멀티 노드에서의 NQS Loadbalancing 설계 및 구현

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

The NQS Design and Implementation of Load balancing in Multi Node

Yuong-Joo Lee, Jin-Woo Sung, Sang-Dong Lee,
Joong-Kwon Kim
Korea Institute of Science Technology Information

요 약

시스템의 한정된 자원을 다수의 사용자들에게 효율적으로 분배하기 위해서 작업관리 시스템이 사용된다. 이러한 작업관리 시스템은 그 종류가 여러 가지가 사용되고 있는데, 시스템의 종류나 특성에 따라서 적당한 작업관리 시스템을 사용한다. NEC 시스템에서는 작업관리 시스템으로 NQS를 사용하는데, 이 시스템을 이용하여 시스템의 한정된 자원을 다양한 사용자들의 요구에 적절히 자원을 배분 할 수 있도록 설계하고 이러한 배분 방식으로 각각의 노드에서 균형 있게 작업이 실행될 수 있도록 구현한 노드간 최적 분배 방법 등을 기술한다.

1. 서론

작업관리 시스템은 다수의 사용자가 이용하는 시스템의 자원을 효율적으로 활용될 수 있도록 자원을 분배하고 작업에 대한 처리를 관리하는 기능을 한다. 이러한 기능의 시스템에는 대표적으로 NQS, LoadLeveler, PBS, LSF, Condor 등 많은 작업관리 프로그램이 있다. NEC 시스템은 주로 작업관리 시스템으로 NQS(Network Queuing System)을 사용하여 사용자들에게 자원을 제공하고 있다. 일반적으로 작업관리 시스템을 사용자 시스템에 적용하는 경우, 그 설계 방식에 따라 시스템 전체의 작업처리 효율에 많은 영향을 줄 수 있기 때문에 작업관리 시스템을 설계하고자 할 때는 시스템의 환경이나 특성 사용자 작업의 패턴 등을 충분히 분석하여 반영하여야 한다.

본 논문에서는 작업관리 시스템인 NQS 설계에 있어서 시스템 환경에 대한 설명과 시스템의 특성과 NQS에 대한 구조 및 처리 방식 등 큐 설계 시 고려하여야 할 사항과 설계된 NQS를 여러 노드에 적용할 때 노드 간의 Load balancing에 대한 구현 방법 등에 대하여 기술하고자 한다.

2. 시스템 환경

2.1 NEC 시스템

NEC 시스템은 <표 1>에서 보는바와 같이 SX-5, SX-6 2대로 구성되어 있으며 이들 시스템은 벡터시스템으로서 3대의 클러스터 형태로 연결되어 있다. 단일 노드의 시스템은 8개의 CPU로 구성되어 있으며, 8개의 CPU가 하나의 메모리를 공유하는 SMP 시스템이다. SX-5 시스템과 SX-6 시스템은 기가 네트워크로 서로 연결되어 있으며, SX-6 시스템 간은 서로 IXS의 고속채널로 연결되어 있어서 이 시스템간은 MPI 작업이 가능하다.

NEC 시스템의 접속은 Login 노드를 통해서만 접속할 수 있으며, 홈디렉토리는 Login 노드에 직접 연결되어 있고 다른 노드는 GFS를 통하여 공유되어 있다.

사용자의 대형 프로그램을 위하여 각각의 노드에 스크래치 디스크가 있으며, 각각의 노드에서 공유하여 사용할 수 있는 있는 스크래치 디스크가 별도로

설치되어 있다.

<표 1> KISTI의 NEC SX 시스템 사양

구분	내 용	
모델명	SX-5/8B	SX-6/16M * 2대
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 구조

NQS의 큐는 크게 파이프 큐와 배치 큐가 있다. 파이프 큐는 사용자가 보낸 작업을 조사하여 적당한 배치 큐에 할당하는 역할을 하는 큐이고, 배치 큐는 작업이 실제 실행되는 큐로서 배치 큐에 정의한 자원을 가지고 작업이 처리된다. 하나의 파이프 큐는 여러 개의 배치 큐를 가질 수 있다.

