

웹 기반 분산/병렬 프레임워크상에서 실시간 자원 모니터링

김수자*, 정재홍*, 송은하*, 한성국*, 주수중*, 정영식*

*원광대학교 컴퓨터 및 정보통신 공학부

e-mail:{sujkim, jhjeong, ehsong, skhan, scjoo, ysjeong}
@wonkwang.ac.kr

Real-Time Monitoring of Resource for Distributed/Parallel Framework on the Web

Su-Ja Kim*, Jae-Hong Jeong*, Eun-Ha Song*, Sung-Kook Han*,
Su-Chong Joo*, Young-Sik Jeong*

*School of Computer · Information Engineering, Wonkwang
University

요 약

웹의 다양한 자원을 이용하여 고성능 작업 처리를 요구하는 분산/병렬 시스템은 균형적인 작업 할당을 위해 각 호스트의 성능 평가가 필요하다. 하지만 성능 평가에 대한 지속적인 신뢰하기가 어려우며 뿐만 아니라, 작업 도중 호스트의 성능 변화를 예측하기가 어렵다. 성능 변화에 따른 효율적인 작업 스케줄링이 필요하며, 자원 관리자는 작업을 수행중인 호스트에 대한 모니터가 요구된다. 본 논문에서는 자원 관리자와 시스템 관리자에게 효율적인 자원 정책을 제안하기 위해 각 호스트의 자원을 모니터하고, 분산/병렬 시스템의 작업 할당 메커니즘에 의해 각 호스트의 성능 평가 기준을 정한다. 또한 관리자에게 실시간으로 호스트의 성능 변화에 따른 자원 정보를 관리하도록 다양한 시각화를 제공한다.

1. 서론

자원이 풍부한 웹 환경에서 새로운 기종과 성능이 좋은 자원의 증가로 인해 기존의 많은 자원들이 낭비되고 있는 상황이다. 웹을 통하여 지리적으로 떨어져 있는 자원들을 사용하기 위한 분산/병렬 컴퓨팅 연구가 발전하며 그 결과 소비되지 않고 또 그 성능에 비해 효율성이 극히 떨어지는 호스트들을 하나의 고성능 컴퓨터로서의 역할을 해줄 수 있다[1]. 이러한 환경에 참여하는 호스트들은 대용량 작업을 수행하기 위한 간단한 벤치마크를 통한 스케줄링을 한다. 이때, 사용되는 결과가 일정시간 즉, 연산 수행 도중의 현재 자원 이용률과 호스트의 장애 추적(troubleshooting) 검출 등의 연산 도중에 발생 가

능한 호스트 신뢰성 관측이 요구된다. 또한, 약결합(loosely coupled) 이기종의 호스트로 구성된 분산 시스템에서 특정한 작업 스케줄링에 의한 추가 연산을 요구할 때 기준치 결정과 호스트 내부적으로 수행되는 처리 자원에 대한 모니터링이 필요하다.

DRMonitor[2]는 네트워크상의 개인 컴퓨터의 계산 자원 사용률을 모니터링 하는 시스템으로 모니터링된 자원 정보를 기록하지 않지만 부하 균형 정책을 도와주기 위한 성능 평가 값을 일정시간 마다 갱신한다. 고성능 클러스터 모니터링 시스템인 Supermon[3]은 기존의 방법보다 빠르게 노드의 행동 모니터가 가능하나 특정한 시스템에 대한 모니터링으로 제한된다. NWS(Network Weather Service)[4]는 SNMP로 관리되는 네트워크에서 사용하기 위한 서비스이다. SNMP 에이전트를 사용하기 때문에 광범위한 자원 요소 집합을 가지고 있지만

* 본 연구는 2002년 정보통신부에서 지원하는 기초기술연구지원사업(C1-2002-105-007-3)으로 수행되었음.

SNMP를 지원하지 않는 환경에서는 모니터링 할 수 없다.

(표 1) 기존의 모니터링 시스템과 비교

System Item	DRMonitor	Supermon	NSW	제안 시스템
fault tolerance	×	○	×	○
Real-time usage	○	×	○	○
Running Env.	agent	mon	SNMP agent	applet
log report	×	×	×	○
native technology	C	RPC	SNMP	JNI
job scheduling	×	×	×	○
visualizing	○	×	○	○

본 논문에서는 분산 환경을 위한 작업 스케줄링 모니터링 엔진을 RPM(Resource Process Monitoring)이라 정의하며, 웹 기반 실시간 다양한 자원 요소 집합에 의해 부하 균형 및 결합 허용을 제공하며 이들의 자원 정보를 시각화한다.

본 논문에서 요구되는 모니터링 기능은 다음과 같다.

