JENNIFER
JENNIFERSOFT, Inc.

Oct. 2011

성능분석모델

JENNIFERSOFT, Inc.

Table of Contents

Executive Summary

Executive Summary

Challenge

치열한 비즈니스 경쟁 속에서 살아남기 위한 일환으로 고품질 정보 시스템으로의 전환은 필수적이다. 하지만 이러한 비즈니스 요건에 부합하기 위한 막대한 노력에도 불구하고, 기업들은 비즈니스에 치명적인 성능 장애에 빈번히 노출되고 있다.

Opportunity

정보 시스템 성능 분석에 필수적인 성능 이론/모델에 대한 이해를 통하여 정보 시스템 기변에 깔려 있는 성능에 대한 근본 원리를 이해할 수 있고 벤치마크,성능 테스트, 용량 계획 등의 실무적인 성능 분석 활동들에 적용할 수 있는 적용방안을 마련할 수 있다.

Benefits

본 백서는 웹 기반 시스템 하에서의 성능 이론/모델 소개와 더불어 실 세계에서의 성능 분석 방안을 함께 소개함으로써 정보 시스템 성능 분석에 지침서로 활용할 수 있다.

오늘날 많은 기업들은 치열한 비즈니스 경쟁 속에서 살아남기 위해서 고품질 정보 시스템으로의 전환을 추진하고 있다. 글로벌 시장의 무한 경쟁 시대에 기업 경쟁력 제고를 위하여 정보 시스템은 대용량 데이터를 원하는 시간 내에 처리할 수 있어야 하고, 소비자들이 필요로 하는 정보를 원하는 시간 내에 제공할 수 있어야 한다.

"저희 회사 마케팅 시스템은 캠페인 기간 동안 시간당 1억 건 이상의 단문메시지를 고객에게 전송 할 수 있어야 합니다."

"저희 은행 계정계 시스템은 월 마감 시에 초당 7천 건에 달하는 업무 요청들을 3초안에 처리할 수 있어야 합니다."

기업들은 이와 같은 비즈니스 요건에 부합하기 위해서 정보 시스템 구축 시에 수많은 활동들을 수행하게 된다. 예컨대 프로젝트 착수 단계에서는 벤치마크(Benchmark)를 통하여 비즈니스 요건에 최적화된 하드웨어와 소프트웨어를 도입한다. 그 후 프로젝트 수행 단계에서는 각 개발 단계 별로 사전 계획된 성능 테스트 및 성능 튜닝 등의 다양한 품질 관리 활동을 통해 비즈니스 요건에 적합한 시스템을 구축하게 된다.

하지만 이와 같은 막대한 투자 노력에도 불구하고, 시스템 오픈 첫날 여지없이 성능 장애로 이어지거나, 오픈 이후 얼마 지나지 않아 성능이 점차적으로 느려지면서 성능 장애로 치닫게 된다.

"과연 무엇이 이러한 현상을 일으키는 것일까?" 박복한 프로젝트 예산, 단축된 프로젝트 기간으로 인한 테스트 부족, 투입된 프로젝트 인력들의 전문성 결여 등 여러 가지 이유가 있겠지만, 그 중 가장 큰 이유 가운데 하나는 성능 분석에 참여하는 인력들의 시스템 성능에 대한 이해 부족과 잘못된 논리 전개에 기인한다고 봐야 한다.

본 기고에서는 이러한 악순환을 단절하고 궁극적으로 정보 시스템 성능 분석 모델에 지침서로 활용할 수 있게끔 성능 이론/모델 정립과 더불어 실 세계에서의 성능 분석 방안에 대해서 소개하고자 한다.

Section 1

성능 모델(Performance Model)

현실세계에서 일어나는 현상에 대해서 근본원리를 이해하고자 할 때 수학적 모델을 정립하여 활용하곤 한다. 즉 어떤 현상의 특성 및 구조를 가장 잘 대표하는 변수들을 찾아내고, 그 변수들의 상관관계를 수식적으로 재 구성함으로써 수학적 가상공간을 설정하고, 수식과 논리의 해법을 통해 의미 있는 결론 혹은 현상을 이해하게 된다. 그런 다음 현실세계의 현상에 다시 적용함으로써 복잡하게만 보이는 현상을 이해할 수 있고, 하나의 변수가 변화할 경우 어떻게 현상이 진행될 것인가를 예측할 수 있게 된다.

정보 기술 분야도 마찬가지다. 정보 시스템에 대한 성능 분석 활동 - 벤치마크, 성능 테스트, 용량 계획- 의 근본원리를 이해하기 위해서는 수학적 모델에 기반한 성능 모델 정립은 반드시 필요하다. 지난 80년대를 풍미했던 IBM 메인프레임부터 90년대 CS시스템/TP-MONITOR에 이르기까지 여러 시대에 걸쳐 다양한 분야에 활용되고 있는 수학적 모델은 큐잉 이론이다. 이 이론은 복잡한 정보 시스템을 단일 큐 또는 단일 큐들의 조합으로 구성된 큐잉 네트워크로 표현하고, 성능 모델 변수들간의 상관관계를 수식으로 계산하여 시스템 성능을 예측하는데 활용되고 있다.

하지만 큐잉 이론은 인터넷의 폭발적인 성장과 더불어 시작된 웹 기반 시스템에 그대로 적용하기에는 다음과 같은 제약사항이 따른다.

