論文
5-7-9

웹 환경의 실시간 일괄처리 성능 분석 및 개선 방안 (The efficiency analysis and improvement plan of real time batch processing in web environment)

이선현(Sun-Hyun Lee)[1]

요 약

인터넷 기반 기술의 발전과 더불어 정보 기술을 응용한 애플리케이션 개발에 따라 C/S(Client/ Server) 환경에서 Web 환경으로 애플리케이션이 전환됨에 따라 사용자의 요구는 다양, 다변화되었다. 이에 사용자의 요구를 충족하다 보면, 실시간 일괄처리 성격의 업무들은 제한된 기존 시스템으로써는 감당하기 어려운 난관에 직면하게 된다. 이를 해결하기 위한 웹 환경의 인프라에 대해 정기적인 튜닝과 진단을 통하여 시스템에 대한 확장 내지는 새로운 시스템 도입에 결과적으로 도달하게 된다. 본 논문에서는 시스템의 성능을 분석하는 과정을 통하여 개선 방안과 시스템 도입 시 고려할 사항, 상관관계 및 개선 방안을 제안하고자 한다.

ABSTRACT

The demands of users have become variety and diversification by the conversion of application to Web environment from C/S(Client/Server) environment as well as by the development of application which puts into practice an information technology with the advance of internet basic technology. Therefore if it attempt to satisfy their all demands, the services like a real time batch processing are faced with a difficult obstacle that the limited existing system cannot manage.

Finally it is come to a conclusion that the extension of system or the induction of new system is necessary in order to solve this problem according to a regular tuning and an examination about infra of web environment. This thesis intends to propose about inter- relation and improvement plan as well as considerable facts in case of the introduction of system through the efficiency analysis process of the system.

논문접수 : 2004. 7. 5.
심사완료 : 2004. 7. 20.

1) 정회원 : 경기대학교 전산정보원 운영팀장

1. 서 론

인터넷 환경의 웹 기반 애플리케이션 업무들 중에는 실시간 일괄처리를 해야 하는 업무들이 증가하는 추세이다. 이에 국내 모든 대학들은 특히, 수강신청, 입학전형 업무 등과 같은 경우에는 민감하게 대응하고 있는 것이 현실정이다. 수강신청 업무는 실시간 일괄처리 업무의 대표적이라고 할 수 있는데 문제점으로는 다음과 같다.

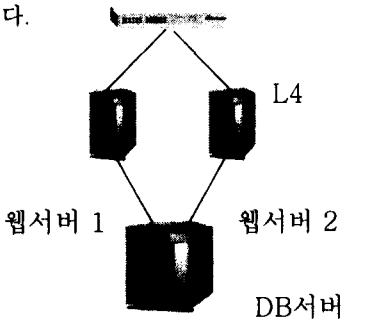
첫째, 제한된 전산 인프라를 이용하여 정해진 시간 내에 처리를 해야 하는 것과 둘째, 제한된 조건들(수강인원 제한, 부족한 강의실, 미확보 강사 등)에 의하여 전산처리를 하다보면 최초 시작 시점의 CPU 피크타임이 발생하게 되어 병목 현상이 일어난다. 셋째, 수강신청 업무는 학사관리의 모든 테이블을 Select함으로써 I/O측면에서 성능저하를 초래한다. 결과적으로 사용자들은 응답속도 지연에 따른 안정성과 신뢰성에 대해 의문을 제기하고 악순환은 학기마다 반복되는 것이다. 따라서 본 논문은 현재 보유하고 있는 인프라를 기반으로 시스템의 구성과 시스템 유틸리티, R/DB의 구성요소와의 관계, 애플리케이션 등을 진단하고 분석하여 문제점을 개선하는 방안을 연구한다.

2. 관련연구

본 장에서는 기존의 전산 인프라의 구성과 현재 수행하고 있는 WEB 서버에 탑재되어 있는 JEUS (Java Enterprise-User Solution)구조에서 수행 후 시스템 모니터링 분석, R/DB 진단 Tool을 이용한 분석, 튜닝을 통한 조치 결과, 개선해야 할 과제, 시스템 선정 시 고려해야 할 사항 등을 기술한다.

