

ERP 시스템 성능 테스트 방법 연구

신현종*, 김경현*

*LG CNS 테스트팀

A Study on Methods of Performance Testing for Enterprise Resource Planning System

Shin, Hyun Jong*, Kim, Kyung Hyun

LG CNS Test Team

E-mail : hjongshin@lgcns.com, kimkhyun@lgcns.com

요 약

ERP 시스템은 전사 통합 관점에서 기업 전체의 업무를 관리하기 때문에, 업무 집중도가 높고, 처리량이 많으며, 업무가 Waterfall 모델로 설계된 부분이 많은 특성이 있어, 어느 한 부분의 성능 문제가 전체 시스템 운영의 중대한 Issue로 확대될 가능성이 크다. 본 논문을 통해서 ERP 시스템의 성능 Issue를 사전에 점검할 수 있는 성능 테스트(Performance Testing) 방법을 연구했고, 실제 ERP 시스템 구축 Project에 투입되어 성능 테스트를 진행한 사례를 고찰하여, 발견된 Issue를 토대로 ERP 시스템의 성능을 최적화하고 시스템을 안정적으로 운영할 수 있는 방안을 연구해 보았다.

1. 서론

ERP 시스템은 기업 내 Process 전반을 하나의 체계로 연계해서 관리하고, 정보를 공유함으로써 전사적인 업무 효율을 극대화할 수 있는 장점이 있는 반면, 구축하려는 시스템의 규모가 매우 크고, 업무 처리량, 부하 집중도가 무척 높은 특성이 있기 때문에 많은 Risk를 가진 시스템이다. 특히 기업 업무의 특성상 마감, 결산, 보고 등은 특정한 짧은 기간 동안 처리량이 집중되는데 이때 시스템은 Transaction 폭주를 원활히 처리할 수 있어야만 한다. 따라서 ERP 시스템의 안정적인 운영을 위해서는 반드시 성능 관련 Issue를 점검해야만 한다. 그러나 ERP 시스템 구축 시 성능

테스트 수행 방법을 알지 못해 생략하거나 올바르게 수행하지 못하는 사례가 많다. 특히 동일 Package를 적용해서 구축한 시스템에 대해 성능 시험을 수행한 결과, 발견한 성능 Issue를 공유하는 사례는 거의 찾아 볼 수 없고, System Test의 일부로서의 성능 테스트의 단편적인 용어 설명이나 절차 소개 외에는 깊이 있는 연구가 선행된 전례가 거의 없는 실정이다. 따라서 본 논문을 통해서 ERP 시스템의 안정적인 운영을 검증할 수 있는 방안으로, 성능 테스트 방법에 대해 Focus를 집중하여 고찰해 보고, 실제 ERP 시스템 구축 Project의 성능 테스트 업무에 투입되어 정립한 Process, Action, Metrics, Tool, Issue, Lesson

Learned에 관련된 Know-how를 정리하여 ERP 시스템에 대한 성능 테스트 표준 Guideline을 도출해 보고자 했다.

2. ERP System/Package/Module 현황

2.1 ERP System

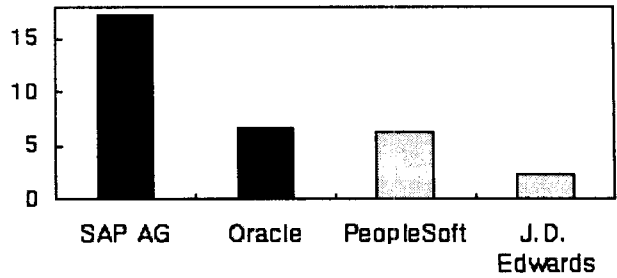
ERP 시스템은 기업 활동을 위해서 사용되고 있는 기업내의 모든 인적, 물적 자원을 효율적으로 관리하여 궁극적으로는 기업의 경쟁력을 강화시키는 역할을 하게 되는 통합정보시스템이라고 할 수 있다. 최신 정보 기술을 이용하여 수주, 제조, 출하, 회계, 물류, 재무 및 인사를 포함한 전사적 자원의 활용을 정의하고, 계획하며, 생성된 정보를 통합한 Database에 보관하여 프로세스를 표준화하고, 품질관리를 일원화하며, 성과분석을 손쉽게 수행할 수 있기 때문에 기업의 업무를 최적화하는 Tool로써 널리 사용되고 있다.

2.2 ERP Package

각 기업의 특수성을 고려해도 공통적으로 처리해야 할 업무가 있는데, 이런 공통의 업무를 매번 새롭게 개발하는 것은 구축 기간이나 비용 면에서 효율적이지 못하다. 이런 가운데 특정 Software 개발 업체에서 ERP 시스템을 반복적으로 구축한 Solution과 Know-how를 집적하여 상품화한 ERP Package를 선보였다. ERP 시스템을 구축하는데 드는 비용, 기간, 위험 요소 등이 감소하게 됨에 따라 순수한 ERP 시스템 개발보다는 ERP Package를 도입하여 수정이 필요 없는 부분은 Standard 모듈을 그대로 사용하고 기업의 특성에 맞도록 변경할 필요성이 있는 부분은 Customized 또는 원하는 기능을 Bolt On하여 ERP 시스템을 구축하는 것이 일반화되었다.