- 다양한 자원 요소 : 호스트에 영향을 주는 자원 과 각 특징을 나타내는 자원 정보를 표현한다.
- 동기화 : 시간 흐름에 따라 자원 정보값을 출력 하는데 있어서 동기화가 되어야 한다.
- 시각화 : 관리자가 작업에 참여한 호스트들의 자원 정보를 비교하기 쉽도록 다양한 디스플레이가 지원되어야 한다.
- 재사용 : 관리자가 재사용할 수 있도록 자원 정보를 파일에 기록한다.
- 성능 평가 : 전체 작업이 완료되면 각 호스트의 자원 정보 기록을 사용하여 성능 평가 레벨값을 설정한다.

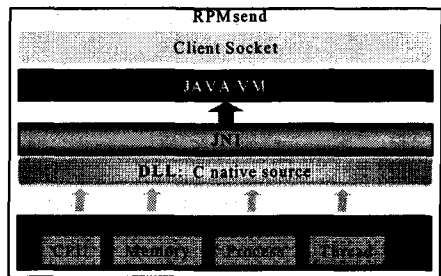
2. 실시간 자원 추출과 성능 레벨 측정

2.1 자원 요소와 추출 메커니즘

자원을 모니터링 하는데 있어서 성능을 평가하기 위해 어떠한 자원을 추출할 것인지 또는 그 자원이 시스템에 어느 정도의 영향력을 주는지 고려해야 한다. 자원 요소를 시간에 흐름에 따라 변하지 않고 일정한 값을 가지는 정적 요소와 변하는 동적 요소로 구분한다. 정적 요소로는 IP 주소, 컴퓨터 이름, 운영체제의 종류, 전체 메모리량, CPU 개수와 종류를 포함한다. 동적 요소는 CPU 사용률, 메모리 사용

률, 사용 중인 메모리 용량, 생성된 프로세서 수, 쓰레드 수 등이다.

자원을 추출하는 메커니즘은 JNI(Java Native Interface)를 사용한다. 자바는 바이트 코드를 인터프리트하여 수행하므로 네이티브 코드의 속도를 얻을 수 없기 때문에 빠른 처리가 요구되는 대량의 계산 작업이나 실시간 처리에서 자바를 이용하기 힘들다. 따라서, 자바환경에서 하드웨어의 접근과 실시간으로 모니터링하기 위해 JNI를 사용한다.



(그림 1) 시스템의 에이전트 구조

본 논문의 시스템은 (그림 1)과 같이 자바에서 자원 정보를 추출할 메소드를 선언하는 Snapshot을 정의하고 그에 대한 헤더 파일을 생성한다. 자바에서 선언한 메소드의 구현은 C native source에서 이루어진다. 구현한 C 소스파일은 DLL로 컴파일하고 Snapshot에서 DLL을 로드하여 사용한다.

자원 정보를 전송할 때, 요소 정보를 나타내는 토큰을 달아서 전송하여 수신측에서 데이터를 필터링이 용이하도록 한다. 토큰 종류는 NAME, STATIC, DYNAMIC이다.

2.2 성능 레벨 측정

RPMrecv은 소켓을 통해 들어온 패킷을 수신하고 각 패킷의 토큰을 분리하여 정보를 각 요소에 저장하고 (그림 2)처럼 파일에 기록한다.

```
##### Resource Provider Information #####
Write Date: Tue Mar 18 11:28:48 GMT+09:00 2003
RP#1Name: forvee IP: 210.112.129.108
STATIC CPUINFO: 1296 1 0 15
STATIC MEMINFO: 261424 2097024 632032 99
DYNAMIC USAGE: 100 61
DYNAMIC MEM: 240612 2464 1918688
DYNAMIC ETC: 38 0 440
```

(그림 2) 모니터링 자원 정보 기록 파일 샘플

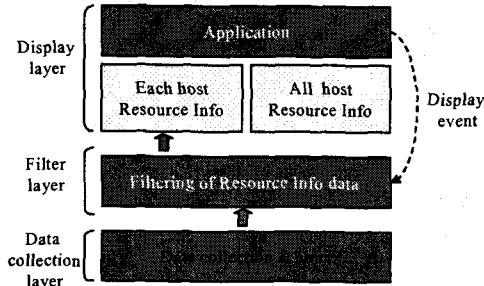
작업이 완료된 후, 각 호스트에 대해 기록된 자원 정보를 가지고 성능 평가 레벨값을 설정한다. 호스트의 성능을 비교할 수 있는 자원 정보는 지원되는 CPU 속도, 전체 메모리량과 작업 중의 CPU 사용률 변화 빈도수를 이용한다. 메모리 사용률 변화는

CPU 사용률 변화 추이보다 성능 평가에 적용하기에는 약하므로 성능 레벨 측정에서 배제한다. 이 성능 레벨 측정값은 작업에 참여할 때 성능적 작업 할당 매커니즘에서 호스트의 성능 평가 기반이 된다.