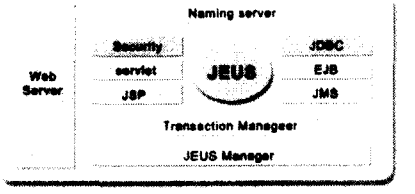
2.1 기존 전산 인프라 환경

기존 전산 인프라의 환경과 제한된 O/S의 자원 한도 내에서 최대한의 성능을 보장하기 위해서는 각각의 애플리케이션 특성에 따른 데이터베이스의 구축과 튜닝이 요구된다. [그림1]은 웹 서버인 UNIX 기반의 서버와 NT 기반의 서버를 L4 스위치 장비를 연동하여 부하를 분산하였고 DB 서버는 UNIX 기반으로 구성되어 있다.



[그림1] 실시간 일괄처리 서버 구성도

웹 서버 내에 웹 애플리케이션을 통해 서로 다른 O/S들을 연동하여 JSP, Servlet, EJB, JMS 등의 다양한 개발 모듈을 제공하고 있다.[1]



[그림2] Web Application Server 구성도

각 서버의 주요 환경으로는 <표1>에서 나타내었다.

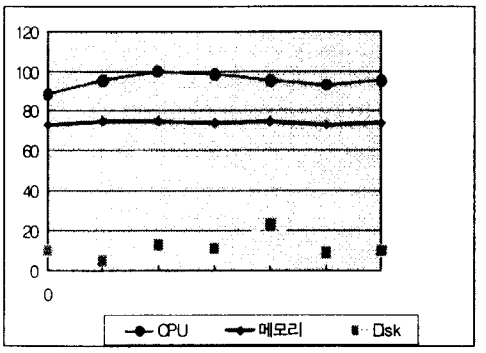
<표1> 각 서버의 주요 사양 환경

모 델 명	O/S	CPU	M/M	용 도
HP V/2500	UNIX	4CPU	4GB	DB서버
HP N/4000	UNIX	4CPU	4GB	웹서버
컴팩ML370	NT	2CPU	1GB	웹서버

DB서버의 R/DB는 오라클 8.0.5 버전을 사용하였고 수강신청업무는 오전 10시에 시작되었다.

2.2 시스템 모니터링에 의한 성능 분석

시스템 모니터링 분석 Tool은 HP사의 Glance -Plus c.02.40을 이용하여 수강신청 기간에 모니터링 한 결과 [그림3]과 같다.[2]



[그림3] HP/V2500 DB서버 사용률 현황

[그림3]에서 CPU, 디스크, 메모리에 대하여 시스템 전반에 걸친 분당 사용률 그래프를 나타내고 있으며 분당 CPU 최대 사용률은 10시 15분에서 10시30분 사이에 100%의 최고치를 기록하였고 memory 사용률 역시 평균 73%와 최고 75%로 나타났다. [그림4]는 전체 시스템 사용 현황 중 CPU별 각 사용량을 나타 낸 것으로 다음과 같다.

```
CPU  Util   SSSSSSSSSSSSSSSAAARUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU| 100%  100%  100%
Disk Util   FFFFFFFFF                                           |  18%   18%   18%
Mem  Util   SSSUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUBBBBB          |  75%   75%   75%
Swap Util   UUUURRRRRRRRRRRRRRRRR                               |  43%   43%   43%

 CPU   Util   LoadAvg(1/5/15 min)   CSwitch   Last Pid
   0   92.4    4.0/  4.3/  4.2       1651       8253
   1  100.0    3.2/  3.8/  3.7       1552       7963
   2   97.8    3.5/  3.9/  3.8       1694       7020
   3   89.6    4.0/  4.2/  4.0       1800       7958

                                                          CPU REPORT
State              Current     Cumulative      High

Load Average           3.1            4.4         6.1
Syscall Rate       56520.1        63005.5     95597.3
Intrpt Rate         2010.9         2138.7      3240.9
CSwitch Rate        1908.6         1773.8      2521.3

Top CPU user: PID  8079, oracleORAV         21.3% cpu util
```

[그림4] 각 CPU별 사용현황

결과적으로 System call Rate가 56,520의 수치가 발생하였는데 이상적인 기준치는 Processor당 3,000 시스템 콜이 적정수준이고 Load average도 3 이상이면 CPU의 병목현상이 발생한다고 본다.