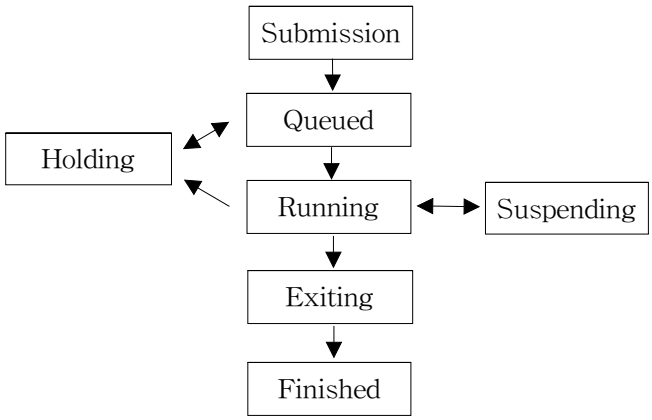
배치 큐에 작업을 할당하는 방법은 크게 두가지가 있으며, 하나는 파이프를 통하여 다시 배치 큐로 할당하는 방법이며, 다른 하나의 방법은 작업이 직접 배치 큐로 할당하는 방식이다.

2.3 NQS 처리 과정

다음 <그림 1>은 NQS에 작업이 보내져서 해당 작업이 종료되기까지의 과정을 나타내고 있다. 이러한 과정을 작업의 처리 단계로 보면 크게 2단계로 구분할 수 있다. 하나는 사용자가 보낸 작업이 어느 한 큐를 할당받기 전까지 대기하고 있는 대기시간(Waiting time)과 작업이 적당한 큐에 할당되어 해당 작업이 종료될 때까지의 경과시간(Elapsed time)이다.

NQS 설계를 위하여 각각의 큐 성능을 결정하기 위해서는 큐의 특성을 나타낼 수 있는 어떤 성능 비교 지수를 구해야 하는데, 다음과 같은 세가지의 시간을 사용한 관계식을 통하여 얻었으며, CPU 시간은 작업을 처리하는데 실제로 소요된 CPU time이다. 이것의 관계식은 다음과 같다.

$$\text{Queue factor} = (\text{Waiting} + \text{Elapsed}) / \text{CPU}$$



<그림 1> NQS 작업 처리 흐름도

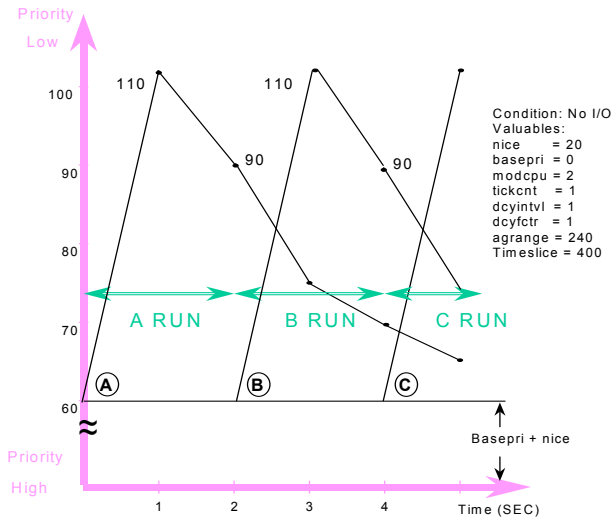
2.4 Processor Scheduling

시스템에서 프로세서가 처리되는 시스템의 스케줄링 방법은 <그림 2>와 같으며 각각의 프로세서에 대한 우선순위 결정은 아래의 식으로 계산되어 진다.

$$(\text{Execution priority}) = (\text{CPU counter}) >> (\text{modification value}) + (\text{nice value}) + (\text{base priority}) + (\text{constance})$$

다음의 <그림 2>은 세 개의 프로세서에 대하여 시간이 흐름에 따라 우선순위가 결정되고 결정된 우선순위에 의하여 프로세서가 활동하는 과정을 보여주고 있다.

여기서 보는 바와 같이 우선순위에 대한 주요 변수는 Priority와 Time slice이다.



<그림 2> Processor Scheduling Method

0 Second
Process A run
1 Second
A: (CPU counter)=200
(Execution pri)=200>>2+20+40=110
B: (Execution pri)=20+40=60
C: (Execution pri)=20+40=60
A:Sleep B:Run C:Sleep

2.5 NQS load balancing 방법

NQS의 load balancing 방법에는 다음과 같은 3가지가 있다.

