NEC 시스템에서 NQS 및 Load balancing 최적화

이영주*, 이상동*, 김중권*
*한국과학기술정보연구원
e-mail:yjlee@kisti.re.kr

The Implement and Optimization of NQS & Load balancing in NEC System

Young-Joo Lee*, Sang-Dong Lee*, Jung-kwon Kim*
KISTI

요 약

작업관리 시스템인 배치처리 시스템은 한정된 공유자원을 동시에 요구하는 사용자들에게 시스템의 자원을 효율적으로 배분하여 주며, 제시된 다수의 작업들을 순차적으로 수행시켜 준다. 이러한 작업관리 시스템을 시스템 규모 사용자의 요구 조건에 맞게 최적화함으로써 시스템의 자원을 최대한 이용할 수 있게 하며, 다양한 사용자 요구에 적절한 자원 배분할 수 있도록 정의되어야 한다. NQS방식을 개선시켜 이종 모델로 구성된 NEC SX-5 및 SX-6을 단일시스템처럼 관리할 수 있도록 하였으며, 다수의 연동 계산을 원활히 수행키 위한 큐 차등화 서비스방식을 적용시켜 사용자 대기 시간을 최소화시켰다. 본 논문에서는 NQS를 사용하여 이종 다중 노드의 연동 구성 방법과 이것의 최적화 방법 그리고, 노드간 load balancing을 최적 수행 방법 등을 소개한다.

1. 서론

다수의 사용자에게 시스템의 자원을 균등 배분하여 제공해야 하는 경우, 작업 관리 스케줄러로 배치처리 시스템을 사용한다. 배치처리 시스템은 다수의 사용자가 한정된 시스템 자원을 동시에 사용코자할 때 일부 사용자에 의한 자원의 독점을 막아 균등 분배를 원칙으로 배정해 주며, 사용자의 다양한 요구에 대해서도 자원의 분배가 효율적으로 이루어지도록 배정을 진행시킨다.

본 연구는 NEC SX계열의 시스템에 주로 사용하고 있는 작업관리 시스템인 NQS(Network Queueing System)를 개선시켜, 시스템 사양이 다른 NEC SX-5와 SX-6를 하나의 관리 체제하에서 배치처리 시스템이 운영될 수 있도록 새로이 구성하는 것으로서, 사양이 다른 시스템들을 엮어 하나의 운영체제하에 배치시스템을 사용하는 경우의 노드사이의 load balancing을 최적화하는 방법을 제안한다.

2. 시스템 구성

2.1 NEC 시스템 구성

<표 1>에서 보는바와 같이 작업관리 시스템을 구현하고자 하는 NEC 시스템은 모두 3대로서 SX-5와 SX-6로 구성되어 있으며, SX-6의 두 노드가 서로 고속의 IXS(Internode Crossbar Switch)로 연결되어 있고. SX-5와 SX-6는 GiGa 네트웍으로 연결되어 있다. NEC의 login node SX-6a이고 사용자는 login node로만 접속이 가능하고 배치작업의 입력도 load balancing의 구현을 위하여 login node에서만 가능하도록 하였다. 사용자의 홈디렉토리는 서로 공유되어 있고. 사용자의 작업 공간을 위한 두 가지의 큰 스크래치를 제공하는데, 하나는 /xtmp 디렉토리로서 각각의 노드에 local로 연결되어 있어서 I/O가 가능하고 다른 하나는 /ytmp로서 3개의 노드에서 서로 공유되어 있다. 홈디렉토리의 공유방법은 NFS을 개선한 GFS로 구성하였다.

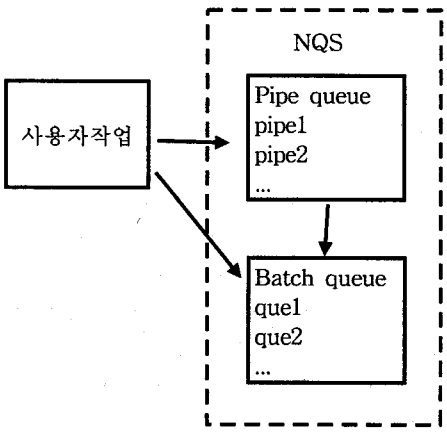
<표 1> KISTI의 NEC SX 시스템 사양

구분	내 용	
모델명	SX-5/8B	SX-6/16M
운영체제	SUPER-UX R13.1	SUPER-UX R13.1
CPU	8개	16개
이론성능	80Gflops	160Gflops
메모리	128GB	128GB
디스크 용량	2.85TB	2.85TB