메모리의 평균 사용률은 70 내지 80%가 적당하다고 본다. 따라서 시스템의 확장이나 새로운 시스템을 도입해야 된다는 결과가 나오지만 R/DB인 오라클에 대한 튜닝과 응용 프로그램의 테이블 등 부수적인 최적화 작업을 병행하지 않고서는 판단을 내리기가 어려운 실정이다.

2.3 데이터베이스 진단 Tool에 의한 성능 분석

안정된 성능을 나타내기 위해서는 데이터베이스의 튜닝은 필수적이며 튜닝에는 여러 가지 많은 부분이 있지만 일반적으로 메모리 튜닝, DB -WR, Checkpoint, Redo Log, Rollback Segment, SORT 등이 있다.

메모리 튜닝은 대표적으로 SGA (System Glo -bal Area) 튜닝을 말한다. 모든 작업은 이 SGA를 통하여 이루어지므로 최적의 상태를 유지해야 한다.

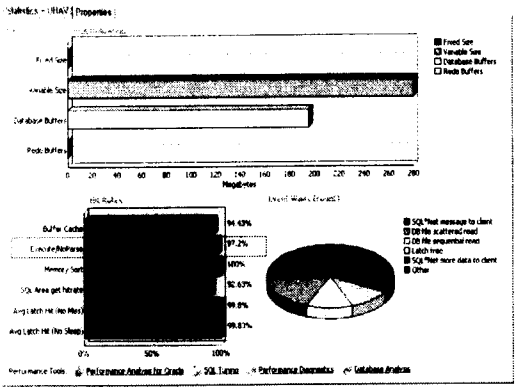
SGA는 데이터베이스 버퍼, shared pool, log buffer로 구성되어 있으며 각각의 구조별로 사용 용도가 다르며 또한, 데이터베이스 구조는 하부 구조인 운영체제와 이를 이용하는 애플리케이션과도 연결되어 있기 때문에 어느 한 분야의 단편적인 튜닝 보다는 O/S, 데이터베이스, 애플리케이션 등 시스템을 구성하고 있는 전 분야를 망라한 종합적인 관점에서 튜닝을 수행하여야만 극대의 효과를 얻을 수 있다. [3][4][5][6][7]

데이터베이스 진단 분석 Tool은 Quest Cent -ral for Database 4.0을 이용하여 Performance Analysis, SQL Turning, Spotlight, Database An -alysis의 과정을 진단 분석 한다.

분석 대상은 2.2절에서 표현한 바와 같이, 앞단의 웹 서버의 문제가 아니라 DB서버의 I/O와 CPU와의 관계이므로 DB서버로 한정한다.

[그림5]는 진단 Tool을 이용한 현재의 DB서버의 데이터베이스 상태의 SGA Utilization과

Hit Rate 상태를 나타내었다.[8]



[그림5] 데이터베이스의 SGA Utilization과 Hit Rate 상태

위의 내용을 자세히 설명하면 사용자가 SQL문을 사용했을 때, 데이터베이스는 Parsing → Fet -ch 단계를 거친다. Parsing은 SGA의 Shared Pool에 같은 SQL문이 있으면 Parsing을 하지 않으며 Parsing 할 때는 Shared pool latch, library cache latch의 자원을 사용해야 하는데 Shared pool latch는 1개이므로 shared pool에 같은 SQL이 있으면 latch를 사용하지 않는다. latch를 사용하지 않으면 CPU 사용이 감소하나 shared pool에 같은 SQL문이 존재하더라도 shared pool lat -ch는 사용하지 않지만 table 이름, 컬럼 이름을 조사하기 위해 library cache latch는 사용한다.