SAP사 및 Oracle사 외에, People Soft, JDE 사 등에서도 ERP Package를 출시하였으며, 2001년 ERP Package 시장의 매출액을 기준으로 시장 점유율을 조사한 결과 SAP이 17.2%(4,066 M\$)로 1위인 것으로 나타났고, Oracle이

6.7%(1,585M\$)로 2위인 것으로 나타났다.



<그림 1> 2001년 매출액 기준 ERP 시장 점유율

2.3 ERP Module

초기 ERP 시스템은 기본적으로 C/S 환경을 지원했는데 User Computer 환경이 PC를 기반으로 한 Web 환경으로 전환되자 C/S 환경과 더불어 Web 환경을 지원하도록 변화했다. SAP의 경우에는 기존의 SAP R/3 외에 Web 환경을 지원하는 mySAP.com을 선보였고, Oracle의 경우 개발 환경이 CORBA(Common Object Request Broker Architecture) JAVA에 근간을 두었던 만큼 Web 환경을 통해 바로 JAVA Applet Program을 사용하도록 지원한다.

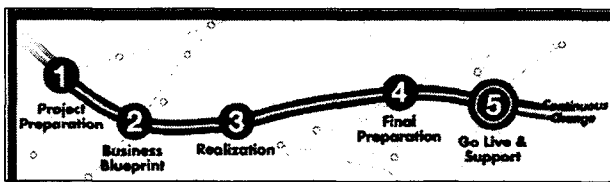
SAP 및 Oracle ERP Package는 기업의 업무별로 지원하는 Program을 Module이란 개념을 사용하여 구분한다. SAP은 판매관리(SD : Sales & Distribution), 자재관리(MM : Material Management), 생산관리(PP : Production Planning), 품질관리(QM : Quality Management), 설비관리(PM : Plant Management), 프로젝트 관리(PS : Project System), 인사관리(HR : Human Resource), 재무회계(FI : Financial), 관리회계(CO : Controlling), 자금관리(TR : Treasury), 투자관리(IM : Investment Management), 부동산 관리(RE : Real Estate Management) 등의 Module로 구성되어 있으며, Oracle은 총계정원장(GL : General Ledger), 미지급금관리(AP : Account Payables), 미수금관리(AR : Account Receivable), 고정자산 관리(FA : Fixed Assets), 현금 관리(CM : Cash Management), 재고관리(INV : Inventory), 부품 관리(BOM : Bill of

Material), 공정관리(WIP : Work In Process), 원가관리(COST : Costing), 구매관리(PO : Purchasing) 등의 Module로 구성되어 있다.

3. ERP 구축 Project의 성능 점검방안

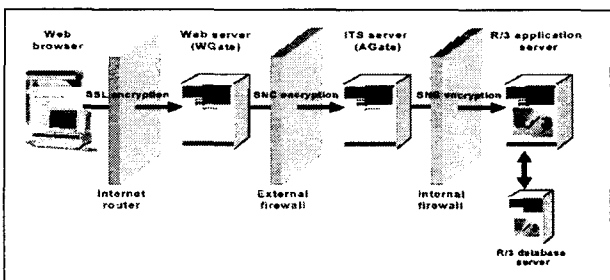
3.1 SAP 개발 방법론

SAP ERP 시스템은 전통적인 DW 개발 방법론을 수용하면서 SAP의 BW 구축 Know-how가 접목된 ASAP for BW라는 개발 방법론을 적용하여 개발할 수 있다. ASAP 방법론은 ① Project Preparation, ② Business Blueprint, ③ Realization ④ Final Preparation, ⑤ Go Live & Support와 같은 Process로 구성되어 있는데, Business Blueprint 단계에서 To-Be 모델 설계 시, 미래의 시스템 사용량 증가를 고려하여 Architecture를 설계할 필요성이 있다. 더불어 Go Live & Support 단계에서 성능 시험을 통해 시스템의 성능을 측정하고 성능 Issue가 발견되었을 경우 System Tuning을 실시하도록 하여 System 성능 관련 품질을 점검할 수 있는 Process를 마련해 놓았다.