2.2 NQS 시스템 구성

NQS의 큐는 크게 파이프 큐와 배치 큐의 두 가지로 구성된다. 파이프 큐는 사용자의 작업을 적당한 배치 큐로 할당해주며 배치 큐는 사용의 작업이 최종 도달하여 해당 큐의 정의된 처리 환경 변수에 따라서 작업을 처리하는 큐이다. 파이프 큐는 여러 그룹의 배치 큐를 포함할 수 있으며 배치 큐는 필요한 특성에 따라서 여러 가지로 만들 수 있다.

NQS의 동작은 사용자가 작업을 NQS에 보내면 먼저 파이프 큐로 들어가게 되고 파이프는 이들 작업의 특징을 분석하여 적당한 배치 큐에 할당 한다.



<그림 1 > NQS 구성도

그리고, 사용자가 보낸 작업이 큐를 찾는 방식은 크게 두가지로 분류된다. 하나는 사용자가 직접 자신의 작업 특성에 맞는 해당 큐를 지정하는 것이고, 다른 하나는 자신의 작업에 대한 특성을 적어주면 NQS에서 이를 분석하여 자동적으로 그 작업에 가장 알맞은 배치 큐에 할당시켜 주는 방식이다.

우리가 설계한 NQS 큐는 CPU시간이나 메모리용량에 대한 제한을 두지 않는 것이 특징이며, 때문에 사용자는 큐의 이름만 지정하여도 가능하며 필요에 때라 조건을 부여하도록 되어 있다. 만일 큐의 이름이나 조건을 주지 않으면 default 큐를 할당하게 된다. 우리가 관심을 가지는 작업 노드의 선택은 사용자가 특정 노드를 지정하는 경우를 제외하고는 NQS에서 노드별 가용자원을 분석하여 사용자의 작업이 수행 가능한 배치 큐들을 찾아 자동적으로 각각의 노드에 배분되도록 설계되었다.

2.3 NQS 큐 설계

NQS의 큐의 속성을 정의하는 NQS 환경 변수는 여러 가지가 있지만 그 중에서 시스템 프로세서의 처리에 크게 관련된 변수는 Base priority와 Time slice의 두가지 변수가 있다. Base priority는 프로세스의 할당에 관련되는 변수로서 그 수치가 적을수록 프로세스의 재 할당을 빨리 받게 된다. Time slice는 프로세스가 할당 받는 시간을 간격을 의미하며 해당 시간 동안 계속해서 프로세스를 점유할 수가 있어서 작업 처리가 빠르다.

사용자에게 다양한 서비스를 제공하기 위한 방법으로서 서비스 레벨에 따른 차등큐를 두어서 사용자가 작업의 처리에 대한 긴급성을 고려하여 큐를 선택하게 하였으며 차등큐의 등급은 0.5 ~ 2까지 4단계로 구분하였다. 이렇게 함으로써 긴급한 작업의 빠른 처리가 가능하도록 하였다.

차등큐의 설계에 있어서는 먼저 시스템의 특성과 그동안 사용자의 작업에 대한 패턴을 분석하여 그 결과를 참조하여 설계하였으며 메모리는 충분히 크므로 큐의 정의에는 반영하지 않았다.

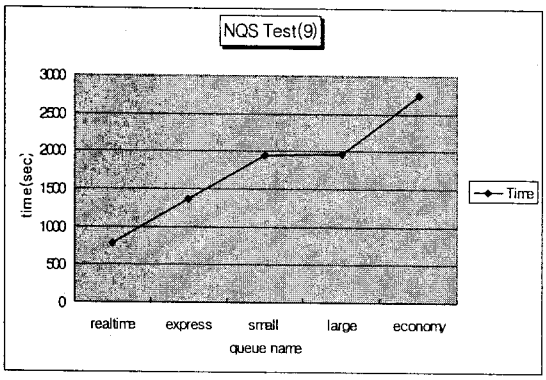
<표 2 > 서비스 레벨 큐 구성

큐 이름		CPU Time	서비스 레벨	Run Limit	Base Pri	Time Slice
dedicated		무제한	2	1	60	2000
realtime		무제한	2	1	60	2000
express		무제한	1.5	2	75	1000
normal	small	180분	1	8	80	1000
	large	무제한	1	2	80	1000
economy		무제한	0.5	4	85	1000

