HPC environment Design and Implementation for Efficient Job Processing in an NEC system

Ji Woo Choi†, Jung Oung Park†, Hyun Joung Kim†,
Jong Won Yang††, keyng Ju Ha†††, Chang Ho Seo†

ABSTRACT

As several users use a system simultaneously, a job management system is used to prevent exclusive uses in the system by a certain user and share entire resources for all users. The job management system is a type of scheduler that distributes limited resources for each job inquire efficiently. The system can be classified as different types and is to be properly selected according to the characteristics of systems and jobs. An NEC system is a job management system and usually applies NQS. The job management system has several queues, which are able to process jobs through distributed resources in which the efficiency of such job processing can be varied according to the composition of these queues.

This study designs and configures differentiated queues according to service levels in order to improve the efficiency of the entire system as NQS is designed and that performs job processing. Also, this study analyze the performance of the job processing.

Keywords : HPC Environment, NEC System

* This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea Government (MEST) (No. R01-2009-000-86148-0).

† 정 회 원: 공주대학교 일반대학원(바이오정보학과)
†† 정 회 원: 한국과학기술정보연구원
††† 정 회 원: 대구한의대학교 모바일콘텐츠학부 부교수
논문접수: 2010년 5월 3일, 심사완료: 2010년 5월 31일
* 본 논문은 2009년 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 연구되었음(No. R01-2009-000-86148-0).

1. Introduction

A job management system is a type of system utility that distributes and manages limited system resources efficiently according to the inquiry of several jobs. This system is to prevent exclusive uses in the system by a single user and distribute the resources inquired by several programs and process them. The job management system can be classified as different types and is to be properly selected according to the characteristics of systems and jobs. The job management system installed at the KISTI is the LoadLeveler by IBM, p690, and the NEC SX-5/6 system, SUN's SUN Grid Engine, PBS by the PC cluster, and NEC system[1][2] use NQS. Because a job management system processes the entire resources inquired by each job through distributing them, the type of job management and composition of queues directly affect the performance of job processing. There are several environmental variables, which are used to design and compose a job management system, in which CPU and memory significantly affect the system.

This study designed the job processing performance of batch queues as a differential manner from various methods used in the design and composition of job management systems in order to select the system according to the job condition of users. Then, this study analyzes the results of the application of each queue and tests its performance.

2. Related studies

2.1 Structure of NQS (Network Queuing System)

NQS can be largely classified as pipe queues and batch queues[3]. The pipe queues play a role in the distribution of jobs and implementation of jobs through distributed resources. As jobs are transmitted to the NQS, the jobs are to be allocated to batch queues through pipes. However, jobs can be directly allocated to the batch queues if the batch queues are not subordinated to a specific pipe queue. Pipe queues can be determined as multiple numbers, and jobs can be allocated as a proper queue is determined while the jobs directly transmitted by users are temporarily stayed. Batch queues can be configured as various types according to the characteristics of systems and job types and that are the execution queues, which perform jobs through distributed inquiry resources. Fig. 1 shows the structure of the NQS.

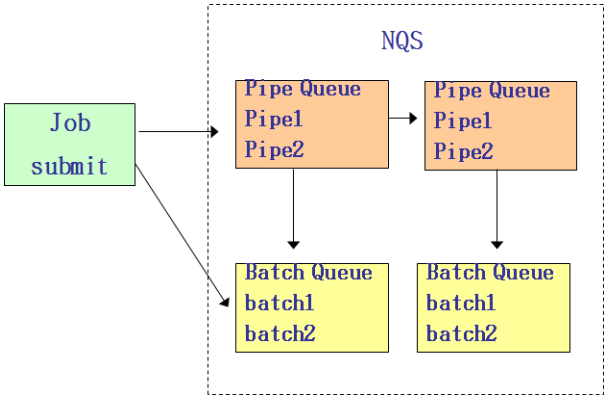

Fig. 1 Structure of the NQS

2.2 Process scheduling

When operating processes through Network Queuing System(NQS), there are three methods for process scheduling using pipe queue. The methods are as follows:

- Round Robin(RR): RR is one of the simplest scheduling method, which assigns time slices to each process in equal portions and in consecutive order, handling them without priority.