<그림 2> SAP ERP 개발 방법론 ASAP의 Roadmap

3.2 SAP Architecture 특성

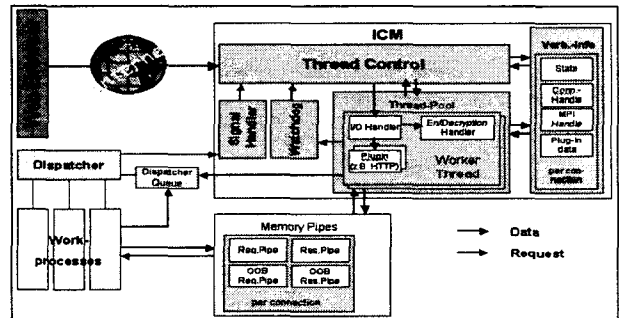


<그림 3> SAP Layer 계층 구조(Version 6.20 미만)

대부분의 ERP System은 DB ↔ Application ↔ Client로 구성된 3 Tiers Architecture로 구성되어 있다. 이를 기반으로 해서 부하 분산(Load

Balancing)을 위해서 L4 Switch 장비를 사용하는 경우, System의 가용성을 보장하기 위해서 HA(High Availability) 혹은 RAC(Real Application Cluster) 환경을 사용하는 경우도 증가하고 있다.

SAP의 경우 Web 환경을 지원하기 위해서 mySAP.com란 Internet Base Solution을 가지고 있는데, Version 6.2 미만에서는 ITS(Internet Transaction Server)란 일종의 Middleware를 사용해서 HTTP 환경에서 발생한 Request를 SAP R/3로 변환하여 처리할 수 있도록 시스템 Architecture를 구성했으나 Version 6.2 이상에서는 SAP WAS(Web Application Server) 안에 ICM(Internet Communication Manager) 및 ICF(Internet Communication Framework) Component를 내장하여 자체적으로 Internet Base Request를 처리할 수 있도록 Application 관련 Architecture를 변경했다.



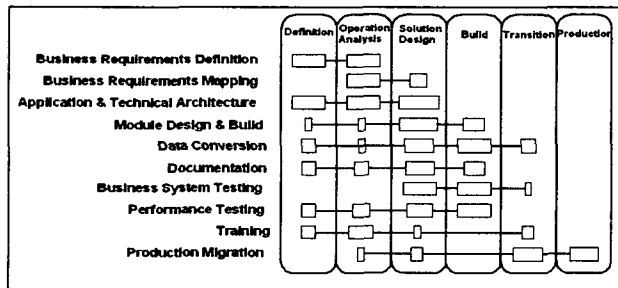
<그림 4> SAP WAS 구조(Version 6.20 이상)

3.3 Oracle ERP 개발 방법론

Oracle ERP 개발 방법론인 AIM(Application Implementation Method)은 Application Package를 최적으로 구현하기 위한 방법으로, Project 진행 초기에 Business 요건을 정의하고 전체 구현 단계에서 시스템을 유지/관리하는 방법을 사용한다. AIM 방법론은 규모가 큰 Project를 대상으로 개발된 것으로 기술적인 부분을 보완하면 ERP 외에 다른 시스템 구축 적용 시 활용할 수 있다.

AIM 방법론은 ① Business Requirements Definition, ② Business Requirements Mapping,

③ Application and Technical Architecture, ④ Module Design and Build, ⑤ Data Conversion, ⑥ Documentation, ⑦ Business System Testing, ⑧ Performance Testing, ⑨ Training, ⑩ Product Migration 등 10개의 Process로 나뉘어져 있고 Phases는 Initiation, Definition, Operation Analysis, Solution Design, Build, Transition, Production 등 7개로 구성되어 있으며 각 Process와 Phases의 상관 관계는 <그림 5>와 같다. 시스템 성능 검증과 연관된 Process는 Application and Technical Architecture와 Business System Testing 부분으로 전자에서는 Architecture 미래 환경 즉 Workload 증가에 따른 System Capacity의 확보가 고려되어야 하며, 후자는 Business 조건과 Test 조건을 연결하여 단위(Unit) 혹은 통합(Integrated) 성능 테스트를 수행하고 성능 요건에 만족하지 못할 경우 System Tuning 혹은 Architecture의 변화에 대한 제안이 필요하다.

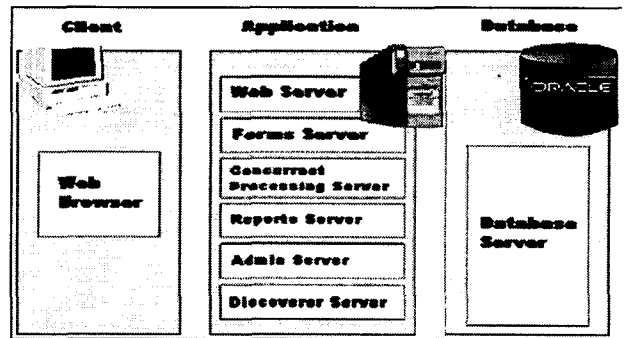


<그림 5> Oracle ERP 개발 방법론 AIM의 Framework

3.4 Oracle ERP Architecture 특성

Oracle ERP Package 역시 3 Tiers Architecture 구조를 기반으로 구성되어 있다. Application Layer 중 Web Server는 Apache, Forms Server는 CORBA Java 환경, Database Layer는 Oracle DB를 사용하고 있다. 통합 개발 환경인 Developer를 사용하며, Forms Server에 위치한 Application Program을 통해 Database의 Data를 처리할 수 있도록 구성되어 있다. JInitiator가 설치된 Client Layer에는 Application Layer와 연결 시 자주 사용되는 Java Class