3. 작업관리 시스템 구현

3.1 차등 큐 실행 결과

<그림 2>는 큐를 설계하여 테스트한 결과 중에서 하나의 예를 보여주고 있다. 테스트 방법은 시스템에 다양한 많은 작업을 넣고 CPU의 usage가 포화상태가 되도록 하였으며 테스트 결과 중에서 큐의 factor에 근접하는 테스트 결과를 얻기 위하여 큐의 변수를 변화 하면서 많은 테스트를 하였다. <표 2>는 테스트 한 결과 중에서 차등큐의 factor에 가장 근접한 결과를 보여주고 있다.



<그림 2 > 차등 큐의 테스트 결과

<표 3>은 <그림 2>의 결과를 가지고 단일 노드에 적용한 큐의 변수이다. 큐의 factor가 정해지면 다음은 각각의 큐에 대한 실행 작업 수 설정이 매우 중요하다. .

<표 3 > SX-5 queue table

queue name	CPU time	Service Charging Factor	Job limit			비고
			run	grp	user	
dedicated	unlimited	2	∞	∞	∞	
realtime	unlimited	2	4	2	2	
express	unlimited	1.5	4	2	2	
large	unlimited	1	6	4	2	
small	180 min	1	8	4	2	(8 4 2)
normal	unlimited	1	2	1	1	
economy	unlimited	0.5	4	2	2	

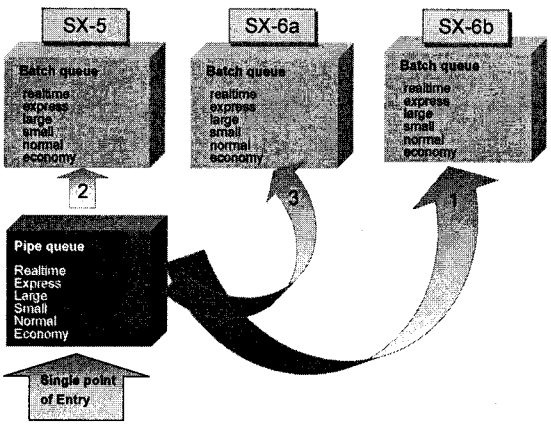
큐의 허용 작업 수를 너무 적게 설정하면 해당 큐로 설정된 작업 수보다 많이 들어왔을 때 작업이 실행되지 못하고 대기하기 때문에 시스템 전체의 효율이 저하될 수 있으며, 또한 실행 작업 수를 너무 많게 설정하면 하나의 큐에서 실행되는 작업만으로도 전체의 CPU를 점유할 수 있기 때문에 다른 큐에서 작업하는 실행 작업에 영향을 줄 수 있다.
이러한 점을 고려하여 실행 작업 수는 시스템의 특성과 작업의 패턴 등을 분석하여 적당히 설정해야 한다

3.2 NQS 다중 노드 큐 구현

다중 노드에서의 NQS 설계를 하기 전에 몇가지 검토 사항이 있다. 첫째는 각 노드끼리 시스템 호환성이고 두 번째는 각각의 노드에서도 각 큐의 특성이 그대로 유지되어야 하며 세 번째는 사용자의 작업이 각 노드의 배치큐에 어떻게 할당할 것인가 하는 방법이다.

<그림 3>은 설계된 큐를 각각의 노드에 설정한 것이다. 큐를 각각의 노드에 설정하는 방법은 여러 가지 방안이 있을 수 있으며 여기에서 설정한 방법은 시스템의 성능이 비슷하다는 전제하에 각각의 차등 큐의 특성을 그대로 유지하기 위한 하나의 방법을 제시하였다.

다중 노드를 연계한 큐 설계에서는 각각 노드 사이에서의 load balancing을 고려하여 설계하여야 하는데 이를 위해서는 배치 큐를 할당하기 전에 CPU usage를 참조해야 하는데 NQS에서 제공하는 load balancing에서 이 기능은 미약하다.



