

그리드 환경에서 모니터링 주기와 오버헤드에 관한 연구

김혜주^o 허의남 이웅재

서울여자대학교

{judy012^o, huh, wjlee}@swu.ac.kr

An Optimal Monitoring Interval for Grid Resource Information Services

Hye-ju Kim^o Eui-Nam Huh woong-jae lee

Division of Information and Communication, Seoul Women's University

요 약

그리드 환경은 수많은 컴퓨팅 자원을 활용하는 시스템으로 자원에 대한 정보가 필요하다. 자원 정보 검색을 위해서는 모니터링이 전체 자원에서 일어나야 하므로 이에 따른 오버헤드가 예상된다. 본 논문에서는 모니터링의 오버헤드를 줄이면서도 효율적으로 정확하게 자원 정보를 수집 할 수 있는 모니터링 기술을 구현하였다. 여기서 제안된 모니터링 기술은 시스템 성능에 따라 자동적으로 모니터링 주기를 조절하게 된다. 이를 통해 모니터링에 의한 오버헤드가 줄었음을 측정 결과를 통해 보일 것이다.

1. 서 론

인터넷 대역폭의 확장으로 인해 LAN이나 자체 인터넷을 이용하는 클러스터링에서 벗어나 국가나 전세계를 포함하는, 여기에는 단순히 컴퓨터만 피어로 참여하는 것이 아니라 대용량 저장 장치, 다양한 고성능 연구 장비들이 포함되고 있는데, 이렇게 통합된 형태를 그리드라고 한다. [1]

그리드 환경에서의 다양하고 많은 자원들을 그리드 자원 관리 시스템에서 자원을 수집하고 자원의 정보와 상태를 파악하여 사용자에게 분배를 한다. 이때 자원에 대한 모니터링이 필요하다. [2][3]

실시간으로 변하는 그리드 자원을 정확하게 파악하기 위해서는 모니터링 또한 실시간으로 이루어져야 할 것이다. 하지만 모니터링이 실시간으로 이루어진다면 모니터링에 의한 오버헤드가 성능을 저하시킨다는 것이다. 그래서 반드시 고려되어야 할 사항이 얼마나 자주 모니터링이 이루어져야 하는가 하는 모니터링 주기이다. 너무 자주 모니터링이 이루어진다면 모니터링에 의한 소요 비용인 오버헤드를 많아지게 될 것이며, 너무 늦게 모니터링이 이루어진다면 변화된 자원에 대한 정보가 불확실하게 될 것이다.

본 연구에서는 그리드 환경에서 모니터링이 이루어져야 하는 주기를 시스템 상태 변화를 주는 방법을 제안하고자 한다. 시스템 상황이 얼마나 자주 변하는지, 자원의 오버헤드 양에 따라, 자원 정보의 불확실성에 대한 손실률 등을 이용하여 주기를 계산하는 방법을 모델링 하였으며, 이를 이용해 주기를 실제 계산해서 적용해 보았다.

2. 관련 연구

현재 그리드 환경에서 그리드 모니터링 구조를 분석해 보면 그리드 타입의 요소들 사이에 모니터링에 필요한 데이터는 성능 문제의 소스를 결정하는 것 그리고 시스템과 어플리케이션을 변화시키는 것이다. 이 데이터들로부터 오류 발견과 회복 메커니즘을 결정된다. 그리드 모니터링 구조의 세 가지 구성요소를 보면 directory service는 정보를 발행하고 발견하도록 지원하고, producer(performance event source)는 성능 데이터를 이용하도록 만든다. Consumer(performance event sink)는 성능 데이터를 받는다. [3]

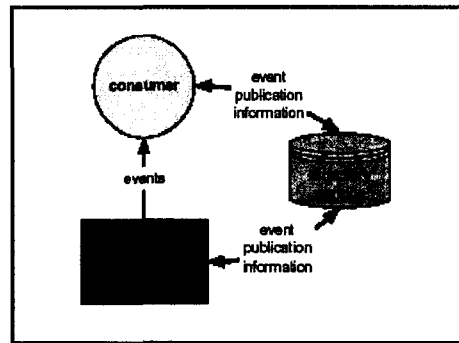


그림 1 그리드 모니터링 컴포넌트

위의 그림의 컴포넌트 각각의 구조의 기능을 살펴보면, producers는 LDAP을 이용해 directory service에게 접근하여 자신의 존재를 알린다. 또한 consumer는 관심 있는 producers를 발견하기 위해 directory service를 이용할 수 있다. directory service는 producers와 consumer의

위치를 알아내는데 이용된다.