File(*.JAR)을 Web Server로부터 Download해서 사용하며 사용 빈도가 적은 *.JAR File들은 필요할 때 Download해서 사용하는데 일단 Download한 *.JAR File들은 Network Traffic을 줄이기 위해 Client Layer의 Disk Cache에 남게 된다. Client 사용자는 Forms Server에서 제공하는 Interface를 통해 작업을 수행하고, 이때 특별한 원칙에 의해 주기적으로 실행되는 Reporting Program과 Data를 주기적으로 갱신하는 Program은 Background로 수행되는데, 이런 Batch 작업은 Concurrent Processing Server가 처리를 담당하게 된다.



<그림 6> Oracle ERP Layer 계층 구조

4. 성능 테스트 방법

4.1 성능 테스트 Process

Stage	Step
착수(B)	타당성 검토 후 문서 작성 → 성능 테스트 수행 전 준비 항목 검토 → WBS 및 진행 계획서 개발 → 착수단계 Output 이송
계획(P)	성능 테스트 계획서 작성 → 성능 테스트 Kick Off Meeting
분석(A)	업무 분석 사전 준비 → 업무 분석 → 성능 요구 사항 분석 → Architecture 분석 → 성능 측정 대상 업무 선정 → 검증용 테스트 환경 구축 → 툴 적용 검증 → Application 기능 사전 검증 → Workload Model 작성(초안) → Risk 파악
설계(D)	성능테스트 Workload 모델 확정 → Workload 모델 작성 → 실행용 테스트 환경 구축 및 Script 작성 → 모의테스트 수행 → Workload 모델 공유 회의 수행
실행(F)	준비 → 진행 → 측정 → 기록 → 정지 → 반복 → 종료 및 정리(단위/통합 성능 테스트)
종료(C)	테스트 결과 정리 → 시정 계획 및 내역 요청
평가(E)	성능 테스트 최종 평가서 작성 → 성능 테스트 최종 평가서 공유 → 종료 보고

<표 1> 성능 테스트 Process 단계

성능 테스트를 수행하기 위해서 ① 착수(B : Begin), ② 계획(P : Plan), ③ 분석(A : Analysis), ④ 설계(D : Design), ⑤ 실행(F : Fulfillment), ⑥ 종료(C : Close), ⑦ 평가(E : Evaluation) Process에 의해 업무를 진행하게 된다. 각 Stage에는 Step이 있는데 착수 단계의 타당성 검토 Step은 적용 Tool, 테스트 환경을 고려하여 성능 테스트가 가능한지 확인하는 작업이며, 분석 단계에서는 고객의 성능 요구 사항, 구축 System 환경, 전체 업무 중 성능 테스트 대상으로 선정할 업무, 생성할 부하량을 확인하는 작업이다.

업무 분석 시 <표 2>와 같은 기준을 고려하여 우선 순위를 파악하고, 고객 혹은 수행 팀의 성능 요구 사항/Architecture 구성/시간당 처리량을 분석한 자료를 사용해서 Workload Model을 산정한다. 일반적으로 실행 단계에서 기준 부하량의 80%, 100%, 120%를 생성하는 것을 원칙으로 하나, Peak Time의 부하량의 예측이 정확히 불가능한 시스템, 동시 사용자가 단위 시간 동안 처리하는 Transaction 수의 표준편차가 큰 시스템, Test 대상 시스템이 버틸 수 있는 한계 상황을 측정하고자 할 경우 Peak Time 시의 100%, 200%, 300%에 해당하는 부하를 생성하는 Stress Test를 수행한다.

번호	설명
1	발생 빈도가 높은 Application은 무엇인가?
2	데이터 처리량이 많은 Application은 무엇인가?
3	복잡한 Process를 가진 Application은 무엇인가?
4	공통으로 사용하는 Application은 무엇인가?
5	하나의 Entity를 여러 개의 Application이 처리하는 경우, 이 Entity에 관련된 Application은 무엇인가?
6	마감 작업 등, 특정 시점에 집중적으로 사용되는 Application은 무엇인가?
7	On-Line 시간대에 운영되는 Batch Application은 무엇인가?

<표 2> 업무 우선 순위 선정 양식

4.2 성능 지표 및 지표 해석 Metrics

성능 테스트는 크게 ① 설계한 부하량을 단위 시간 동안 Server에 생성하는 작업, ② 부하를

생성하면서 동시에 Server의 상태를 파악하는 시스템 모니터링 작업으로 구분할 수 있다.