[그림5]에서 Execute/NoParse를 보면 loop내에서 같은 SQL문이 여러 번 실행된 것을 볼 수 있다. 이에 library cache latch wait이 발견되는 것을 알 수 있다. 다시 말하면 parse는 hard pars -ing, soft parsing, no parsing으로 구분되는데 hard parsing은 shared pool에 같은 SQL문이 없어 shared pool latch와 library cache latch를 사용하고 soft parsing은 shared pool에 같은 SQL문이 존재하므로 shared pool latch는 사용하지 않으나 library cache latch를 사용한다.

No parsing은 PGA영역에서 SQL문을 존재를 check 하므로 shared pool latch, library cache latch를 사용하지 않는다.

No parsing 사용 방법은 사용자가 데이터베이스에 접속할 때 서버 프로세서와 메모리(PGA)를 사용한다. PGA영역에 Private SQL Area를 사용할 수 있다.

Private SQL area를 사용하기 위해서는 세션을 접속한 후 Alter session set SESSION_CAC HED_CURSORS = 20을 사용하면 된다.

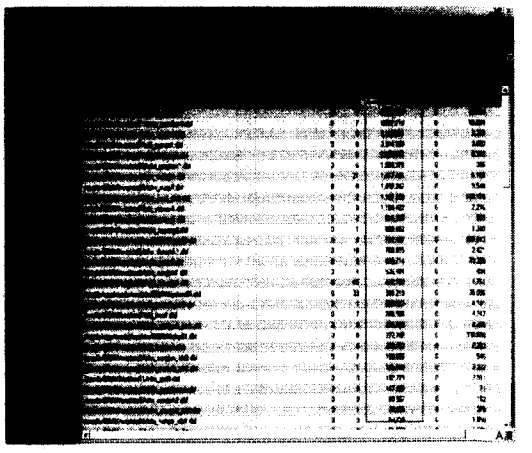
사용되는 SQL문의 결과를 보면 Active time의 증가 원인은 latch wait 때문임을 알 수 있다.

실제 CPU 사용시간은 1초미만 → SQL 튜닝 보다는 latch 튜닝을 해야 됨을 알게 된다.

PGA영역에서 같은 SQL이 있으면 SGA (shar -ed pool)영역으로 가지 않고 PGA영역에서 pars -ing 일어난다. 위 그림에서 원 그래프는 Physi -cal read가 많이 일어남을 볼 수 있다.

Physical read는 full table scan과 buffer cache size가 작아서 발생한다. 기타 여러 가지 이유가 있으나 Active time이 많은 SQL문을 조사 한 결과 Buffer cache size를 증가하면 된다.

[그림6]은 수강신청 업무를 처리 시에 특정 DB F 파일 I/O가 많이 일어나는 상태를 나타내었다. 따라서 Table Space 분리가 요구된다.[8]



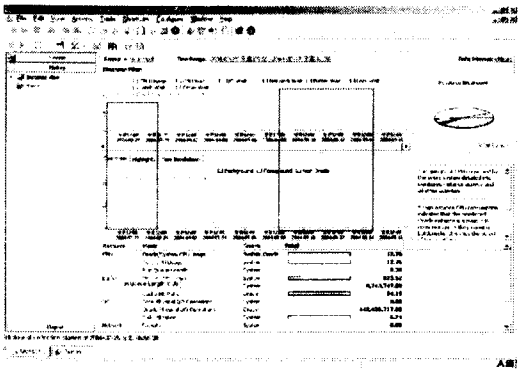
[그림6] 특정 DBF 파일 상태

데이터베이스 내의 상태분석 한 결과를 요약하면 Execute/NoParse의 수치가 97.2%는

Parse Request가 높은 현상은 library cache latch, shar -ed pool latch 의 wait로 인한 CPU 증가가 원인이며 해결 방안으로는 Execute count 과다 세션은 최소화하도록 logic을 점검하고 Alter sess -ion set SESSION_CACHED_CURSORS = 20을 사용하여 Literal SQL → Bind SQL로 수정하고 CURSOR_SHARING = FORCE로 조정한다.