첫째, 최근의 웹 기반 시스템은 웹 어플리케이션 서버를 근간으로 상당히 복잡한 아키텍처로 구성되어 있다. 과거 메인프레임이나 TP-MONITOR의 경우 미들웨어와 DBMS로 구성된 비교적 단순한 형태의 아키텍처로 구성되었지만, 최근의 웹 기반 시스템은 분산 트랜잭션 처리, 서비스 지향 아키텍처와 같은 최신 트랜드에 발맞추어 복잡한 아키텍처로 구성되기 때문에, 성능 모델 수립에 많은 어려움이 따른다.

둘째, 웹 기반 시스템이 사용하는 HTTP 프로토콜의 특성으로 인해 성능 모델 변수 특히 동시 사용자 수를 측정하기가 매우 어렵다. 과거 메인프레임이나 TP-MONITOR의 경우 동시 사용자는 클라이언트와 서버가 맺고 있는 TCP 연결 개수를 계산하여 측정할 수 있지만, 웹 기반 시스템은 업무 요청 처리가 끝난 이후 기존 TCP 연결을 끊어버리기 때문에 이 방법으로는 측정이 불가능하다.

셋째, 웹 기반 시스템은 일반적으로 불 특정 다수를 대상으로 서비스를 제공하기 때문에, 성능 모델 변수에 대한 입력 값을 예측하기도 매우 어렵다. 장기간에 걸친 시스템 모니터링을 통해 해당 변수에 대한 값을 예측하더라도, 마케팅 활동과 같은 외부 요인에 크게 영향을 받기 때문에 데이터 신뢰도는 높을 수가 없다.

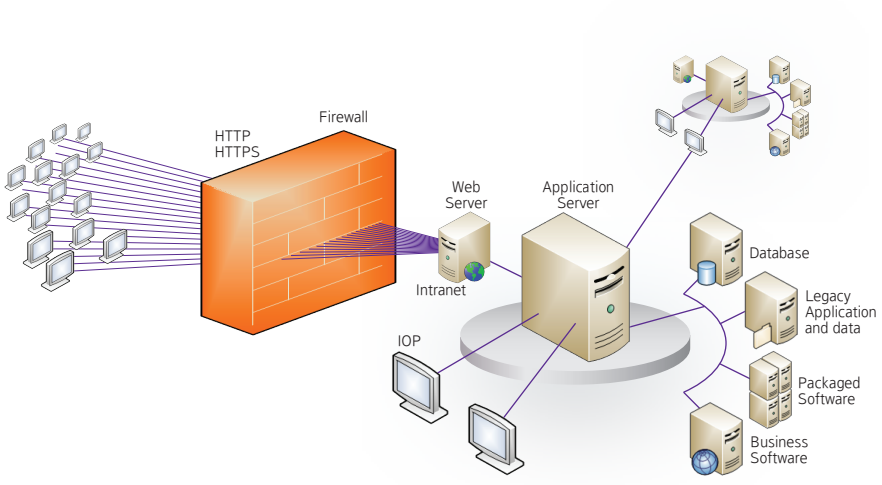
따라서 웹 기반 시스템하에서 일어나는 성능 현상을 올바르게 이해하기 위해서는, 앞으로 소개할 웹 어플리케이션 서버 환경에 맞게끔 재 조명된 성능 이론/모델링에 대한 이해가 필수적이다.

1.1 성능(Performance)이란?

정보 시스템 성능은 정보 기술 업계에 종사하는 사람들에게는 매우 익숙한 용어이다. 하지만 성능의 의미에 대해서 정확하게 알고 있는 사람은 많지 않다. 성능은 자주 쓰이는 용어이긴 하지만, 올바른 과학적인 정의 없이 통용되어 혼란을 주고 있는 대표적인 용어이기도 하다.

성능을 설명하기 위해서 우리 주변에서 흔히 볼 수 있는 상황을 예로 들어보자. “주말에 대형 마트에 장을 보러 가면 계산대 앞에서 한없이 기다려 본 적이 많을 것이다. 주중에 비해서 대형 마트를 찾는 고객이 많아서 계산대 직원 수가 상대적으로 부족할 뿐만 아니라, 고객별 구매 물품 수도 많아서 계산 시간도 오래 걸리기 때문이다.”

정보 시스템의 성능을 측정 할 때도 앞서 소개한 대형 마트 예와 매우 유사한 특성을 갖는다. 계산대의 성능이 고객들에게 원활한 서비스를 제공할 수 있느냐가 관심이듯이 정보 시스템의 성능은 얼마나 많은 사용자에게 안정되고 원활한 서비스를 제공할 수 있느냐가 관건이다.

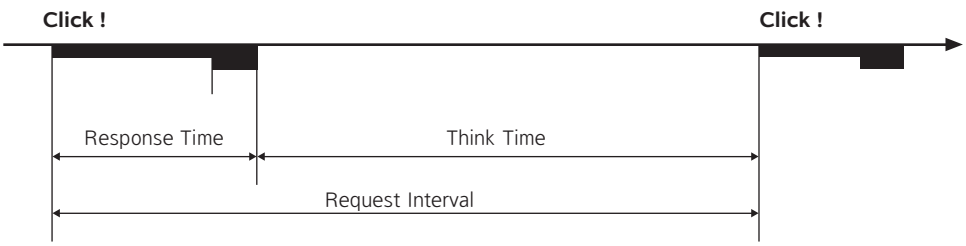


따라서 정보 시스템 성능은 다음과 같은 세가지 관점으로 집약 될 수 있다.

- How many clients?: 얼마나 많은 사용자(clients)들에게 서비스를 할 수 있는가?
- Resonable response time: 적정한 응답시간 이내의 안정된 서비스를 제공하는가?
- Cost-effective: 가능하면 비용 대비 효과(cost-effective)가 높아야 한다.