-Load Information Collection: Load Information Collection is the method which assigns the

process to the node which takes the lowest use ratio referring to CPU information.

- Demand Delivery Method(DDM): DDM is the method which assigns the process to the batch queue of each node in the pipe queue after the process is submitted to the pipe queue.

As above, the proper resource management method can be selected by analyzing the characteristic of the system and the pattern of the process.

There are several parameters which NQS considers to decide the priority for handling processes simultaneously in Table 1.

Table 1. Processing variables in the NQS process

Variable	Description	Basic Vale
Basepri	This variable is used by the system administrator or NQS administrator to adjust the execution priority	20
Modcpu	At the time of recalculation of the execution priority, the CPU counter is shifted with this value and added to the execution priority	2
Tickcnt	The CPU counter of a process being executed on the CPU is incremented by this value every tick	0
Dcyfatr	This value is used to perform the shift operation for the CPU counter every decay interva	1
Dcvintvi	The shift operation is performed for the CPU time with the decay factor each decay interval	1
Tmslice	The operation of the NQS can set the *timeslice*valueinunitsofjobsthatusethecharacteristicofschedulinggroupsinunitsofjobs	1000
Agrange	Tick quantum (tickcnt) is added to CPU counter until CPU counter becomes this value	160

As illustrated in Fig. 2, NQS determines the priority of processes according to a specific algorithm in which the priority can be determined by the results of the calculation performed by NQS variables, and then the process is to be implemented by the priority.

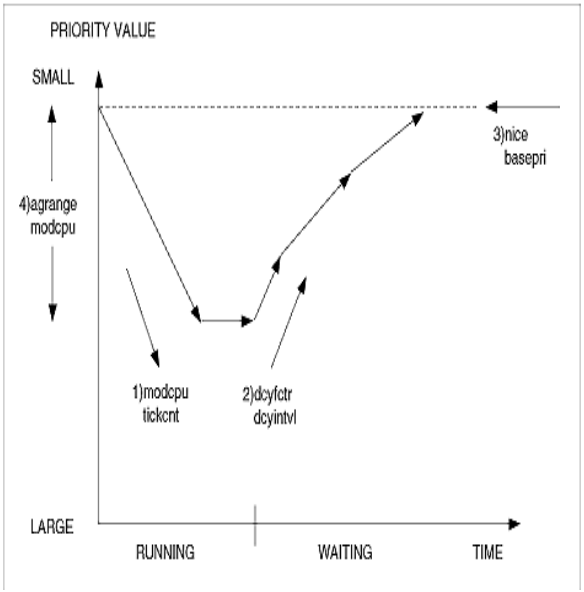

Fig. 2 Process of scheduling parameters

In a scheduling method, the determination of the priority, which performs a prior occupation in jobs, can be determined by the results of the calculation defined as follows.

(Executionpriority)=
(CPUcounter)>>(modificationvalue)+ (nicevalue) + (basepriority)+(constant)

Fig. 3 illustrates the procedure of the implementation of processes according to the determined priority after completing the priority of three given processes according to the passage of time.

In the factors that affect the calculation of the priority of processes, Baseprity and Timeslice most affect the priority.

0 Second: start
1 Second later
A: (CPU counter) = 200
(Execution priority) = 200 >> 2 + 20 + 40 = 110
B: (Execution priority) = 20 + 40 = 60
C: (Execution priority) = 20 + 40 = 60

2 Second later
A: (CPU counter) = 240 >> 120
(Execution priority) = 120 >> 2 + 20 + 40 = 90
B: (Execution priority) = 20 + 40 = 60
C: (Execution priority) = 20 + 40 = 60
A: Sleep B:Run, C:Sleep

3 Second later
A: (CPU counter) = 120 >> 1 = 60
(Execution priority) = 60 >> 2 + 20 + 40 = 75
B: (Execution priority) = 200
C: (Execution priority) = 20 + 40 = 60