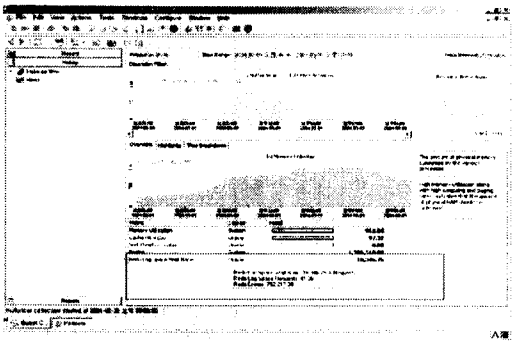
Physical I/O가 높은 이유는 DB file sequential (scattered) read wait event가 발생하기 때문이고 Latch free Event인 library cache latch, cac -he buffer chain latch가 원인이므로 해결책으로는 SGA를 증가시키면 된다. 즉, data buffer cac -he 증가시키면 된다. SGA 사이즈를 현재 476 MB에서 1500MB로 증가시키고 특정 FILE I/O가 집중되는 것을 방지하기 위해서는 DBF 파일을 분리하는 것이 해결 방안이다.

[그림7]은 단기간에 데이터베이스 진단 Tool에 의한 Log file을 수집하여 실시간 일괄처리 작업인 수강 신청 시점과 수강 신청 시점이 아닌 상태에 대해 전체 성능분석을 한 결과를 나타낸다. 수강 신청 처리 시점에는 시스템의 CPU Wait와 Latch Wait가 발생하고 일상적일 때에는 시스템의 정상 상태를 나타내고 Wait Events에 대한 진단 결과로 CPU Wait는 Latch Wait임을 알 수 있으며 Latch Wait의 원인은 library cache, cache buffers chains latch임을 알 수 있다.[8] 빨간 표시가 수강신청 기간임을 나타낸다.



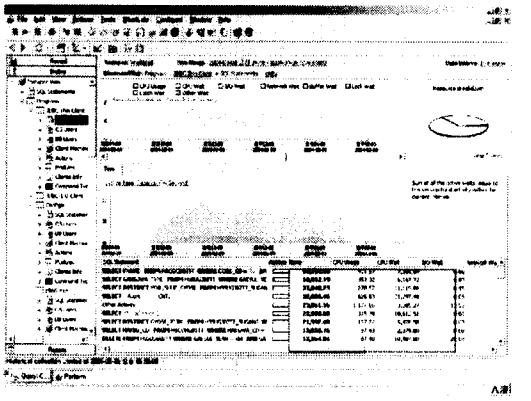
[그림7] CPU & Latch Wait 전체 성능분석

[그림8]은 Buffer의 상태를 진단한 것으로 redo log buffer 사이즈가 너무 작음을 알 수 있으며 log buffer에서 Disk(redo log file)로 내리는 작업 보다 log buffer에서 사용되는 작업이 많다는 것 으로 예상되며 해결책으로는 Log Buffer를 현재 64KB에서 512KB로 증가시켜야 한다.[8]



[그림 8] Redo Log Buffer 상태 분석

[그림9]는 프로그램과 JDBC Thin Client의 관계를 진단한 것으로 JDBC Thin Client 프로그램을 보면 앞에서 설명한 것처럼 CPU Wait는 Lat -ch Wait가 높기 때문이며 SQL문의 Active타임 이 매우 높은 이유로는 SQL문이 실제 실행 된 시간보다 CPU Usage가 훨씬 높기 때문이다.[8]



[그림9] 프로그램과 JDBC Thin Client 관계

결과적으로 SQL문 튜닝보다는 Latch Wait을

튜닝을 해야 되며 Library cache latch 튜닝인 SESSION_CACHED_CURSOR작업을 진행해야 된다. 또한, 데이터베이스의 파라미터 값을 변경하고 CPU 개수를 증가시키고 수강신청 관련 테이블을 분리하여 재조정해야 할 필요가 있다.

2.4 진단 성능 분석 후 조치 결과

시스템 및 데이터베이스의 진단 성능 분석 후 각 웹 서버와 DB서버에 대해 조치한 결과를 <표2>를 통해 나타내었다.