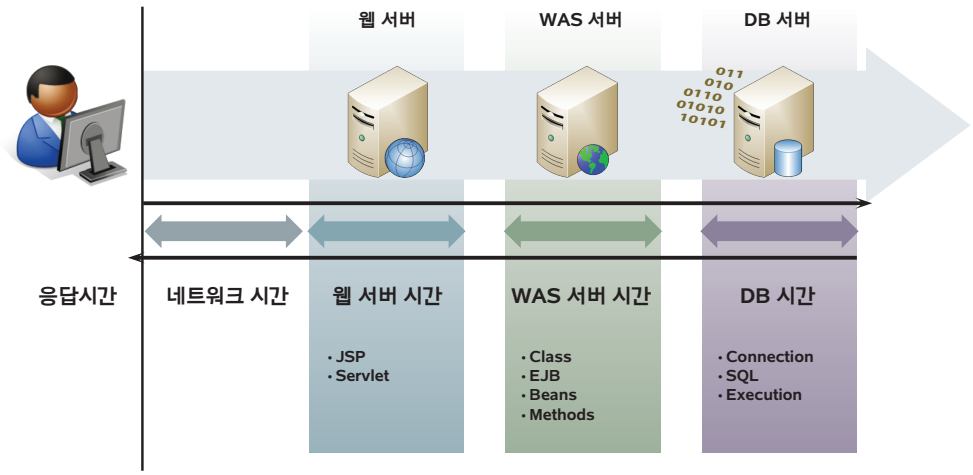
1.2 성능 관련 용어

본격적인 성능 이론/모델링 소개에 앞서 성능 분석 시에 주로 사용되는 각종 용어들에 대해서 알아보자



1.2.1 응답 시간(Response Time)

응답 시간은 앞서 정의한 정보 시스템 성능에 대한 세가지 관점 중 “Reasonable response time”에 해당되는 성능 평가 항목으로써, 사용자의 업무 요청에 대한 응답 결과를 받을 때까지 걸리는 총 소요시간으로 정의된다. 웹 기반 시스템의 경우 아래 그림과 같이 클라이언트 시간, 네트워크 시간, 웹 서버 시간, WAS시간, DB시간 등으로 세분화 되는데, 이들 응답시간 상세 분석 결과는 성능 저하를 야기하는 병목 원인(bottleneck)을 찾고자 할 때 매우 유용하게 사용된다.



1.2.2 대기 시간(Think Time)

대기 시간은 사용자가 다음 업무 요청을 보내기 위해서 준비하는 시간으로 정의되며, 직전 요청에 대한 응답 결과를 보고 있거나, 다음 요청을 보내기 위해서 PC단말 화면상에 값을 입력하는 시간 등으로 구성된다. 여기서 대기 시간은 사용자마다 고유한 업무 사용 패턴,업무 이해도, 업무 능숙도 등에 따라 달라 질 수 있기 때문에, 각 시스템 별로 해당 수치를 측정하는 방안이 필요하다. 만약 실제 운영 환경에서 측정이 불가능할 경우에는, 아래 표에 제시된 비즈니스 도메인 별 프로젝트 경험치를 사용해도 무방하다.

아울러 SAP, Oracle과 같은 솔루션 벤더들의 경우, 각 사에서 제공하는 솔루션에 대해서 벤치마크와 성능 테스트에 적용 가능한 대기 시간 권고치를 별도로 제공하고 있다.

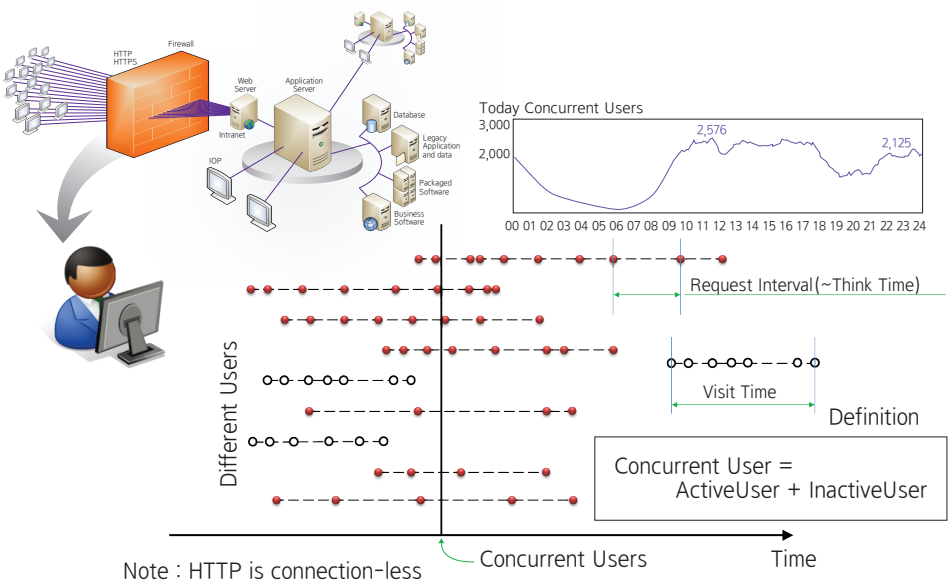
구분	Think 시간	비고
TM(Telemarketing) 시스템	10 ~ 15초	
MIS/인트라넷 시스템	15 ~ 20초	
인터넷뱅킹 시스템	30 ~ 35초	
온라인쇼핑몰 시스템	30 ~ 40초	
포털 시스템	40초 이상	

1.2.3 호출 간격(Request Interval)

호출 간격은 사용자가 업무 요청을 보낼 때 소요되는 총 시간 간격으로 정의되며, 응답시간과 대기시간의 합으로 구성된다. 시스템 부하량과 직결된 성능 모델 변수 중 하나로써, 성능 테스트 자동화 솔루션에서는 현실 상황과 유사하게 이를 재현 해 주는 기능들을 제공한다.

