러스터 시스템을 관리하는 관리자에게는 맞춤형 클러스터 모니터링 툴이다. infiniband 네트워크를 계산노드간의 스위치로 구성된 소규모 SMP 클러스터 시스템용 모니터링 툴의 개발 내용과 툴의 성능(CPU사용율과 메모리 사용량)을 측정하여 웹기반의 툴들과 비교한 결과를 기술하였다.

1. 서론

클러스터 시스템은 일반적으로 수십 대에서 수백 대 이상의 노드로 구성되어 있으며, 각각의 노드는 독립적인 운영체제로 구동된다. 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.[1] 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, 금년에는 Infiniband 스위치 구성의 소규모(112CPU/7노드) SMP 클러스터 시스템의 모니터링을 위한 툴을 개발하였다.[2]

본 논문에서는 금년에 개발한 소규모 SMP클러스터 시스템을 위한 툴의 개발 내용과 웹기반의 툴들과의 성능 비교한 결과를 설명한다.

논문의 구성은, 2장에서 모니터링 도구에 대해 간략히 기술하였으며, 3장에서 모니터링 툴 개발에 대하여 기술하고, 4장은 성능 시험, 5장에서는 결론에 대하여 기술하였다.

2. 모니터링 도구[3]

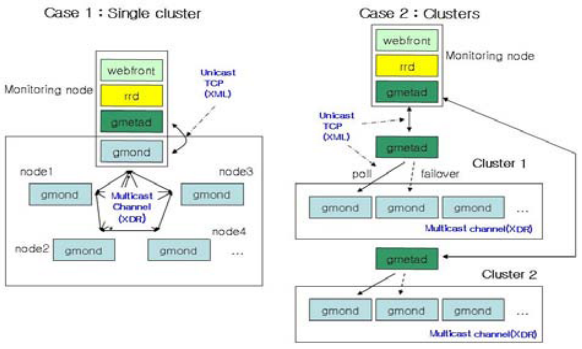
Ganglia

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.

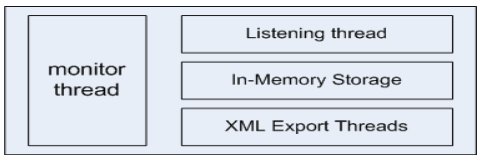
Ganglia는 분산된 서비스 구조와 multicast를 통해 안정적인 scalability를 보장하며, 기본적으로 그리드 환경에서 사용할 수 있는 구조를 가지고 있다. 그리고 RRD를 database로 사용하여, 시스템의 상태와 성능에 대한 정보를 주기적으로 저장하고, 웹을 통한 편리한 관리 기능을 지원한다. 그러나 다음과 같은 몇 가지 보완되어야 할 점 들을 지적할 수 있다. 첫째 사용자가 모니터링하고자 하는 노드 정보를 추

가하는 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. 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 1) Ganglia 프로그램 구성도



(그림 2) gmond 구성도

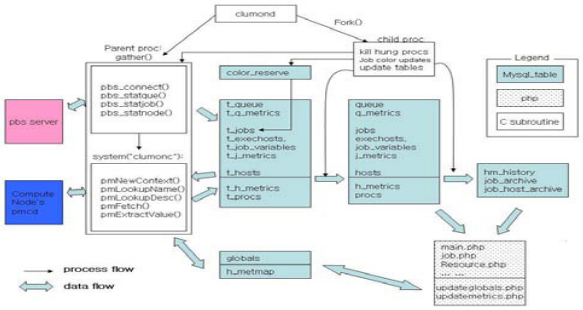
Clumon

Clumon은 공개 클러스터 시스템 모니터링 도구로서, NCSA(The National Center for Super-computing 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.

Clumon은 클러스터 시스템의 성능에 대한 정보를 얻기 위해 PCP 및 PBS 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.



[그림 3] Clumon의 프로그램 구성도

3. 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 같다. 점검 결과는 파일로 보관하도록 하였다.

<표 2> my-mon 소스 일부

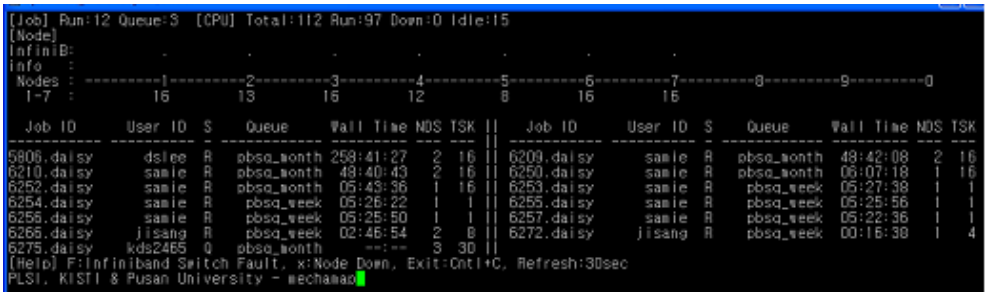
```
NODELIST="node1 node2 node3 node4 node5 node6
node7"
for i in `echo $NODELIST`
do
  if ping $i -c 1 -w 1 > /dev/null 2>&1
  then
      echo > /dev/null
  else
      echo "$i x" >> $Giga_State_File
  fi
done
```

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 발생하였음을 쉽게 알 수 있다.