<표2> 각 서버 진단 성능 분석 후 조치 내용

Category (HP/V2500)	Before	After
SGA	476MB	1.32G
shared_pool_size	250MB	450MB
db_block_buffers	100000	400000
log_buffer	30720	1M
PGA(process한개당)		
cursor_space_for_time	FALSE	TRUE
session_cached_cursors	0	20
db_block_lru_latches	4	8
Category (HP/N4000)	Before	After
shared_pool_size	14MB	250MB
db_block_buffers	2048	50000
log_buffer	152KB	1MB
Category (컴팩/ML370)	Before	After
db_cache_size	35MB	350MB
pga_aggregate_target	25MB	125MB
dispatchers		주석처리
shared_pool_size	35MB	350MB

<표2>와 같이 진단 성능 분석후의 문제점을 개선하여 상당히 많은 효과를 보았으나 어디까지나 소프트웨어적인 해결이 될 수밖에 없으며 미지않아 하드웨어적으로 해결해야만 상황에 도달 한다.

3. 고려해야 할 사항

본 장에서는 기존 인프라 환경에서 새로운 서버 도입 시 고려해야 할 소프트웨어적인 밀접한 관계와 서버 선정 시 고려해야 할 사항들에 기술하고자 한다.

3.1 소프트웨어적인 관점에서의 고려 사항

새로운 서버를 도입하다 보면 일반적으로 응용 애플리케이션을 새로 개발해야 하지만 응용 애플리케이션을 새로 개발하지 않을 경우에는 아래와 같은 상관관계의 소프트웨어적인 관점에서 고려해야 할 문제점이 발생하게 된다.

- ✔ 기존에 사용 중인 서버의 O/S 버전과 도입되는 서버 간의 O/S의 버전
- ✔ 기존에 사용 중인 데이터베이스의 버전과 도입되는 서버 간의 데이터베이스의 버전
- ✔ 중간 미들웨어의 엔진 버전
- ✔ 현재 개발하여 사용 중인 애플리케이션의 개발 Tool 버전
- ✔ 사용자의 개인용 컴퓨터의 환경
- ✔ O/S 버전과 데이터베이스의 버전의 관계
- ✔ O/S 버전과 데이터베이스 버전과 미들웨어 엔진의 버전 관계
- ✔ O/S 버전과 데이터베이스 버전과 미들웨어 엔진의 버전 및 애플리케이션 개발 Tool의 버전 관계
- ✔ 기 개발된 프로그램의 활용 내지는 컨버전 작업을 해야 할지 말아야 할지 하는 의사결정 등

위의 내용들에 대해 전산 실무자라면 반드시 경험하게 되는 과정이며, 지금과 같은 IT 분야의 신기술이 새롭게 등장하고 응용 애플리케이션의 Life Cycle이 점점 짧아지고 있는 상황에서는 고민을 해보아야 하는 상황에 도달하게 되었다.

일반적으로 Data 컨버전으로 해결되는 문제가 아니며 과거와는 다르게 Data 컨버전 비용 역시 상당한 대가를 지불해야 하는 상황이 되었다.

3.2 하드웨어적인 관점에서의 고려 사항

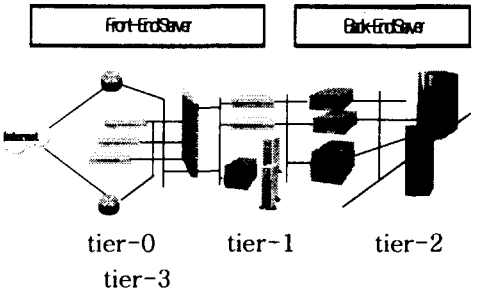
최근에 와서는 서버들에 대한 새로운 정의가 하드웨어적 관점에서 새롭게 정의가 되어 있다.