1.2.4 동시 사용자(Concurrent User)

특성 시점에 대상 시스템에 접속하여 현재 업무 요청을 보내고 있거나 다음 업무 요청을 보내기 위해서 준비하고 있는 사용자를 일컫는 것으로써, 시스템 부하량과 직결된 성능 모델 변수 중 하나이다. 앞서 소개하였듯이 웹 기반 시스템의 경우 HTTP 프로토콜 특성으로 인하여, 동시 사용자 측정이 용이하지 않기 때문에 아래와 같은 새로운 접근 방법을 모색해야 한다.



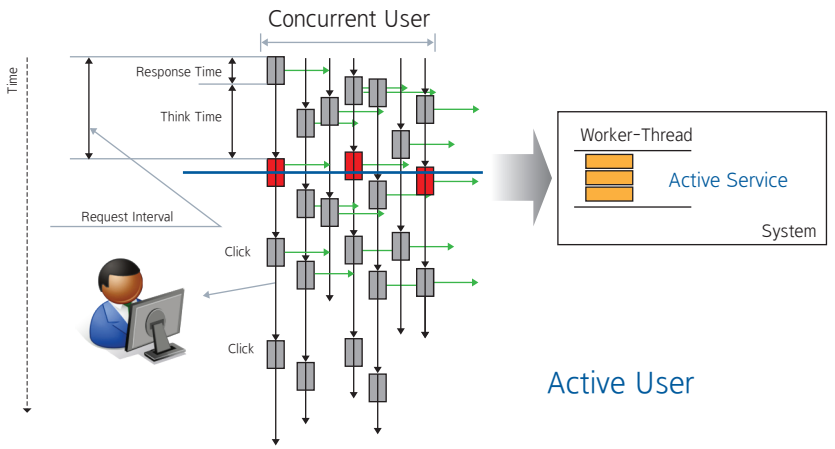
위 그림에서와 같이 각 시점 별 개별 사용자의 업무 요청을 그래프상의 하나의 점으로 표현해 보았을 때, 각 사용자는 나름의 호출 간격으로 업무 요청을 반복해 나간다. 이때 업무 요청은 클라이언트/서버간의 TCP 연결은 매번 끊어지지만, 지속적인 요청 형태를 띠고 있기 때문에, 시스템을 사용하고 있다고 봐도 무방하다.

아직까지 웹 기반 환경에서 동시 사용자에 대한 정의는 표준화되어 있지 못한 실정이다. 어떤 이는 “동시 사용자”를 “전체 사용자(Named User)”로, 어떤 이는 아래에서 설명할 “액티브(Active)사용자”로 재해석하여 혼란을 가중시키고 있다. 따라서 본 기고에서는 웹 기반 시스템하에서의 동시 사용자는 해당 시스템을 사용하기 위해, 단말 앞에 앉아 있는 서로 다른 사용자 수로 정의하기로 한다.

1.2.5 액티브 사용자(Active User)

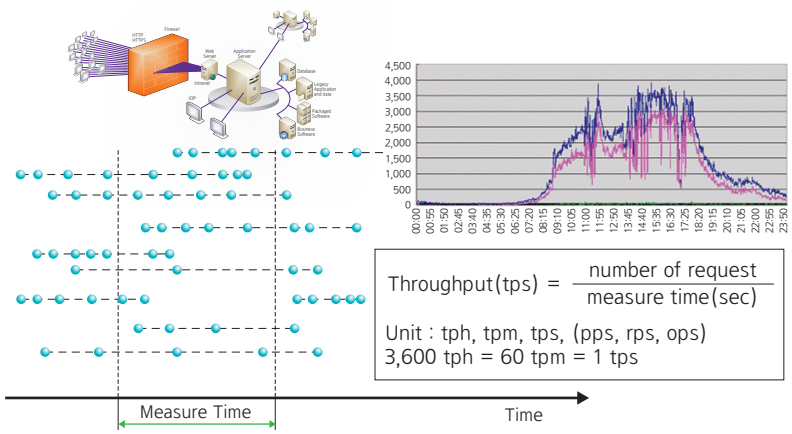
각 개별 동시 사용자는 각각의 응답시간과 대기시간을 가지며 요청을 반복하게 된다. 특정 시각에서 시간을 칼로 무 자르듯 잘라 보면, 해당 시각에 요청 이후 아직 응답을 받지 못한 사용자, 혹은 해당 요청이 현재 수행되고 있는 사용자들이 존재한다. 아래의 그림에서는 3명의 사용자가 해당 시점에 요청을 날린 후 응답을 기다리고 있는 사용자이다. 이와 같이, "요청을 날리고, 응답을 기다리고 있는 사용자"를 "액티브(Active)사용자"로 정의한다.

만약, 네트워크 병목이 없다면, 액티브(Active)사용자의 요청은 서버 측에서 해당 시점에 서비스가 수행된다. 여기서 서버 측에서 해당 시점에 서비스가 수행되고 있는 개수를 "액티브(Active)서비스"로 정의한다. 액티브사용자가 클라이언트 측에서의 관점이라면, 액티브서비스는 서버 측 관점에서 바라본 수치이다.