Server의 상태를 파악하기 위해서는 기본적으로 ① System Resource, ② Response Time, ③ Throughput 관점에서 측정 방법 및 측정 지표를 가지고 있어야 하며, 부가적으로 ④ Database, ⑤ Application 관점에서 Server의 상태를 파악하여 성능 시험 결과를 통해 System Resource의 적정 사용 현황, 부분별 목표 응답 시간 준수 여부, 단위 시간 동안 목표 처리량 처리 여부 및 Database Server 및 Application Server의 처리 상태 및 정상 동작 여부를 확인할 수 있어야 하며 이를 파악할 수 있는 지표와 지표 해석 Metrics 기준이 필요하다. 일반적으로 System Resource의 Utilization을 측정하기 위해서는 크게 4가지 개체 즉 CPU(Processor), Memory, HDD, NIC 자원이 올바르게 사용되고 있는지 사용 현황을 조사해야 한다. UNIX 계열과 Windows 계열은 시스템의 차이점으로 인해 성능 측정 지표와 지표 측정 및 해석 방법에 차이가 있다. 지표는 절대적인 수치는 아니며, System Spec이 높거나 부하 분산(Load Balancing) 계획이 고려된 Hardware 환경일 경우 일반적인 성능 지표보다 높은 Utilization 상태에서도 안정적으로 동작하는 경우도 있다.

진단 항목	진단 내용	비고
CPU	가용 CPU 자원 확인, 실행열의 Run Queue 적정 확인, User, System, I/O Wait, Idle Mode에서 적정 사용 확인.	Windows, Unix, Linux 등 System O/S 종류에 따라
Memory	Available Memory Space 확인, Memory Page Scan Rate 확인, Memory Page Fault 확인.	System Component 별 Issue 분석 방법에 차이가 있음.
HDD	Disk Busy Rate, Disk Time 확인.	
Network Interface Card	Total Throughput 확인, Income Packet/Sec, Outgoing/Sec, Packet Error/Collision 확인.	
System Calls Interrupts Device Queues	적정 시스템 상태 파악을 위한 부가 지표 확인.	

<표 3> System Resource 사용 진단 항목 Sample

Response Time은 Max, Avg, Min, 90% Transaction 등이 있는데 응답 시간의 가장 정확한 지표가 될 수 있는 것은 90% Transaction 구간의 응답 시간으로 이는 최소에서 최대까지 응답 시간을 순서대로 나열했을 때 90%의 구간에서의 응답 시간을 나타낸 것이다. 90%까지는 해당 응답시간 이하의 결과를 얻고 10%는 해당 응답시간을 초과한 결과를 얻는다는 것을 의미한다. 90% Transaction 응답시간과 더불어 중요한 지표는 부하 생성 시 Ramp Up 기능을 사용해서 동시 사용자 수를 점진적으로 늘렸을 때 Transaction Average Response Time이 얼마나 증가하는지 나타내는 수치이다. 이 수치가 동시 사용자 수에 증가에 비례하여 급격하게 증가한다면 System Resource, Database Monitoring을 통해서 급격한 응답 시간의 증가 원인을 찾아 System, Database, Application 등을 Optimizing 하거나 Tuning할 수 있어야 한다.

Throughput은 Peak Time 시 예상했던 Transaction을 원활하게 처리하는지 확인할 수 있는 지표로 TPS(Transactions Per Second) 혹은 단위 시간동안의 TPH(Transactions Per Hour)로 나타낸 값을 통해 알 수 있다. 일반적으로 시간 단위로 TPS를 살펴보면 동시 사용자 수의 증가에 비례한 것으로 나타났으며, 이런 Pattern이 아닐 경우 예를 들어 Ram Up 기능 사용 시 사용자 수의 증가에 무관하게 TPS가 일정 수치에 묶여 있을 경우 이상 설정 여부를 점검해야 한다.

Database에 대한 Monitoring을 실시할 경우 DB의 종류 및 측정 방법에 따라 Biz-Max, Toad, Precise SQL 등의 다양한 Database Monitoring Tool이 사용된다. Tool에서 지원하는 Monitoring 항목에 따라서 SGA Library/Dictionary/Data Cache Hit Ratio, Logical/Physical Read, Direct I/O, Wait, Redo Entry 등 각기 조금씩 다른 성능

지표에 대해 Monitoring할 있다.

구분	진단 항목	진단 내용 및 결과 요약
데이터베이스 구성	Parameter 구성	SGA 영역의 설정 적합성 검증, I/O 관련하여 일부 Parameter 수정. On Line Redo Log File, Rollback Segment, Extent 검증 및 수정.
데이터베이스 자원	Memory	Buffer Cache hit율이 저조하며 Setting 수정 및 검증.
	Contention	Library Cache Lock이 발생하면서 Error 발생 DBMS 검증. 특정 Object들에 대한 Single/Multi Block Read Wait가 과다 발생 원인 분석 및 현상 제거.
	Disk I/O	특정 Data에 대한 I/O 집중으로 인한 병목 발생. I/O 분산 대책 필요.
데이터베이스 성능	Index 구성	Index Scan 효율성이 좋지 못한 인덱스들이 일부 있으며 또한 어플리케이션 튜닝을 통한 인덱스 추가 생성 및 조정 작업 필요
	성능	Response Time내 Wait Time의 비율이 높은 원인 분석 및 현상 제거.