- Round Robin
  가장 단순한 방법으로서 각 노드에 대하여 일정한 순서에 의하여 작업을 할당한다.
- Load Information Collection
  각 노드에 대한 CPU usage의 정보를 일정한 간격으로 참조하여 가장 사용율이 낮은 노드에 작업을 할당하다.
- Demand Delivery Method
  이 방법은 작업이 NQS에 보내지면 그 작업이 우선 Pipe 큐에 들어가서 적당한 Batch 큐를 찾아 할당하는 방법이다. 전체 노드에 대한 작업 수 등을 고려하여 load balancing을 가능하게 한다.

3. NQS Load balancing 설계

3.1 NQS 설계 시 고려사항

작업관리 시스템 설계 시 고려하여야 할 사항은 각각의 시스템의 성능과 특성 그리고 이들 시스템 간의 프로그램 호환성이 가능한지를 고려하여야 한다.

SX-5 시스템과 SX-6 시스템은 동일한 O/S를 사용하는 벡터 시스템으로서 프로그램의 특성에 따라 성능의 차이는 있지만 전반적으로 같은 성능을 나타내고 있으며, 사용자 프로그램이 서로 다른 노드에서 컴파일 되고 실행될 수 있도록 컴파일 환경을 구성하였다.

3.2 NQS Queue Factor

NQS 설계에 있어서 먼저 고려되어야 하는 부분은 시스템에서 처리할 작업의 종류와 특성을 분석하고 작업의 유형에 대한 큐의 종류를 분류하고 각각의 큐에 대하여 특성을 정의하였다.

큐의 종류는 우선 크게 4단계로 구분하였다. 그리고 각각의 큐의 종류에 대한 전체 실행 작업 수는 시스템에서 무리 없이 최대한 실행할 수 있는 전체 작업의 범위로 정했다. 시스템의 CPU usage 보다 작업의 수가 증가하면 swap이 빈번하게 발생하여 시스템 전체의 처리 효율이 저하되기 때문이다.

NQS 큐의 성능을 위하여 NQS의 많은 요소들을 가지고 테스트 하였다. 그 결과 앞에서 언급한 바와 같이 큐의 성능에 가장 큰 영향을 주는 변수로 Base priority와 Time slice라는 사실을 알 수 있었으며 Base priority는 우선순위를 결정하는 주요 변수로서 해당 프로세서의 활동에 가장 큰 영향을 주며, Time slice는 해당 프로세서가 실행되는 시간을 결정하기 때문에 이 두가지 변수를 이용하여 큐의 성능에 대한 주 설계 요소로 사용하였다.

다음의 <표 2>와 같이 다양한 큐를 만든 후 같은 프로그램을 여러개를 NQS에 넣어 시스템의 사용율이 100%가 되도록 하여 실행하였으며, 큐의 factor를 변경하면서 이같은 실험을 반복 수행하면서 그 중에 가장 적당한 함수 관계를 선택하여 얻을 수 있었다.

<표 2 > 서비스 레벨 큐 구성

큐 이름	CPU Time	서비스 레벨	Pri	Base Pri	Time Slice
dedicated	∞	2	30	60	2000
realtime	∞	2	30	60	2000
express	∞	1.5	30	70	500
large	∞	1	20	80	1000
small	180분	1	30	82	1000
normal	∞	1	29	80	1000
economy	∞	0.5	30	140	1000

3.3 NQS Queue 구성

NEC 시스템은 각각의 노드마다 큐 테이블을 가지고 있으며, 아래의 <표 3>은 그 중에서 SX-6b에 대한 실행 작업 수를 나타내고 있다. 하나의 노드는 8CPU로 구성되어 있으며 해당 노드에서 실행할 수 있는 작업의 최대 개수를 12개로 하였다. 이는 한 노드에서의 최대 적정 작업 수를 8개로 가정할 때 각각의 큐에서 최대로 실행 가능한 실행 작업 수는 2/3이내로 정하였다. 또 한 사용자가 여러 개의 작업을 수행할 경우가 있기 때문에 한 노드에서의 사용자 당 작업 수를 하나로 정의하여 그 외의 작업은

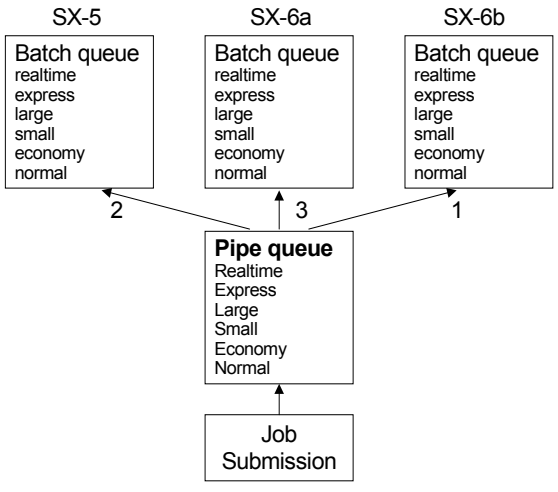
다른 노드로 넘어갈 수 있도록 고려하였다.