1.2.6 처리율(Throughput)

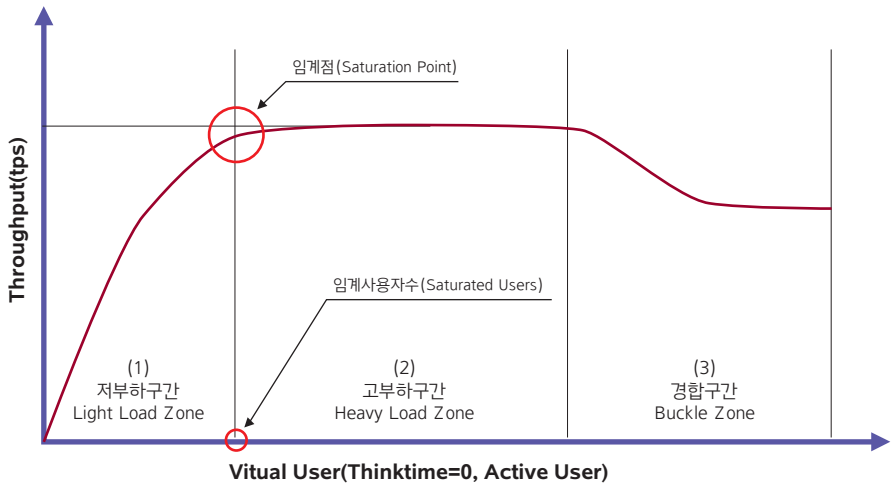
처리율은 앞서 정의한 정보 시스템 성능에 대한 세가지 관점 중 "How Many Client"에 해당하는 성능 평가 항목으로써, 단위 시간 동안에 처리된 사용자 요청 건수로 정의된다. 처리 대상 유형에 따라서 서로 다르게 표현되는데, 네트워크 리소스는 bits/sec 또는 bytes/sec로, CPU 리소스는 mips(millions of instructions per second), OLTP(On-Line Transaction Processing)은 trans/sec 또는 TPS로 각각 불려진다. 본 기고에서는 주로 웹 기반 시스템하에서 트랜잭션 처리 기준으로 성능을 평가하기 때문에 trans/sec 즉 TPS로 표현하기로 한다.



처리율은 임의의 측정 구간(Measure Time)동안 처리된 사용자 요청 건수를 측정 구간으로 나눠서 구하는데, 서비스 큐잉에 의한 처리 지연이 발생하지 않은 한 서비스 요청율(Requests rate)과 같아야 한다.

아울러 처리율은 사용자 요청 건수 즉 부하량 증가에 따른 변화 추이에 따라 아래 그림과 같이 세 구간으로 나누어진다.

첫 번째 저부하구간(Light Load Zone)에서 처리율은 동시 사용자 증가에 따라 선형적으로 증가한다. 이 구간에서는 병목 현상이 없기 때문에, 부하량 증가에 비례하여 처리율이 증가하기 때문이다. 두 번째 고부하구간(Heavy Load Zone)에서 최대 허용 가능 동시사용자 즉 임계사용자수까지 부하량이 증가하게 되면, 처리율은 병목 현상에 의해서 더 이상 증가하지 못하고 수평을 유지하거나 들쑥날쑥해진다. 세 번째 경합구간(Buckle Zone)에서 부하량이 더 증가하더라도, 처리율은 오히려 감소(Buckle Zone)하는 패턴을 보이는데, 1차 병목 지점 이외에 추가적인 병목 지점이 추가로 발생하여, 하나 이상의 병목이 중첩되어 오히려 처리량을 끌어내리는 효과 때문에 발생한다.

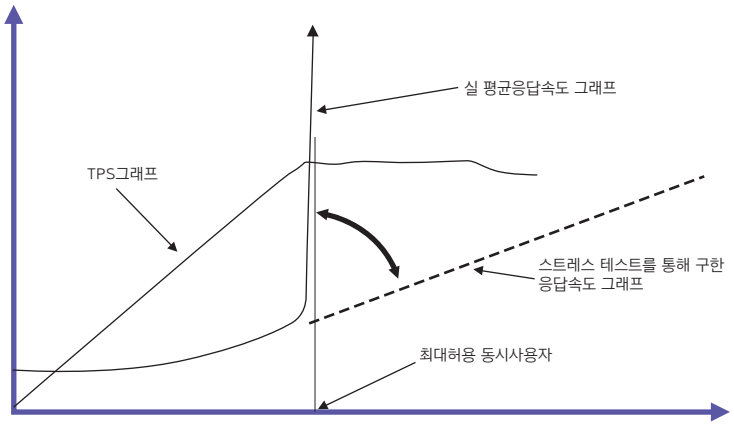


1.2.7 처리율과 응답시간 상관관계

정보 시스템 성능 평가 시에 반드시 포함되어야 할 중요한 관문은 다름아닌 처리율과 응답시간과의 상관관계 분석이다. 모든 벤치마크, 성능 테스트 결과에서 해당 분석 결과가 항상 거론되는 이유도 이러한 중요성에 기인한다고 볼 수 있다.

원론적으로 처리율과 응답시간 현황은 부하량 증가에 따라 임계점을 기준으로 증가 패턴이 서로 뒤바뀌는 양상을 띤다. 즉 처리율은 동시 사용자 증가에 따라 임계점 부근까지 선형적으로 증가하다가, 그 이후 구간부터는 병목 현상에 의해서 더 이상 증가하지 못하고 수평을 유지하거나 감소하게 된다. 하지만 응답시간은 동시 사용자가 증가하더라도 임계점 부근까지는 일정하게 유지되다가, 그 이후 구간부터 병목 현상에 의해 야기된 큐잉 시간 증가에 따라 가파르게 증가한다.