4 Second later
A: (CPU counter) = 60 >> 1 = 30
(Execution priority) = 30 >> 2 + 20 + 40 = 68
B: (CPU counter) = 240 >> 1 = 120
(Execution priority) = 120 2 + 20 + 40 = 90
C: (Execution priority) = 20 + 40 = 60

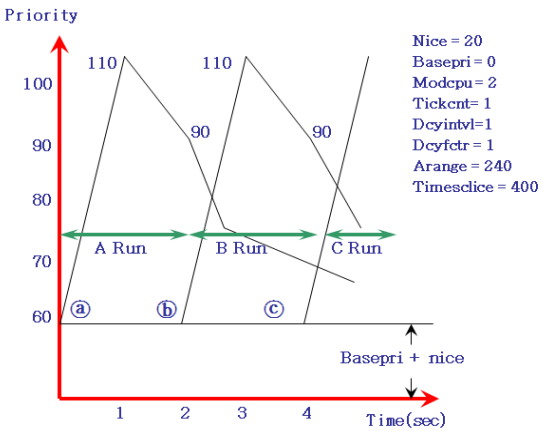

Fig. 3 Example of process allocation

In RR method, because of different processing time at each node, distribution of the process is difficult. Hence this method is rarely preferred in practical use. Although Load Information Collection method is the most suitable in theory, this can occur unfairness in the portion of processes because the process is always allocated to the node which has the lowest use ratio of CPU. In DDM, the process is allocated to the batch queue which is the most proper for that process, that is, the process the batch queue requires. Therefore, the process is not equally distributed when DDM is used.

3. Configuration of the system environment

To improve the problem of previous studies, this paper proposes more efficient NEC system.

3.1 System configuration

An NEC system consists of three different nodes, such as SX-5, SX-6a, and SX-6b, and these three nodes are connected through an EtherNet network in which a SX-6a plays a role in a server, and a home directory is shared as a single one[4][5][6].

Table 2. Job environment without excessive memory usage

Item	Description	
Model	SX-5	Two SX-6s
O.S	SUPER-UX R13	SUPER-UX R13
Number of CPU	8	16
Performance	80GFolps	160GFolps
Main Memory	128GB	128GB

3.2 Design of NQS queues

A basic direction in the design of batch queues is to provide a band of selection for users and lay a fee on the use of it according to differentiated queues. Thus, as presented in Table 4, the queues are classified as four different classes according to service levels. Users can select a proper queue for their jobs.

Because the fee is to be collected according to

the performance of job processing queues, a high level queue can be used in the case of urgent jobs. However, a low level queue is used to implement a regular job that may be processed without hurry. Thus, the job processing in a system can be implemented through distributing the entire resources efficiently due to the more allocations in jobs in the case of the job, which represents a higher priority than others while there are some competitions between the jobs.

Accounting = Time × Machine Charging Factor × Service Charging Factor

3.3 Test of NQS queues

A simple array program, which can be executed for 10 minutes, was prepared to test the performance of NQS queues. In the test system, there are some competitions between jobs by applying lots of jobs, which are enough to be executed in each queue, for the prepared program without any specific jobs. Then, the processing time for completing the given jobs in each queue.

In the job processing test, the test was applied by about 100 times by changing variables in each queue, and then similar 10 graphs were collected from the test trials (Fig. 4).

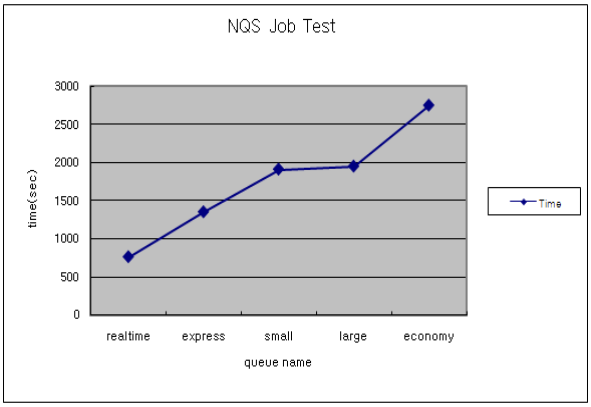

Fig. 4 Results of the execution test of batch queues

Table 3. Composition of batch queues

Name of Queue	CPU Time	Service Level	Pri	Base Pri	Time Slice	Aging Rang
dedicated	-	2	30	60	2000	160
realtime	-	2	30	60	2000	109
express	-	1.5	30	70	500	109
large	-	1	30	80	1000	109
small	180 min	1	29	82	1000	109
economy	-	0.5	30	140	1000	160