3. 모니터링 정보 시각화

3.1 시각화 구조

시각화는 (그림 3)과 같이 데이터 수집층, 필터층, 디스플레이 층으로 이루어진다.



(그림 3) 시각화 3레벨 구조

데이터 수집층은 RPMsend에서 수신받은 데이터를 받아 토큰별로 분리하는 역할을 하며, 필터층은 관찰자가 선택한 디스플레이에 맞도록 모니터 화면을 설정하고 디스플레이에 알맞은 요소값을 필터링한다. 디스플레이층은 자원 정보를 보여주는 GUI 부분으로 기능별로 여러 디스플레이를 제공한다.

3.2 시각화 표현 방법

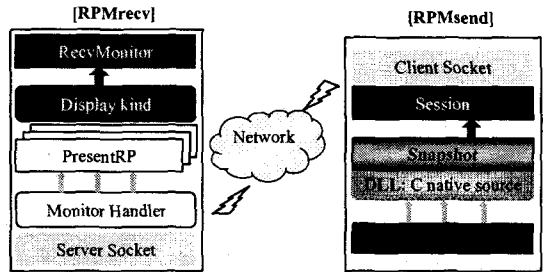
자원 정보를 디스플레이 하는 방법으로는 텍스트 표현방식, 시간 처리 다이어그램, 다중 동시 뷰로 시각화된다. 텍스트 표현 방식은 가장 보편적인 형식으로 호스트 전체 자원 정보를 그룹화하여 표현한다. 특별한 디스플레이 알고리즘이 요구되지 않고 정보를 단순히 텍스트로 변환한다. 시간 처리 다이어그램 방식은 CPU 사용률과 메모리 사용률을 시간 흐름에 따라 표현하기 위해 사용한다. 이 방식에서 Y축은 각 자원의 퍼센트를 나타내고 X축은 시간을 표시한다. 다중 동시 뷰 방식은 텍스트 표현 방식과 유사하지만, 현재 작업에 참여한 호스트의 성능을 비교하기 위해 주요 자원 정보에 대해서만 표현한다.

4. RPM 엔진 설계 및 구현

4.1 전체 시스템 구성도 및 통신 프로토콜

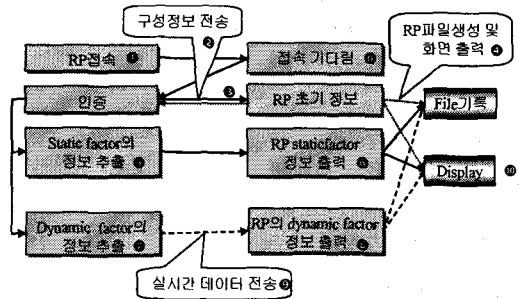
RPM(Resource Provider Monitor)는 (그림 4)와 같이 웹 브라우저를 통하여 RM에 접속하고 자신의

자원 정보를 추출하여 전송하는 역할을 하는 RPMsend와 접속한 RP에 대한 핸들러를 생성하여 파일 생성 및 호스트를 관리하고 각 자원 정보를 시각화하는 하는 RPMrecv로 구성된다.



(그림 4) RPM 전체 구성도

RPMsend는 자원 정보를 추출하기 위한 native method를 가진 Snapshot와 추출한 정보를 서버 소켓에 전송해 주기 위한 session이 있으며, session은 일정 시간 간격으로 자원 정보를 갱신하여 RPMrecv에게 전송한다. RPMrecv는 서버 소켓에 접속한 호스트를 관리하기 위해 MonitorHandler 핸들러를 생성한다. MonitorHandler는 각 호스트의 자원 정보를 저장할 PresentRP와 자원 정보를 기록할 파일을 생성한다. RPMrecv의 GUI는 RecvMonitor로 관찰자가 이벤트를 발생하면 해당하는 디스플레이 정보를 보여주기 위해 각 디스플레이를 설정한다.



(그림 5) RPM 엔진의 통신 프로토콜

^①RPMsend는 웹 브라우저를 통해 RPMrecv에 접속한다. ^②접속한 RPMrecv는 환경 정보를 전송한다. 환경 정보는 화면을 갱신하기 위한 일정 간격 시간(Ti)값 등이다. ^③RPMsend는 IP address, network name을 전송한다. ^④RPMrecv는 접속한 RP의 IP와 name으로 식별 가능한 PresentRP를 생성하고 RecvMonitor에 RP를 추가한다. ^⑤RPMsend는 정적 요소에 관한 정보를 추출하여 전송한다. ^⑥RPMrecv

는 정적 요소의 정보를 저장하고 파일에 기록한다. ①정적 요소에 관한 정보를 전송한 RPMsend는 일정 시간 간격(Ti)마다 동적 요소의 자원 정보값을 전송한다. ②RPMrecv는 받은 동적 요소값을 파일에 기록하고 현재 디스플레이 화면에 맞도록 출력한다.