서버에 대한 정의는 Enterprise Architecture라는 명칭 하에 Front-End

Server, Back-End Ser -ver로 분류되고 Front-End Server에는 Edge Server(Tier-0), Web Server(Tier-1)로 구분되며 Back-End Server에는 Application Server와 Da -ta/Content/Warehouse 로 나뉘어 진다.[9]

- ✔ Tier-0서버 : Firewall, Proxy, VPN, WAP
- ✔ Tier-1서버 : Web Server 등
- ✔ Tier-2서버 : ERP, SCM, CRM 등
- ✔ Tier-3서버 : Database Server 등으로 표현된다.

위의 내용을 [그림10]으로 표현하면 다음과 같다.



[그림10] Enterprise Architecture 구조도

Front-End Server 도입 시 고려할 사항으로는 용도에 맞게 정적 또는 동적인 웹 페이지를 사용자 요청에 맞게 보여주는 역할을 해야 한다. 따라서 여러 대의 엔트리급 서버를 L4 스위치 장비로 연동하여 부하를 조정해야하고 증설을 할 경우에는 서버의 대수를 늘리는 정책을 사용해야 한다.

Back-End Server 도입 시에는 주로 웹서버나 애플리케이션 서버로부터 요청을 처리해 주는 역할을 하므로 데이터베이스 Data의 쿼리 실행 또는 Data Warehouse, Data Mining 등을 위한 데이터 분석 등을 위해 많은 계산 수행 능력이 필요하다. 따라서 Enterprise급의 High Back-End Server가 요구되며, 증설을 할 경우에는 서버의 스팩을 늘리거나 상위 기종으로 바꾸는 정책을 사용해야 한다. 민감한 업무에 적용하기 위해서는 이중화 구성도 고려해 볼 만 하다.

서버 스팩 산정에 있어서는 크게 Front-Tier Sizing, Middle-Tier Sizing, Back-End-Tier Siz -ing으로 분류하며 <표3>,<표4>,<표5>를 통하여 산정 기준을 나타낸다.[9]

<표3> Front-Tier Sizing 산정 기준

SPEC	항 목	단위
CPU	최대 동시 사용자	명
	평균 사용자가 조회하는 평균 Page 수(1회 방문 기준)	Pages
	Page 당 Frame 수	Fr/P
	최대 동시접속 사용자가 요청하는 웹 페이지를 보여주기 위해 필요한 서버 부하 고려하여 CPU 산정	
M/M	O/S 커널 메모리 점유량	MB
	IIS, Apache 등 웹서버 프로그램의 메모리 점유량	MB
	1Page가 차지하는 평균크기	MB
	O/S, Disk 캐쉬 등 기본필요량 + 웹 서비스를 위해 필요한 메모리 공간	
DISK	O/S HDD 사용량	GB
	메모리 Swap영역	GB
	웹 Data HDD 사용량	GB
	O/S, AP, Swap 등 기본필요량 + 웹 Page Data 필요용량	

<표4> Middle-Tier Sizing 산정 기준

SPEC	항 목	단위
CPU	최대 동시 사용자	명/분
	AP서버로의 트랜젝션을 발생시키는 Page 조회수 (1인평균 접속수 기준)	Pages /명
	1 Page 당 트랜젝션 수	Tr/P

SPEC	항 목	단위
CPU	최대 동시접속 사용자가 요청하는 작업에서 AP서버가 트랜젝션을 처리하기 위한 적정 CPU 선정	
M/M	O/S 커널 메모리 점유량	MB
	IIS, Apache 등 웹서버 프로그램의 메모리 점유량	MB
	1Page가 차지하는 평균크기	MB
	O/S, Disk 캐쉬 등 기본필요량 + 웹 서비스를 위해 필요한 메모리 공간	
DISK	O/S HDD 사용량	GB
	메모리 Swap영역	GB
	웹 Data HDD 사용량	GB
	O/S, AP, Swap 등 기본필요량 + 웹 Page Data 필요용량	