<그림 3 > 노드간의 NQS 구성

3.3 NQS load balancing 구현

NQS에서 각각의 노드에 대한 load balancing은 다음의 3가지의 방법이 제안되고 있다.

① Round Robin Pipe

② Load Information Collection

③ Demand Delivery Method

여기서는 Demand Delivery Method을 이용하여 구현하였다. 이 방법은 다른 두가지 방법을 절충하여 구성된 것으로서 작업의 배치큐 할당에 있어서 비교적 load balancing이 비교적 잘 이루어지는 장점이 있다.

<표 4 >SX-6b NQS queue table

queue name	CPU time	Service Charging Factor	Job limit			비고
			run	grp	user	
dedicated	unlimited	2	∞	∞	∞	
realtime	unlimited	2	2	1	1	
express	unlimited	1.5	2	1	1	
large	unlimited	1	3	2	1	
small	180 min	1	3	2	1	(4 2 2)
normal	unlimited	1	2	1	1	
economy	unlimited	0.5	2	1	1	

<표 5>SX-5 & SX-6a NQS queue table

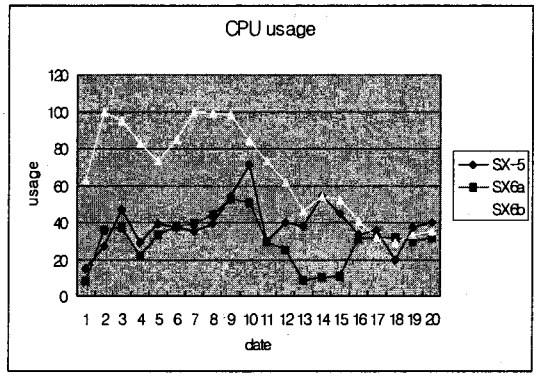
queue name	CPU time	Service Charging Factor	Job limit			비고
			run	grp	user	
dedicated	unlimited	2	∞	∞	∞	
realtime	unlimited	2	3	1	1	
express	unlimited	1.5	3	2	2	
large	unlimited	1	4	2	2	
small	180 min	1	4	2	2	(6 4 2)
normal	unlimited	1	2	1	1	
economy	unlimited	0.5	3	2	2	

사용자가 입력한 작업이 일정한 순서로 각 노드의 배치큐를 찾을 때 먼저 찾는 노드에 작업이 많이 할당될 수 있으므로 이를 감안하여 먼저 찾는 노드의 배치큐의 허용 실행 작업 수를 조정하였다.

이 때 고려 사항으로서는 login node는 다른 노드에 비하여 interactive 작업이 많아 부하가 많을 수 있으므로 load balancing의 할당 순서를 가장 뒤에 할당 되도록 하였다.

3.3 NQS load balancing 실행 결과

<그림 4>는 9일 배치 큐의 허용 작업 수 등을 조정하여 수행한 결과를 보여주고 있다.



<그림 4 >각 노드의 CPU Usage

4. 결론

NQS의 단일 노드 내에서의 서비스 레벨에 따른 차등화 큐 설계를 구현하여 사용자에게 다양한 서비스를 제공하게 만들고 이를 각 노드에서도 그대로 적용되도록 확장하였다. 그리고, 다중 노드에 대한 NQS 설계는 노드 간의 load balancing이 가장 문제가 되는 부분이며 NQS 자체 기능에서는 이러한 기능이 미비하기 때문에 load balancing을 위하여 NQS의 변수와 각 큐의 실행 작업 수, 큐를 찾는 노드의 순서 등으로 구현하였다. 그러나 이러한 방법으로 load balancing을 구현하는 것은 한계가 있으며 이는 load balancing이 가능한 소프트웨어인 ERS 등을 이용하여 노드간의 load balancing을 구현할 경우 최적화가 가능하게 된다.

참고문헌

[1] NEC, "SUPER-UX NQS User's Guide", NEC Corporation

[2] NEC, "SUPER-UX System Design Guide", NEC Corporation

[3] NEC, "Guide to System Operation of Super Computer SX-5 for KISTI", NEC Corporation

[4] 2002년도 슈퍼컴퓨터운영고도화보고서 슈퍼컴퓨터운영실

[5] 우준, 이영주, 김중권 NEC SX-5 시스템에서 NQS 차등화 서비스 레벨 큐 구현 및 최적화