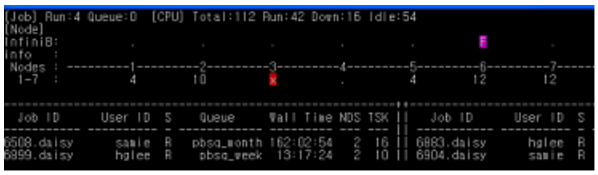
그리고 '.'과 '13', '16' 등과 같이 숫자들로 나타나 있는 정보가 화면에 이어서 나타나 있으며, 이들의 각각은 하나의 노드를 의미한다. 그리고, '.'은 작업을 수행하지 않고 있는 노드이며, 숫자의 표시는 그 노드에서 수행되는 사용자 작업의 수를 나타낸다.



(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 모니터링 화면 예

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.

4. 성능 시험

이 절에서는 모니터링 툴이 소모하는 자원을 알아보기 위하여 실시한 시험에 대한 내용을 기술하고자 한다. 소모하는 자원은 CPU와 Memory 자원 소모만을 비교하였다.

테스트 환경은 표 4에 나타내었으며, 테스트는 한 시스템에서 이루어 진 것이 아니라 두 시스템에서 이루어졌다. 모니터링 툴들은 ganglia, clumon, supermon이며, 이들의 시험 자료는 참고문헌[1]에서 테스트한 결과를 참조하였다. 모니터링 툴(my-mon)의 성능은 **time**과 **top** 명령을 이용하였다. 정밀한 데이터 측정을 위하여 top 명령어의 source를 수정하여 원하는 데이터를 구하였다. 테스트 횟수는 각각 400회를 수행하였으며, 다음의 표 5는 my-mon의 CPU 및

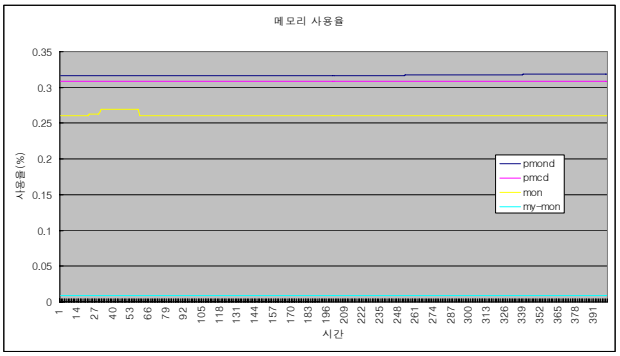
[표 4] 테스트 환경

CPU	메모리	OS	커널	모니터링 도구			
				ganglia	clumon	supermon	my-mon
2.4GHz	8GB	Linux	2.4.21				1.0
1.5GHz	512MB	CentOS 4.0	2.4.31	3.0.0	2.4.0	1.4	

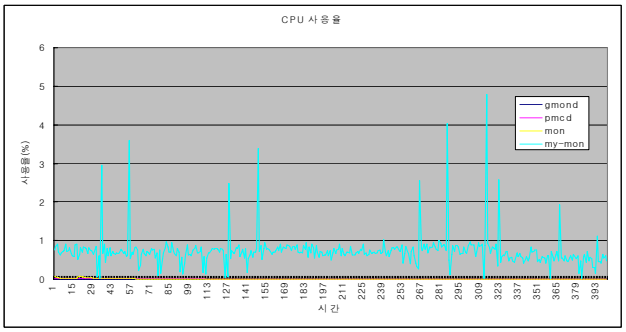
Memory의 자원 소모율을 나타낸 것이다. my-mon의 CPU 사용율은 0.722%이며, 타 툴들은 0.01%이하이다. 이는 타 툴들에 비하여 180배 정도 높게 사용하였다. 그리고 메모리 사용율은 65.5MB를 사용하였으며, 이는 타툴의 40%정도 사용량이다.(그림 6과 그림 7 참조)

[표 5] 시스템 자원 사용 비교

구분	CPU 사용율(%)	Memory 사용량(MB)
gmond	0.004	162.3
pmcd	0.008	65.5
mon	0.004	65.5
my-mon	0.722	65.5



(그림 6) 메모리 사용율



(그림 7) CPU 사용율

5. 결론

클러스터 시스템의 보급과 활용이 높아지고, 구성 형태가 다양해짐에 따라 시스템의 모니터링에 대한 연구 및 개발이 빠르게 이뤄지고 있다. 본 논문에서는 소규모의 SMP 클러스터 시스템의 모니터링을 위한 툴(my-mon)을 개발하였다. my-mon은 128CPU와 512CPU의 모니터링 툴 개발에 이어 개발한 것으로, script 기반이다. 모니터링 대상은 계산노드와 계산노드간의 네트워크 스위치 그리고 PBS 스케쥴러의 작업이다. my-mon의 성능을 확인하기 위하여 시스템 자원의 사용(CPU 사용율과 메모리 사용량)을 측정하였다. 성능시험의 결과는 my-mon이 타툴에 CPU 사용은 180배나 높게 사용하였으며, 메모리는 타툴의 40%정도인 65.5MB정도 사용하였다.

향후 연구는 CPU 사용율을 줄이기 위하여 script 대신에 Python, C 등의 보다 효율적인 언어로 코드를 변환하는 것이다.

참고문헌

[1] 박유찬, 홍태영, “클러스터 시스템을 위한 단일 모니터링 에이전트에 대한 연구”, 한국정보과학회, Vol.32, No2(1), 2005년

[2] 성진우, 이영주, 이상동, 김중권, “클러스터 시스템의 모니터링 도구 설계 및 구현”, 한국정보처리학회 논문집 11권 제2호, 2004년

[3] 조혜영, 홍태영, 홍정우, 클러스터 시스템 관리 도구에 관한 연구, 한국정보처리학회, 제11권, 2호, 1043, 2004년

[4] SGI 007-3773-003, "Message Passing Toolkit (MPT) User's Guide"

[5] PBS Pro 7.0 Administrator's guide