3.4 Composition of NQS

Fig. 5 illustrates the diagram of the entire system. As users transmit some jobs to the NQS, jobs are to be stayed in each pipe and can be allocated by searching a batch queue in a proper node. Here, a Load Information Collection method was used as a job distribution way in pipe queues.

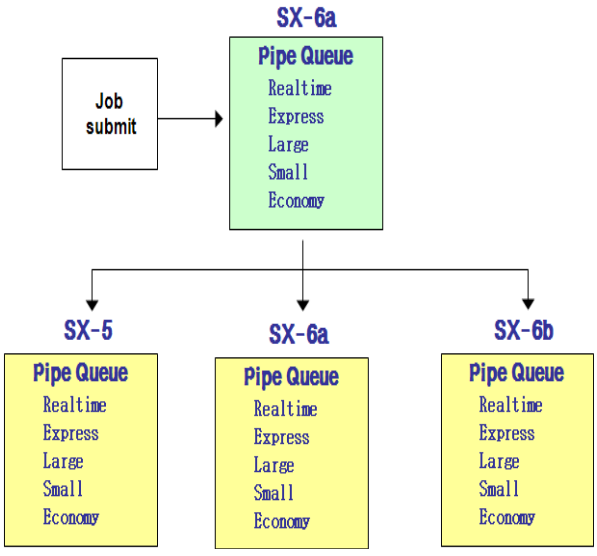

Fig. 5 Diagram of the NQS

4. Results

4.1 Analysis of job statistics

The statistics presented in Fig. 6 represent the results of the analysis of data through establishing an equation using following three factors based on the processing Log of the NQS. For instance, in the assumption that a single CPU is to be used as a single job is processed, the CPU processing time that is closed to that of elapsed job processing time shows a fast process in the CPU processing without any stops. Thus, the more approach to the processing factor of 1 represents the faster job processing.

* *Execution factor = Elapsed time / Cpu time*

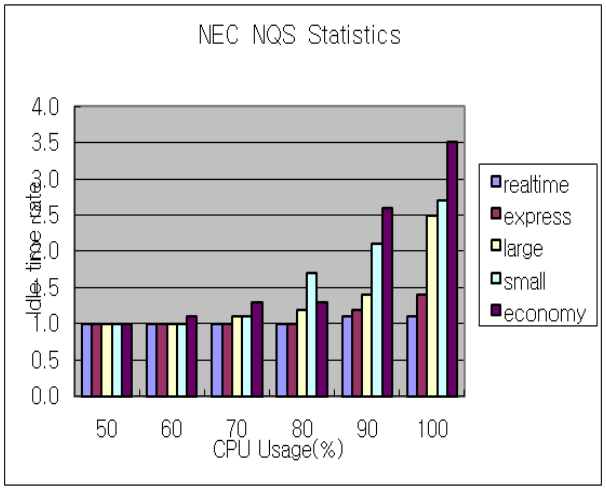

Fig. 6 Usage rates in each queue

5. Conclusion

In the results of the implementation of the job processing through the design of the NQS and test of the performance of each queue, it was verified that the job processing was properly processed according to the characteristics of each queue.

It also can be seen that the usage rate of the CPU is to be determined by more than 70% for applying the performance employed in each queue for the given jobs. It is due to the fact that the competitions between jobs are to be started at this moment through determining the priority of job according to the characteristics of queues.