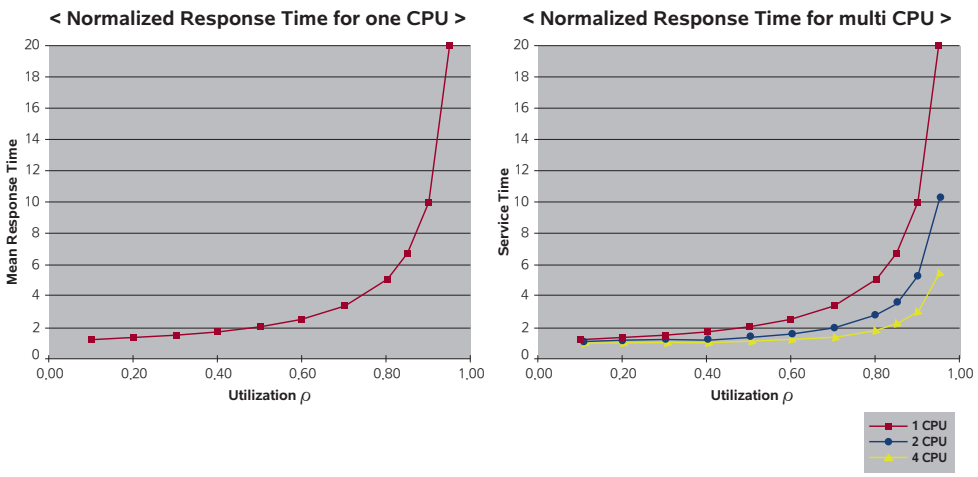
따라서 대상 시스템의 최대 성능을 올바르게 평가하기 위해서는, 처리율은 처리율 현황 그래프상의 임계점 이후 구간에 대한 최대 값을 기준으로 정의되고, 응답시간은 응답시간 현황 그래프상의 임계점 이전 구간에 대한 평균 또는 90 백분위수를 기준으로 정의되어야 한다.



동시 사용자(Concurrent 사용자)

1.2.8 CPU사용률과 응답시간 상관관계

단일/다중 CPU 환경에서 CPU사용률 증가에 따른 응답시간의 변화 추이를 비교해 보면 이들 간의 상관관계에 대해서 흥미로운 결론을 얻을 수 있다.



좌측 그래프(단일 CPU 환경)에서 CPU 사용률이 50% 이상 증가 시 큐 길이가 점차적으로 길어지면서, 응답시간이 크게 느려 지는 현상을 볼 수 있다. 즉 단일 CPU 환경에서는 CPU 사용률이 50% 이상 증가 시 서버의 큐 길이와 서비스 응답시간은 비례 관계 보다 훨씬 가파르게 증가한다.

반면 우측 그래프(다중 CPU 환경)에서는 CPU 개수가 많아짐에 따라 응답시간의 증가치가 완만하게 줄어드는 현상을 볼 수 있다. CPU 개수가 4개 이상일 경우는 CPU 사용량이 80%까지 일정하게 유지되다가 80% 이상부터 완만하게 증가한다.

따라서 정보 시스템 용량 계획 시 서버 CPU 리소스에 대한 임계 사용률을 70~80%로 권장하는 이유도 단지 경험적인 판단이 아니라 성능 모델을 통한 통계 데이터에 기인한다고 봐도 무방하다.

1.3 웹 기반 하에서의 성능 모델

웹 기반 하에서의 성능 모델을 이용한 성능 분석 절차는 다음과 같이 3단계를 거친다.

첫 번째 모델 수립(Model Construction) 단계에서는 성능 분석 활동의 목적을 정의하고 모델 유형을 결정한다. 성능 분석은 벤치마크, 성능 테스트, 용량 계획 등 여러 용도로 활용될 수 있기 때문에, 사전에 목적을 분명하게 해야 한다. 성능 모델 유형도 성능 테스트 자동화 솔루션을 기반으로 하는 실측을 통한 검증 방법과 해석 모델(Analytical Model)을 기반으로 하는 시뮬레이션 검증 방법 등 두 가지 유형이 있기 때문에, 사전에 어떤 방법을 적용할지 결정해야 한다. 일반적으로 벤치마크와 성능테스트는 전자인 실측을 통한 검증 방법을 이용하고, 용량 계획은 후자인 해석 모델 기반의 시뮬레이션 검증 방법을 이용한다.

두 번째 모델 변수화(Model Parameterization) 단계에서는 성능 모델을 결정짓는 모델 변수(Parameter)를 정의한다. 웹 기반 시스템을 위한 성능 모델에서는 두 가지 유형의 모델 변수, 즉 부하 강도(Workload Intensity)를 결정짓는 서비스 요청률, 동시 사용자와 서비스 요구량(Service Demand)을 결정짓는 응답 시간, 대기 시간을 사용한다. 이들 모델 변수에 대한 실제 값은 운영 상황에서 수작업 측정을 통해서 수집되거나, 어플리케이션 성능 모니터링(Application Performance Monitoring, 이하 APM) 솔루션을 통하여 시스템적으로 수집되게 된다.

마지막 모델 검증(Model Validation)단계에서는 앞서 정의된 성능 모델에 실제 변수 값을 적용하여 검증하는데, 벤치마크나 성능 테스트의 경우 사용자 요청을 시뮬레이션 하는 방법을 통해서 시스템 성능을 검증하거나, 해석 모델의 경우 큐잉 이론에 기반한 성능 모델을 통하여 성능 추이를 예측하게 된다. 웹 기반 시스템의 경우 분석 대상이 되는 서비스는 웹 어플리케이션 서버에서 수행되는 모든 동적 컨텐츠(JSP/Servlet/Web Service)기준이기 때문에, 모든 서비스들에 대한 요청률과 응답시간을 측정해야 하는 번거로움이 따른다. 바꿔 말하면 각 변수들에 대한 측정을 시스템적으로 처리하기 위한 방안이 필요하다는 것이다.