<표5> Back-End-Tier Sizing 산정 기준

SPEC	항 목	단위
CPU	최대 동시 사용자	명
	분당 1인 평균 DB서버 Request 발생량	Re/분
	조회, 입력,업데이트 발생량에 따른 Request 당 트랜젝션 예상량	Tr/R
	최대 동시접속 사용자가 요청하는 작업에서 DB 서버가 트랜젝션을 처리하기 위한 적정 CPU 선정	
M/M	O/S 커널 메모리 점유량	MB
	데이터베이스 등 DB 엔진의 메모리 점유량	MB
	OLAP 툴 메모리 점유량	MB
	1인 메모리 사용량	MB
	O/S, Disk 캐쉬 등 기본필요량 + DB 엔진 및 Tool 필요량 + 동시 사용자당 필요한 메모리 공간	
DISK	O/S HDD 사용량	GB
	메모리 Swap영역	GB
	DB, OLAP 툴 HDD 사용량	GB
	O/S, AP, Swap 등 기본필요량 + DB Data 필요용량	

기타 부수적으로는 향후 업무 증가 및 사용자 수 증가를 고려한 여유분에 대한 보정치를 적용하여 최종 웹 서버 스팩을 선정해야 한다.

4. 결론 및 향후 연구 방안

본 논문은 웹 기반의 실시간 일괄처리 성능 분석 및 개선 방안에 있어서 시스템 모니터링과 DB 진단 Tool을 이용한 성능 분석을 통해 현 상황의 최적의 개선 방안을 제시하여 개선의 효과를 보았지만 늘어나는 웹 사용자의 수와 다양한 요구 사항과 늘어나는 업무를 고려해 볼 때, 향후 서버의 증설이나 도입이 요구되어진다.

서버의 도입 시에 고려할 사항과 소프트웨어적인 관점에서의 고려할 사항을 바탕으로 항상 제한된 인프라의 환경을 효율적으로 관리하기 위해서는 정해진 시점에 시스템에 대한 진단과 데이터베이스 튜닝을 정기적으로 해 줄 필요가 있다.

향후에 해결해야 할 연구 과제로는 전반적으로 발생하는 CPU Wait와 Latch Wait가 나타나는 특정 업무들에 대한 DBF FILE Table을 재정립해야하는 과정과 재정립된 상태에서 시스템에 대한 재정의가 요구 되어진다.

최근의 IT 시장에서는 슈퍼 돔과 같이 Multi O/S를 탑재하여 CPU와 Memory의 사이즈에 관계없이 그 시점에 발생되는 업무의 정도에 따라 사이즈를 자유롭게 조절하여 처리하고 있는 실정이며, 실시간 일괄처리 업무인 응용 애플리케이션에 대해 닷 넷 기반으로 개발하여 여러 대의 서버를 연동하여 동시접속 사용자들의 부하를 분산시킴과 동시에 응답 속도를 최적의 상태로 처리해 주는 기법들이 등장하고 있다.

1. 참 고 문 헌
2.

[1] AP(Web Application Server) 중 제우스의 개념 및 구조 [Online]
http://www.tmax.co.kr/product_jeus2.htm
[2] HP사의 GlancePlus c.02.40 시스템 성능분석 Tool
[3] 김형일(2002). 오라클 데이터베이스 성능관리
(2) 중 데이터베이스 튜닝 및 스페이스 메

니지먼트, 웨어벡리, pp. 1-17
[4] Oracle Server Tuning
[5] Oracle Concept Manual
[6] Oracle Server Administration
[7] Oracle 8i & Unix Performance Tuning
[8] Quest Central for Database 4.0 성능분석 Tool
[9] 삼성전자(2004). 낭비가 없는 최적 규모의 하드웨어 도입방안, 대학 정보화를 위한 종합정보시스템 구축 세미나, pp. 1-14

이 선 현

1988년 경기대학교 전자계산학과 졸업(이학사)
2000년 경기대학교 정보통신대학원 정보통신전공 졸업(이학석사)
2002년 ~ 현재 경기대학교 대학원 정보과학부 전자계산학과 박사과정
1988년 중앙교육진흥연구소 정보시스템실 근무
1990년 ~ 현재 경기대학교 전산정보원 운영팀장
관심분야 : 유비쿼터스 컴퓨팅, 유비쿼터스 네트워크