Therefore, the differentiated queue design according to service levels in an NEC system represented the almost same results from the implementation of various jobs. In future studies, this study will focus on the distribution method of the node of jobs including the performance of the job processing in batch queues.

참 고 문 헌

[1] NEC, **"SUPER-UX NQS User Guide"**. NEC Corporation

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

[3] Young-Joo Lee etal, "Design and implementation of NQS Loadbalabcing in multi node". **Korea Information Processing Society Conference, 2004.**

[4] Young-Joo Lee etal, Design and implementation of Task management queue for efficient use of NEC system **Korea Information Processing Society Conference, 2005.**

[5] **Overfeeding Can Cause NEC-Like Enterocolitis in Premature Rat Pups.** Okada, Kyo ; Fujii, Tohru ; Ohtsuka, Yoshikazu Neonatology ,v.97 no.3 ,2010 ,pp.218-224 .

[6]**'Easy' print solutions The easyFairs' quartet of packaging exhibitions to be held in February at the NEC in the UK has a special 'print for packaging'.** zone this year Converting today ,v.24 no.2 ,2010 ,pp.19-21.

[7]2010/C 77/01 **Non-opposition to a notified concentration (Case COMP/M.5535 - Renesas Technology/NEC Electronics).**

Official journal of the European Union. Information and notices ,v.53 no.77 = no.77 ,2010 ,pp.1.

[8]**Strategic capacity decision-making in a stochastic manufacturing environment using real-time approximate dynamic programming.** Pratikakis, Nikolaos E. ; Realff, Matthew J. ; Lee, Jay H.Naval research logistics ,v.57 no.3 ,2010 ,pp.211-224

최 지 우

2009 공주대학교 응용수학(학사)
2009 ~ 현재 공주대학교 바이오정보학과 석사과정
관심분야: 암호 알고리즘, 융합기술보안, 컨텐츠보안
E-mail: liewhite@kongju.ac.kr

박 정 옥

2002 전남대학교 컴퓨터공학과(공학학사)
2002 광양제철 전로용강미니밀 공급전산시스템개발
2005 전남대학교 전자정보통신공학과(공학석사)
2007 ~ 현재 공주대학교 바이오정보학과 박사과정
관심분야: 암호 알고리즘, 네트워크보안, 컨텐츠보안
E-mail: JOPark@kongju.ac.kr

김 현 중

1998 성균관대 기계설계(공학학사)
2000 성균관대 기계설계(공학석사)
2000 ~ 2003 삼성전자 DA연구소 선임연구원
2007 ~ 현재 공주대학교 바이오정보학과 박사과정
관심분야: 기계설계, 암호 알고리즘, 컨텐츠보안
E-mail: HJKim@kongju.ac.kr

양 종 원

2003 공주대학교 전자계산학과(공학학사)
2005 공주대학교 컴퓨터공학과(공학석사)
2009 공주대학교 바이오정보학과(정보보호전공) 박사
관심분야: 융합기술보안, 컨텐츠보안
E-mail: nobody@kongju.ac.kr

하 경 주

1991 경북대학교 컴퓨터공학과 (공학사)
1993 경북대학교 컴퓨터공학과 (공학석사)
1996 경북대학교 컴퓨터공학과(공학박사)
1996 ~ 1999 한국전자통신기술연구원(ETRI) 부호기술연구부 선임연구원
1999 ~ 현재 대구한의대학교 모바일콘텐츠학부 부교수
관심분야: 정보보호, Visualcryptography, Steganography
E-mail: kjha@dhu.ac.kr

서 창 호

1990 고려대학교 수학과(학사)
1992 고려대학교 수학과(이학석사)
1996 고려대학교 수학과(이학박사)
1996 ~ 1996 국방과학연구소 선임연구원
1996 ~ 2000 한국전자통신연구원 선임연구원, 팀장
2000 ~ 현재 공주대학교 응용수학과(정보보호전공) 교수
2001 ~ 현재 공주대학교 바이오정보학과 교수
관심분야: 암호 알고리즘, 융합기술보안, 컨텐츠보안
E-mail: chseo@kongju.ac.kr