한편 본 성능 모델에서 변수들 간의 상관관계들은 아래 수식을 통해서 정의된다. 해당 수식에 대한 자세한 소개는 본 기고의 범위를 넘어서기 때문에 참고 문헌을 참조하기 바란다.

● **Little’ s Law**

이 법칙은 처리율과 응답 시간간의 상관관계를 의미하며, 웹 기반 시스템에서 액티브 사용자 또는 액티브 서비스 측정에 활용된다. 성능 테스트 수행 시에 대기 시간을 0초로 설정할 경우, 가상 사용자 수는 액티브 사용자와 비례하게 된다.

액티브 사용자(명) = 처리율(tps) X 응답 시간(초)

● **Response Time’ s Law**

이 법칙은 처리율과 동시 사용자간의 상관관계를 의미하며, 벤치마크, 성능 테스트에 필요한 가상사용자 수나 호출 간격에 대한 계산, 확장성 테스트(Scalability Test)를 통한 임계 사용자 예측에 활용된다.

동시 사용자(명) = 처리율(tps) X (응답 시간(초) + 대기 시간(초))

● **Active User’ s Law**

이 법칙은 동시 사용자와 액티브 사용자와의 상관관계를 의미하며, 처리율 측정이 용이하지 않을 경우 동시 사용자 수 측정에 활용된다.

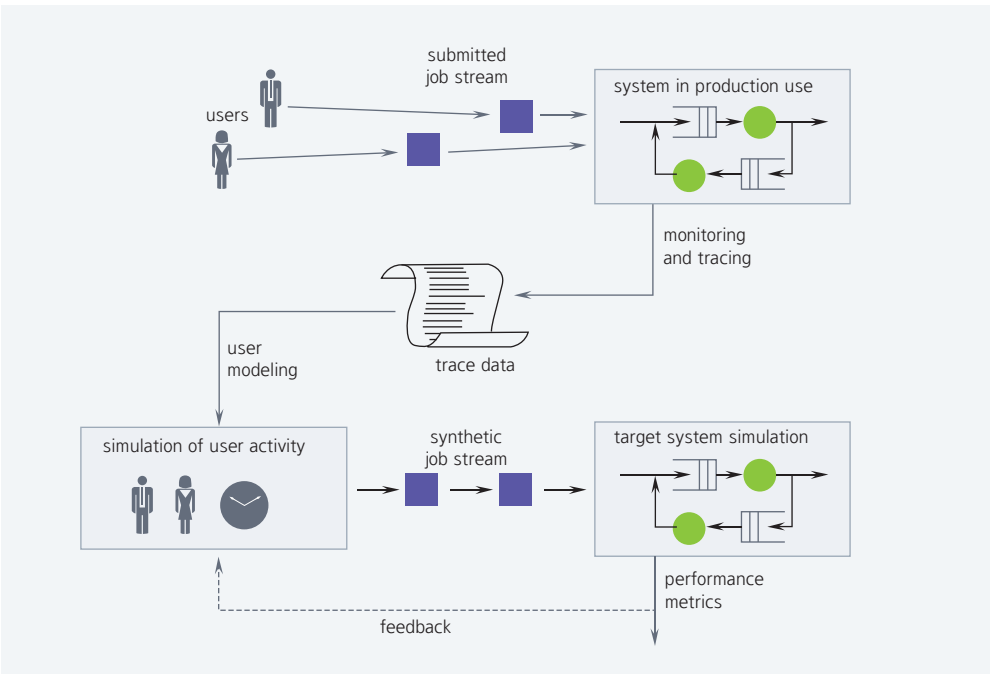
액티브 사용자(명) = 동시 사용자(명) X 응답시간(초) / (응답 시간(초) + 대기 시간(초))

Section 2

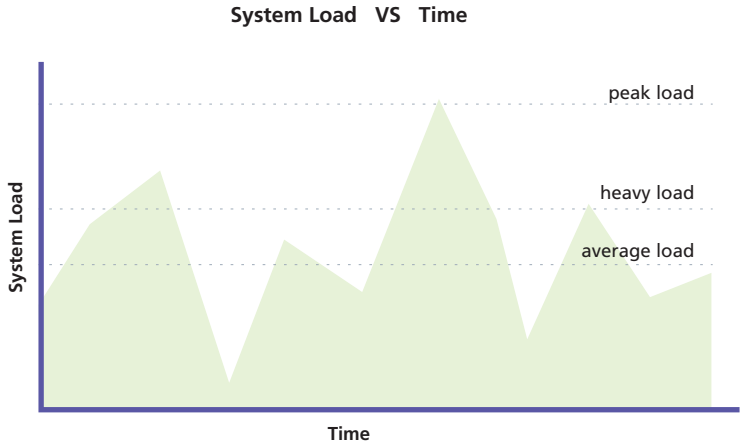
실 세계에서의 성능 분석

앞서 소개한 성능 모델 기반 성능 분석 활동을 수행하기 위해서는 각 성능 모델 변수들에 대한 입력 값을 운영 환경에서 측정(Measurement) 하기 위한 방안이 필요하다. 이들 방안은 성능 모델 유형에 따라 다음과 같이 세가지로 구분해 볼 수 있다. 첫째 하드웨어 선정을 위한 벤치마크에서 성능 목표가 되는 처리율, 응답시간 등을 결정하기 위해서는 현행 시스템에서 관련 항목을 측정한 후, 성능 목표치로 변환할 수 있어야 한다. 둘째 성능 테스트에서 개발 완료된 어플리케이션이 성능 목표가 되는 처리율, 응답시간 만족 하는지 검증하기 위해서는 현행 시스템에서 수집된 성능 관련 기준 데이터(baseline)를 토대로 성능 목표치를 수립할 수 있어야 한다. 셋째 용량 계획에서 성능 목표가 되는 처리율,응답시간 등을 만족하는 적정 규모의 시스템을 유지하기 위해서는 현행 시스템에서 주기적으로 수집된 성능 관련 기준 데이터(baseline)를 토대로 트랜잭션 증가율을 감안하여 해석 모델(Analytic Model)에 적용할 수 있어야 한다.