<표 3 >SX-6b NQS queue table

queue name	Pipe Only	Job limit			Complex (R,G,U)
		Run	Group	User	
dedicated	Y	∞	∞	∞	
realtime	Y	2	2	1	
express	Y	2	2	1	
large	Y	3	2	1	(6 4 2)
small	Y	2	2	1	
normal	Y	3	2	2	
economy	Y	2	2	1	

3.4 NQS Queue 할당 순서

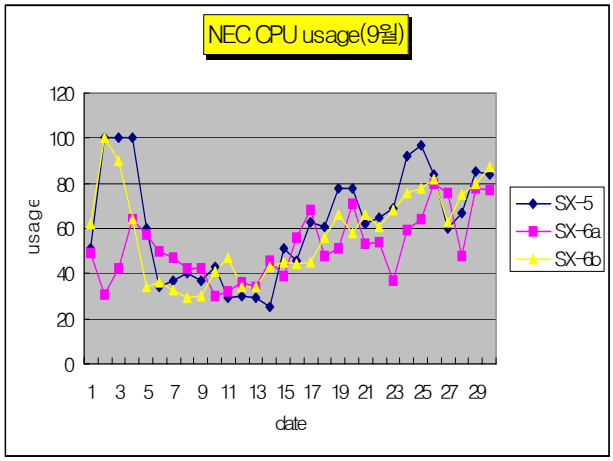
아래의 <그림 3>는 각 노드에 대한 파이프 큐와 배치 큐가 작업을 할당 받아 처리되는 과정을 보여주고 있다. load balancing의 구현에 있어서 고려하여할 사항은 작업을 할당할 노드를 찾는 순서이다. 이는 가장 먼저 찾는 노드에 많은 작업이 많이 할당될 수 있기 때문이다. Login 노드는 많은 서버가 위치할 수 있고, 사용자의 일반 명령어 작업이 여기에서 이루어지기 때문에 이러한 서버의 부하를 고려하여 작업의 할당이 맨 나중에 할당되도록 하여 전체 서버로서의 부하에 대한 역할을 고려하였다.



<그림 3> 노드간의 NQS 구성

4. 실행 결과

아래 <그림 4>는 각 노드별에 CPU Usage에 대한 1개월간의 통계이다. 그림에서 보는 바와 같이 CPU의 usage가 각각 약 80%일 때 그 성능이 가장 잘나타나는 것을 알 수 있었다. 이는 그동안 NQS에 대한 많은 통계 분석을 통하여 시스템의 CPU usage가 약 80% 이상부터 작업간 경쟁을 한다는 사실을 참고로 하였기 때문이다. 그 이하의 CPU usage에서는 각 노드간 load balancing이 정확히 이루어지지 않더라도 각 노드에서의 작업처리 효율에는 크게 영향을 받지 않는다.



<그림 4> 월별 CPU usage

5. 결론

단일 노드 내에서 설계된 NQS를 동시에 각각의 노드에 적용한 결과 노드간의 load balancing이 잘 이루어지는 것을 확인할 수 있었다. 이는 NQS 자체에서 이러한 기능을 제공하지 않기 때문에 수동적으로 노드를 찾는 순서와 큐의 실행 작업 수 등을 이용하여 해결할 수 있었다. 하지만 주기적으로 CPU 사용율의 변동과 사용자의 작업 패턴 등을 분석하여 큐를 알맞게 조정해 주어야하는 번거로움이 있을 수 있다. 향후에는 이러한 문제점을 시스템에서 자동적으로 해결할 수 설계하여 테스트하고자 한다.

참고문헌

[1] NEC, "SUPER-UX NQS User 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] NEC, "사용자지침서". KISTI 2003

[5] 이영주 외 "NEC 시스템에서 NQS 및 Load balancing 최적화" 정보처리학회 2003 추계학술지