그리드 모니터링 구조는 time-stamped event와 같이 전송된 성능 데이터를 다루도록 설계됐으며, 이벤트는 이벤트 스키마에 의해 정의된 구조로 수집된 데이터의 타입이다.[4]

자원 정보를 좀더 정확하게 알기 위해서나 오류를 막기 위해서 설계되어 있다. 하지만 모니터링이 가지고 있는 근본적인 문제였던 모니터링에 의한 자원의 오버헤드에 대한 대책이 세워지지 못했다. 본 연구에서는 그리드 환경에서 모니터링 주기가 자원의 상태 변화정도에 따라 최적의 주기로 모니터링하여 모니터링에 의한 오버헤드를 줄이는 모델을 제안하고자 한다.[5]

3. 최적 모니터링 주기(Optimal Monitoring Interval)

자원 정보 관리 시스템에서는 자원의 정보를 모니터링하여 사용자가 원하는 만큼 자원을 할당해주고, 또한 사용자가 사용중인 자원에 대해서도 지속적으로 모니터링하여 이 자원에 문제가 발생하였을 경우를 대비하고, 문제가 발생했을 경우에는 모니터링된 자원 정보를 가지고 다시 재분배를 해주는 것이 기본 메커니즘이다.[6] 이때 모니터링을 얼마나 자주하는 것이 효율적일까?에 대한 질문은 자원 정보 관리 시스템에서 자원을 정확하게 효율적으로 적용, 관리하기 위해서는 반드시 필요하다. 그리드 환경에서 실시간으로 모니터링을 한다면 가장 정확한 자원의 정보를 알 수 있겠지만, 자원의 성능에 대한 오버헤드가 클 것이다. 만약 시스템의 자원상태 변화가 큰 경우에 모니터링의 주기가 길다면 자원의 정보를 정확하게 분석할 수가 없을 것이고, 반대로 자원의 변화가 거의 없는 상태에서 모니터링 주기가 짧다면 자원 성능에 오버헤드가 될 뿐이다.

본 논문에선 그리드 환경에서 주기를 시스템 상황에 따라 변화를 주어 모니터링하여 정확한 자원의 정보를 제공하면서 모니터링의 비용, 오버헤드를 줄이자는 것이다. 그리드 환경에서의 가변적인 사항을 크게 세가지로 나누면, 첫째 사용자가 원하는 자원의 양을 제공하기 위해서 자원 정보를 보는 오버헤드 양, 둘째 실시간으로 자원정보를 갖지 못하는 손실률, 셋째 자원의 상황이 얼마나 다변적인지를 알려주는 상황 변화율 등이다. 위의 3 가지 가변 사항을 이용하여 모델링에 이용될 변수를 정의해 보면 다음과 같다.

첫째 p는 모니터링 주기동안 자원의 정보가 변화될 가능성을 가리킨다. 둘째 M은 모니터링에 의한 오버헤드, 여기서 M 은 cpu, memory, net의 사용률의 합계를 %로 표현하여 이용한다. 셋째 L은 모니터링 주기동안 자원의 정보가 변화하여 일어날 손실량이며, 여기서 L = Rreq □ Robs 한다. Rreq 은 실제 자원의 성능 양을 비율로 나타낸 것이고, Robs 은 그 구간에 모니터링 된 자원 정보 양을 비율로 나타낸 것이다. 넷째 t 은 모니터링 주기로 최적의 t을 계산해 낼 것이다.

예를 들어, 사용자가 자원을 얻고자 한다면, 정보를 얻기 위해 모니터링에 사용되는 L, 모니터링된 자원의 정보가 틀려 생기는 손실 L, 현재 자원의 정보가 얼마나 많이

변화되는가를 나타내는 P 가 있을 것이다. 크게 다섯 단계의 과정을 통해 구할 수 있다.

첫 단계로, 이 때 모니터링 주기를 t이라 하고 그 동안에 변화될 가능성을 t시간 동안 p의 확률로 나타낸다.

두 번째 단계로, 위의 모니터링 주기동안 자원의 정보가 변화하여 손실되는 비용은 t시간 동안 변화될 가능성에 의한 손실량 L이 된다.

세 번째 단계로 여기에 모니터링에 의한 오버헤드를 포함하면, 모니터링 주기의 모든 비용을 계산하면 위의 t시간 동안에 변화에 의해 발생한 손실량과 모니터링 자체에 대한 오버헤드가 된다.

네 번째 단계로 모니터링 되는 주기 시간당의 비용으로 나타내면, 위의 총비용을 t시간으로 나눈 것이 된다. 시간당 모니터링에 이용된 총비용 c를 [식 1]로 나타내었다.

$$c = \frac{M + L \frac{pt(t+1)}{2}}{t} \quad \text{[식 1]}$$

다섯 번째 단계로 최적의 모니터링 주기 r 의 계산을 위해 시간당 총 모니터링 비용을 얻을 수 있게 된다. 최적의 모니터링 주기 r 을 최소화하기 위해 미분을 해주면 [식 2]와 같이 최적의 모니터링 주기 r 을 구할 수 있다.

$$t_{opt} = \sqrt{\frac{2M}{Lp}} \quad \text{[식 2]}$$

[식 2]의 계산 결과처럼 모니터링 주기 t_{opt} 를 쉽게 계산할 수 있다. 즉 시스템의 상태 변화율에 따라 모니터링을 하는 주기가 변하게 되는 것이다.