<표 4> Database 사용 현황 진단 항목 Sample

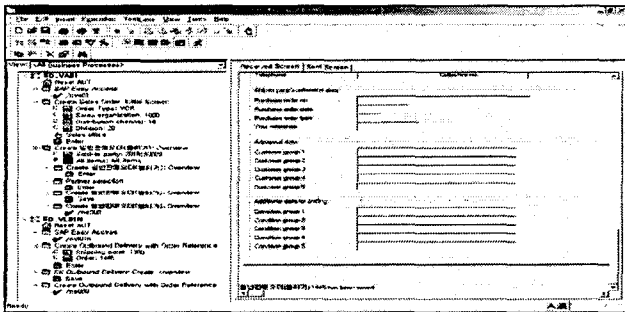
4.3 SAP 및 Oracle ERP 성능 테스트 방법

SAP R/3, mySAP.com 및 Oracle ERP Package로 구축한 시스템에 대해서 적용 가능한 Tool을 조사해 본 결과 Mercury Interactive사의 Load Runner 7.x Version과 IBM(Rational)의 Performance Studio(Robot)을 사용하여 성능 테스트가 가능했다. 해당 성능 테스트 Tool을 활용하여 부하 생성 및 시스템 모니터링 작업을 수행 시 중요한 개념은 아래와 같은 3가지이다.

- User Pattern과 Peak Time 집중도 파악 : Performance Testing Tool을 사용하여 User의 Action을 재현할 때 사용자 Pattern을 파악하여 실제 최대한 유사한 부하를 생성하는 것이 필요하다. 일반적으로 User Action은 ① 접속 또는 인증 획득, ② 메뉴 선택 또는 이동, ③ 작업 수행, ④ 종료 인증 해제 등으로 나눌 수 있는데 Testing Tool을 적용하여 작업을 반복 수행할 경

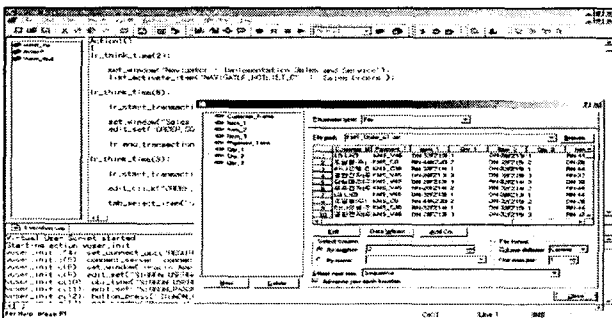
우 1 User의 전체 단위 Action을 모두 똑같이 반복할 지 혹은 접속과 종료는 각 1번을 수행하고 ② 메뉴 선택 또는 이동, ③ 작업 수행만을 반복할지 결정해야 한다. 더불어 Ramp Up 기능을 사용하여 동시 사용자를 점차적으로 시간 간격을 두고 증가시킬 지 아니면 일시에 증가시킬 지를 결정하는 것이 필요하다.

● Correlation 처리 : Client와 Server 사이에서 Correlation이 일어날 경우 상호 정확한 Data를 주고 받게 하기 위해서 Client와 Server 사이에서 일어나는 Communication에 대해서 Correlation을 처리해야 한다. 일반적으로 접속한 사용자, 처리 작업 순서에 따라 Server가 생성한 Unique한 값을 서버로부터 받아다 Client가 특정 작업을 수행할 때 그 Unique한 값을 입력해야 할 경우 사용하게 된다.



<그림 7> Tool에서의 Correlation 처리 예제

● Data Pool 처리 : Data Pool이란 성능 테스트 시 Tool 안에 미리 지정한 Data의 집합을 의미하며 동시 사용자가 각각의 데이터를 Unique, Sequential, Random, User N/O 등 Data를 사용하는 방법을 지정한 대로 가져와서 사용할 수 있도록 조정하는 것을 의미한다.