아래 그림은 성능 모델에 기반한 성능 분석 활동을 도식화 한 것으로, 운영 시스템에 대한 모니터링과 로깅 활동을 통해 성능 모델 변수 실제 값이 측정되고 있음을 나타내고 있다.



여기서 성능 모델 변수 실제 값을 측정하기 위한 모니터링 구간은 사용자가 가장 많이 몰리는 피크 시간대(peak load)로 정해야 한다. 더불어 표본 데이터는 같은 피크 시간대라도 시간 경과에 따라 달라지기 때문에, 여러 차례 측정한 결과에 대한 평균 수치를 사용해야 한다.



웹 기반 시스템의 경우 표본 데이터를 수집하기 위해서는 웹 서버가 제공하는 웹 로그를 사용해야 한다. 웹 로그는 워크로드 분석을 위해서 필요한 다양한 항목들을 포함하고 있기 때문이다. 하지만 시스템 규모가 대형화되고 서버 대수가 증가하고 있는 오늘날의 엔터프라이즈 환경에서는 많은 분석 데이터 량으로 인해 워크로드 분석 용도로 활용하기가 힘들다. 더군다나 분석 작업이 배치로 수행되는 상황에서 실시간 워크로드 측정을 원하는 경우에는 적용자체가 불가능한 실정이다. 최근에는 이러한 제약사항에 대한 대안책으로, APM 솔루션이 제공하는 서비스 모니터링 기능을 활용하는 방안이 각광을 받고 있다.

2.1 APM 솔루션

APM 솔루션 본연의 목적은 정보 시스템에 대한 성능 장애 대응력과 분석 역량을 강화시키는 것이다. 최근에는 이와 같은 기본 목적 이외에 워크로드 모델링(Workload Modeling) 영역 즉 서비스 모니터링까지 제공하려는 움직임이 확산되고 있는데, 이들 정보들을 적절하게 활용할 경우 시스템 성능 분석 과정을 시스템화 할 수 있다.

현재 웹 어플리케이션 서버를 위한 국내 APM 시장에서 독보적인 위치를 확보하고 있는 제니퍼 솔루션도 기본적인 웹 어플리케이션 성능 모니터링 기능 외에 다양한 유형의 서비스 모니터링 기능들을 제공하고 있다.

- 동시사용자 현황
- 액티브 서비스 현황
- 업무 요청률 현황
- 업무 처리율 현황

- 평균 응답 시간 현황
- 평균 대기 시간 현황
- 평균 CPU 사용률 현황

Section 3

결론

우리는 지금까지 정보 시스템의 성능 분석을 위해서 요구되는 성능 이론/모델에 대한 소개와 더불어 실 세계에서의 성능 분석 방법에 대해서 알아보았다.

오늘날의 기업들은 변화 무쌍한 비즈니스 요구사항에 대처하기 위한 노력의 일환으로 고 성능의 정보 기술 체계를 구축해야 한다. 이와 더불어 구축된 시스템이 비즈니스가 요구하는 성능 요건에 부합되는지를 검증하기 위한 방안도 철저히 준비해야 한다.

본 기고에서 소개한 성능 분석 모델이 고 성능의 정보 기술 체계 구축에 참여하는 많은 IT종사자들에게 큰 도움이 되었으면 한다.

〈참고문헌〉

1. “SAP AG” Theory and Practice of Sizing SAP Software
2. “Peter J. Denning” Queuing Network of Computes
3.”Andy Lee” “Mathematical Approach for Web-based System Performance”
4.” Dror G. Feitelson” “Workload Modeling for Computer Systems Performance Evaluation”

Section 4

저자에 대하여

박성훈 컨설턴트는 BTO 컨설팅의 애플리케이션 품질 관리 대표 컨설턴트로 활동하고 있다. 다년간 엔터프라이즈 환경하에서 성능 벤치마크, 성능 테스트, 품질 관리 등의 업무를 담당하였다. BTO 컨설팅 이전에는 한국 HP, 머큐리 인터랙티브 코리아 등에서 프로페셔널 서비스 컨설턴트로 역할을 수행한 바 있다.

박성훈 컨설턴트(Bruce Park),Consultant, mail : bruce@jennifersoft.com

about JenniferSoft...

JenniferSoft, Inc. is the software vendor company with expertise in application performance monitoring and performance problem resolution, providing Application Performance Management (APM) solutions and services to enterprise companies around the world.
Technology is foremost of what's valued in JenniferSoft, as we strive to bring to the market the latest and best technology available.
We think that software solution should not be just a mesh of form and functions but it should be designed with its user in mind, each component formed with intent to elevate the experience of its user. JenniferSoft combines latest technology in APM with field-tested expertise and experience to bring to our customers a well-balanced solution that is both advanced in features and practical. With vision of combining technology and experience into one, JenniferSoft pledges to continue focuing on R&D to develope world class solution.

JENNIFER
JENNIFERSOFT, Inc.