4. 실험 및 결과

최소한의 모니터링 비용을 구하기 위해서 위에서 제안한 가변적인 사항을 적용해 적합한 t_{opt}을 찾아보자. 보내려는 자원을 멀티미디어 데이터라고 할 때 멀티미디어 데이터의 전송 시 잃어버린 패킷 수를 비율로 바꾸어 손실률 L로 하고, 모니터링에 의한 overhead는 cpu, 메모리와 네트워크의 사용률의 평균 값을 M라 하자. 또한 자원 정보의 변화 가능성을 p라 놓고, 이 p 값에 따라, 손실률 L에 따라, 모니터링에 의한 overhead M에 따라 최적의 t을 계산해 보자.

[표 1] optimal 모니터링

L= 5일경우-20%			L= 10일경우-40%			L= 20일경우-80%		
P	M(%)	t _{opt}	p	M(%)	t _{opt}	p	M(%)	t _{opt}
0.2	20	3.1623	0.2	20	2.2361	0.2	20	1.5811
	50	5		50	3.5355		50	2.5
	80	6.3246		80	4.4721		80	3.1623
0.5	20	2	0.5	20	1.4142	0.5	20	1
	50	3.1623		50	2.2361		50	1.5811
	80	4		80	2.8284		80	2
0.8	20	1.5811	0.8	20	1.118	0.8	20	0.7906
	50	2.5		50	1.7678		50	1.25
	80	3.1623		80	2.2361		80	1.5811

실험 결과를 살펴보면, 손실률 L와 이때 모니터링에 의한

오버헤드 M 가 같은 경우 자원의 정보가 변화할 가능성이 커질수록 변화가 잦은 시스템이므로 좀더 짧은 주기를 주어야 정확한 자원 정보를 얻을 수 있을 것이다. 또한 같은 자원의 정보가 변할 가능성을 가질 때 손실률 L 가 클수록 주기 짧아져야 정확한 자원 정보를 얻을 수 있을 것이다. 이를 알아보기 쉽도록 모니터링에 의해 overhead M 가 20%일 때 자원의 변화 가능성이 각각 0.2, 0.5, 0.8일 때 자원의 정보가 변화하여 손실되는 정도 L 를 차트로 표현해 보면 다음과 같다.

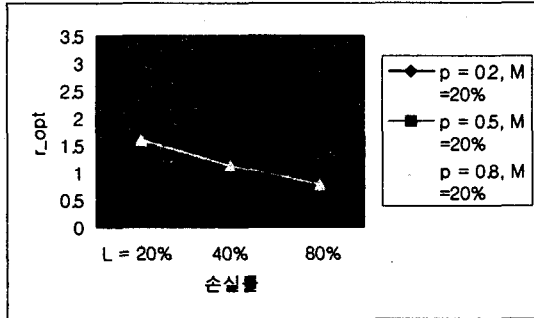


그림 2 모니터링에 대한 오버헤드가 20%인 경우

차트를 보면 자원의 성능 변화 상태 p 가 0.2에서 0.8로 증가하면, 자원의 성능 상태 변화 p 가 크다는 것이므로 자원의 모니터링 주기 t_{opt} 는 짧아지고 있다. 또한 같은 자원의 성능 변화 p 를 가질 때, 즉 자원의 성능변화 p 가 0.5라 한다면 이때 자원 정보의 변화로 인한 손실률 L 이 20%에서 80%가 증가함에 따라 모니터링 주기 t_{opt} 도 짧아진다.

5. 결 론

기존의 모니터링 방식은 그리드 환경의 특성인 유동성이 크고 실시간으로 달라질 가능성이 있는 것을 고려하지 않은 채 일정한 주기로 모니터링을 한다. 다시 말해 현재 그리드 환경이 어떻게 변화하고 있는지도 모른 채 일정한 주기로 모니터링이 이루어진다는 것이다.

본 연구는 자원 정보의 상태 변화율을 파악하여 변화율에 따라 주기를 조절하여 모니터링 하도록 모델링 하였다. 또한 본 논문에서 제안된 모델링을 현재 그리드 모니터링 기반을 두고, 새로운 하나의 모듈을 추가함으로써 모니터링의 주기를 자원 상태 변화율에 따라 optimal 모니터링이 가능하도록 한다. 즉, 본 논문을 적용하는 면에 있어서도 현재의 구성요소와 코드에 변화 없이 적용하기에 쉽게 최적화된 모니터링 주기로 자원 관리를 할 수 있다.

참 고 문 헌

1. I. Foster., "The Grid: A New Infrastructure for 21st Century Science.", *Physics Today*, 55(2):42-47 2002
2. A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, B. Tierney,

- "Giggle: A Framework for Constructing Scalable Replica Location Services.", *Proceedings of Supercomputing 2002 (SC2002)*, November 2002
3. B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swamy, V. Taylor, R. Wolski, "A Grid Monitoring Architecture", *Global Grid Forum Performance Working Group*, March 2000
4. TopoMon: A Monitoring Tool for Grid Network Topology, *Proceedings of ICCS 02*.
5. A. Waheed, W. Smith, J. George, J. Yan. , "An Infrastructure for Monitoring and Management in Computational Grids", In *Proceedings of the 2000 Conference on Languages, Compilers, and Runtime Systems*.
6. monitoring event, " Discovery and Monitoring Event Description(DAMED-WG) ", <http://www-didc.lbl.gov/damed/>