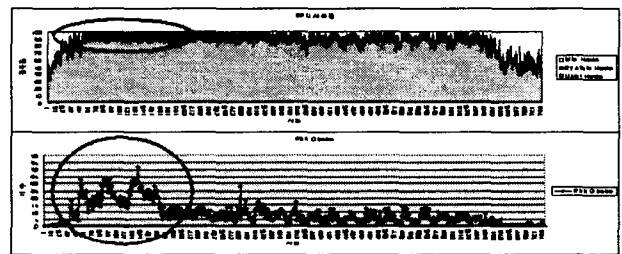


<그림 8> Tool에서의 Data Pool 처리 예제

5. System 성능 Issue 및 사례 연구

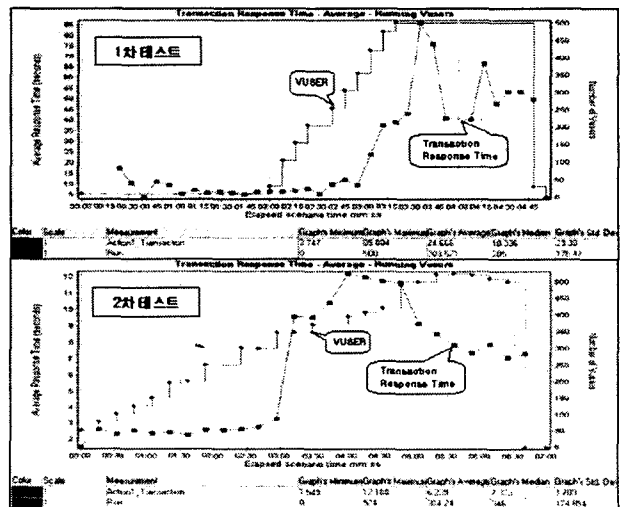
SAP R/3, mySAP.com 및 Oracle ERP Package로 구축한 시스템에 대해서 성능 테스트를 실시한 결과 발견된 Issue는 ① Hardware 관련 Issue, ② O/S 관련 Issue, ③ Database 관련 Issue, ④Application 관련 Issue로 구분할 수 있다. 해당 Issue들은 완전히 독립된 Case가 아니라 상호 연관성을 가지고 있기 때문에 Issue 발생 시 함께 검토할 필요성이 있다.

Hardware 관련 Issue는 CPU, Memory Resource의 부족 혹은 Spec이 낮은 경우, L4나 RAC, HA 등의 환경에서 Setting이 올바르게 설정되지 않아 부하 분산이 이루어지지 않는 경우였다.



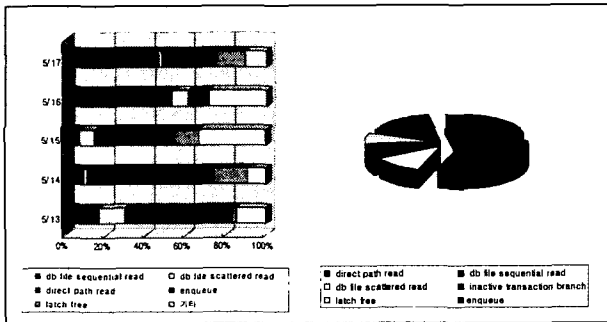
<그림 9> CPU Resource 부족/Run Queue 과다 사례

O/S 관련 Issue는 주로 O/S Parameter 설정이 System Spec이나 Resource Size에 부적합하게 설정된 경우로 TCP Parameter 설정 변경 등을 통해 시스템 사용 현황, 전송량 등이 개선되었다.



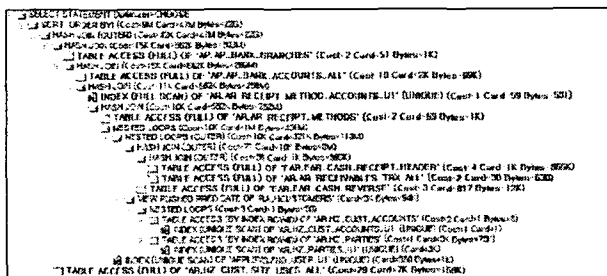
<그림 10> O/S Setting 변경 전/후 비교 사례

Database 관련 Issue는 DBMS(Database Management System)의 Parameter Setting이 올바르게 설정되지 않은 경우, Library Cache Lock 이 발생한 경우, SGA 영역의 Cache Hit율이 기준치보다 낮은 경우, Lock Wait Time이 매우 긴 경우, Dead Lock이 발생하여 특정 Module의 Transaction이 Hold 상태에 빠지는 경우였다.



<그림 11> Database 상의 Event Wait 분석 사례

Application 관련 Issue는 Web Server, WAS Parameter 설정이 적절하지 않은 경우, Application Program Source의 구문 Coding이 부적절한 경우였다.



<그림 12> Source Code 구문별 적합성 검증 사례

6. 결론

ERP System에 대한 성능 테스트를 수행하는데 필요한 선행 지식과 수행 방법, 사례 고찰을 통해서 ERP System 구축 시 발생 가능한 성능 Issue를 최소화할 수 있는 방안을 연구해 보았다.

ERP System 구축 Project의 경우, 착수 단계에서 종료 단계 이후 운영 단계까지 전체 Cycle에서 관리해야 할 많은 요소가 있는데, 각기 ERP 구축 방법론에서도 Project 수행 Process의 일부로써 성능 테스트를 수행하도록 명시하여

System의 성능 점검을 필수적으로 수행하도록 권고하고 있다.