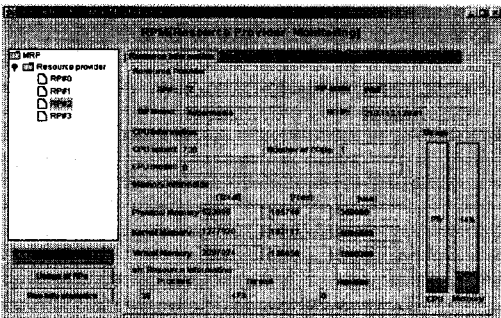
4.2 RPM 구현

RPM 엔진은 4개의 패키지로 구성된다.

- config : 서버 IP, RPM 정보, socket port 번호, 동적 요소 업데이트 시간 간격 종류 및 설정된 간격값을 포함한다.
- factor : 자원 정보값을 가진다. staticfactor와 dynamicfactor로 구분되며, RPMsend측에서 자원을 추출하여 저장할 때 구성 요소로 사용한다.
- send : RPMsend측에 관한 패키지이다. 자원 정보를 추출하는 native method를 가진 Snapshot, 일정시간 간격(Ti)으로 자원 정보를 갱신하고 RPMsend를 관리하는 sendManager, RPMrecv에 접속하여 자원 정보를 전송하는 session으로 이루어져 있다.
- recv : RPMrecv에 관한 패키지이다. 접속한 호스트를 관리하기 위한 MonitorHandler와 각 호스트에 대한 자원 정보값을 저장하고 파일에 기록하는 PresentRP, GUI 부분인 RecvMonitor로 구성된다.

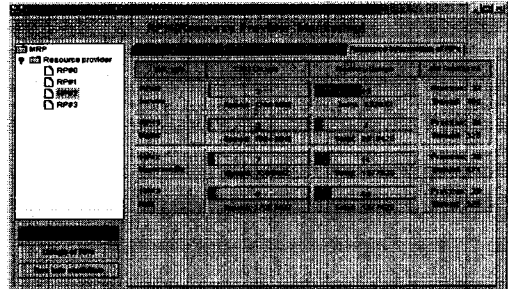
5. 실시간 자원 모니터링 성능평가

RPM 시스템 테스트 환경은 PDSWeb에서 이루어진다. PDSWeb은 RP가 RM에게 작업 참여의사를 보내면 성능비대로 작업을 할당하는 분산/병렬 시스템이다. (그림 6)은 참여한 4개의 RP에 대한 자원 정보를 세부적으로 알 수 있으며 현재 작업 상태를 알 수 있다. 현재 테스트에 참여한 호스트는 작업을 할당받기 위해 대기 상태를 가진다.



(그림 6) 호스트별 자원 정보

(그림7)은 실시간으로 갱신된 정보를 디스플레이 하여 작업에 참여한 호스트의 성능을 비교하는데 있어서 효율적이다.



(그림 7) 작업에 참여한 전체 호스트 자원 정보

6. 결론

본 논문에서는 웹을 통해 자원을 제공하는 호스트에 대한 자원 모니터링을 하기 위한 시스템에 대해 제안하였다. 자원 요소는 정적 요소와 동적 요소로 구분하였고, 실시간으로 자원 사용량의 변화를 관리자가 볼 수 있도록 시각화하였다. 호스트의 자원 정보를 이용하여 성능 평가의 기준이 될 수 있도록 자원 정책을 제안하였고 각 호스트 모니터 방식으로 텍스트 표현방식, 시간 처리 다이어그램, 다중 동시 뷰 방식을 제안했다. 향후 과제로 호스트 작업 참여 상태와 다양한 자원 모니터링이 요구되며, 플랫폼 독립성을 제공해야 한다.

참고문헌

- [1] 송은하, 정영식. "웹 환경에서의 병렬/분산 처리를 위한 동적 호스트 관리 기법의 개발" 정보과학회논문지 8권 3호. 2002.
- [2] Domingues, P. Silva, L. Silva, J.G. "DRMonitor - A Distributed Resource Monitoring System" Parallel, Distributed and Network-Based Processing, 2003. Proceedings. Eleventh Euromicro Conference on 127 -133.
- [3] Sottile, M. J, Minnich R.G "Supermon: A high-speed cluster monitoring system" cluster Computing, 2002. Proceedings. IEEE International Conference on 39-46.
- [4] Busby, R.E. Jr. Neilsen, M.L. Andresen, D. "Enhancing NWS for use in an SNMP managed internetwork" IPDPS 2000. Proceedings. 14th International 506-511.