그런데 성능 테스트를 포함한 품질 점검은 System이 거의 완성되는 Project 단계 말 뿐만 아니라 필요한 Process 단계에서 Check가 이루어지는 것이 바람직하다. 즉 개발이 완료되기 이전에 Issue 발견되면 해결을 위한 작업을 간소하게 처리할 수 있기 때문에, 재작업에 대한 기간이나 비용에 단축에 기여할 수 있게 된다.

특히 연구한 ERP Package로 구축한 System은, Package에서 제공한 Module을 사용해서 각 업무별 Application을 제작하므로, Big Bang 방식으로 한꺼번에 개발이 이루어진다고 해도 Module별 차등적으로 나누어 단위 성능 테스트를 실시한 후 전체 관점에서 통합 성능 테스트를 실시하면 문제 파악에 더욱 쉽게 접근할 수 있을 것으로 판단된다.

더불어 구축하려는 ERP System의 규모가 크고, Hardware Spec이나 Resource가 다른 복수의 System에 각 ERP Module을 분산시켜 System을 구축할 경우, 어떤 Module을 같은 H/W 환경에 위치시키느냐에 따라서 부하 집중도가 크게 달라질 수 있다. 때문에 Module별로 분산 배치할 수 있는 Case를 조사하고 Case 별로 BMT(Benchmark Test)과 유사한 성능 테스트를 실시하여, 최적의 성능을 유지할 수 있도록 Module을 배치한다면 전체 시스템 관점의 부하 분산(Load Balancing)에 큰 효과를 볼 수 있을 것이다.

구축 중 혹은 구축 완료된 System에 대해 단위 시험(Unit Test), 연계 시험, 설치 시험, 통합 시험(Integrated Test), 시스템 시험(System Test)을 시행하게 되는데 단위 시험을 진행하면서 Customized 및 Bolt On으로 기능 변경이 가장 많은 부분에 대해서 중점적으로 Module 성능 테스트를 진행할 필요성이 있다.

본 연구 산출물을 활용한다면 SAP R/3나 Oracle ERP Package를 사용해서 ERP System을

구축하는 Project 현장에서 System Performance 관련 품질 Issue를 최소화하는 Guideline으로 활용할 수 있을 것으로 판단된다.

연구 과제를 진행함에 따라 넓은 연구 범위, 부족한 선행 연구 사례, 설명이 깊어지고 길어지는 특정 부분, 특정 Project에 한정된 사례 등의 난점이 발견되기도 했다. 특히 원래 계획은 자체 개발 혹은 다양한 ERP Package로 구축한 전체 ERP System에 대해서 성능 테스트를 수행하는 방법과 발견된 Issue에 대해서 사례 연구를 시도하려고 했으나 실제 성능 테스트를 지원한 ERP System 구축 Project가 특정 ERP Package로 한정되어 있었기 때문에 일부 Chapter에서는 연구 Focus가 좁아지기도 했으나 최대한 일반적인 ERP System 구축 Project에 Case를 확대할 수 있도록 연구를 이끌어 보았다.

차후 보다 다양한 ERP Package를 사용하여 System을 구축할 경우, 해당 사례를 Case로 추가하여 내용을 보강하여 보다 범용적인 System에 적용할 수 있도록 보완했으면 한다.

[참고문헌]

- [1] ERP 감리지침 연구, 한국전산원, 구자환, 2001
- [2] ERP기반의 CTO시스템 구현에 관한 연구, LG CNS 기술대학원, 임기철, 2003
- [3] 경영혁신의 도구로서 ERP 시스템의 구축에 관한 연구, 한국정보시스템학회 97추계학술대회 논문집, 김영문, 1997
- [4] ERP 시스템 도입효과, 한국정보시스템학회, 97추계학술대회 논문집, 박영용, 1997
- [5] E-비즈니스를 위한 ERP Oracle Applications, 대청미디어, 대청미디어 편집부, 2000
- [6] ERP 전략과 실천, 대청미디어, 이동길, 1999
- [7] 국내 전사적 자원 관리(ERP) 시스템 도입에 관한 연구, 정보통신학술 연구, 이선로, 1999

- [8] ERP의 성공적인 도입전략, 한국경영정보학회 논문집, 오재인/이석주, 1998
- [9] SAP R/3 구현의 핵심 성공요인과 성과분석, 한국경영정보 학회 논문집, 황화정 외 2인, 1999
- [10] 국내 ERP연구에 대한 고찰과 과제, 한국경영정보 학회 춘계학술대회, 최무진, 1999
- [11] 소프트웨어 프로젝트 관리론, 흥릉과학출판사, 류성열/여호영, 1995
- [12] Testing Applications on the Web - Test Planning for Internet-Based systems, Hung Q. Nguyen, John Wiley & Sons, Inc, 2001
- [13] Software Testing Automation - Effective use of test execution tools, Mark Fewster & Dorothy Graham, ADDISON-WESLEY, 1999
- [14] Automated Software Testing - Introduction, Management and Performance, Elfriede Dustin & Jeff Rashka & John Paul, ADDISON-WESLEY, 1999