

July 2011

대규모 프로젝트 하에서의 애플리케이션 성능 관리(APM) 적용 방안

Table of Contents

Executive Summary

| | | |
|--------------------|--|----|
| Section 1 | 애플리케이션 관리 (Application Management) | 04 |
| Section 2 | 애플리케이션 성능 모니터링 (Application Performance Monitoring, 이하 APM) | 06 |
| Section 2.1 | 최종 사용자 모니터링 (End-user experience Monitoring) | 06 |
| Section 2.2 | 애플리케이션 컴포넌트 상세 모니터링 (Application component deep-dive monitoring) | 08 |
| Section 2.3 | 애플리케이션 성능 관리 데이터베이스 (Application performance management database) | 09 |
| Section 2.4 | 사용자 정의 트랜잭션 프로파일링 (User-defined transaction profiling) | 10 |
| Section 2.5 | Application component discovery and modeling | 11 |
| Section 2.6 | 애플리케이션 서비스 모니터링 (Application Service Monitoring) | 12 |
| Section 3 | 결론 | 13 |
| Section 4 | 저자에 대하여 | 14 |

Executive Summary

Challenge

비즈니스 상황이 급변함에 따라 애플리케이션 환경이 복잡해지고 성능 이슈가 빈번히 발생하고 있다. 따라서 기업들은 이러한 상황에 대처하기 위해서 애플리케이션 성능을 모니터링하고 이슈를 조기에 대응하기 위한 체계에 대해서 필요성을 절감하고 있다.

Opportunity

애플리케이션 성능 모니터링 솔루션을 효율적으로 활용함으로써, 기업들은 최종 사용자에게 대한 성능 분석, 복잡한 비즈니스 트랜잭션에 대한 성능 분석, 애플리케이션 매핑 분석, 애플리케이션 서비스 모니터링 등에 대한 대응책을 마련할 수 있다.

Benefits

본 백서는 기업들이 대형 프로젝트에서 필요로 하는 여러 애플리케이션 성능 모니터링 요건들에 대한 대처 방안들을 실무사례와 함께 소개함으로써 APM 솔루션을 효율적으로 사용할 수 있게끔 가이드 한다.

오늘날 비즈니스 환경은 국내외를 막론하고 하루가 다르게 변화하고 있으며, 기업들은 시장 상황의 변화를 지속적으로 감지하고 대처하는데 총력을 기울이고 있다. 이러한 변화는 기업들의 비즈니스 프로세스에 대한 신속한 혁신과 더불어 글로벌화 시대의 국제 표준에 부합되는 거버넌스, 표준화 그리고 가시성을 요구한다.

하지만 시장의 활발한 움직임에 비해, 대다수의 기업들이 보유하고 있는 노후화된 정보 시스템들은 시장의 요구에 효율적으로 대응할 수 있는 유연성과 확장성을 제공하지 못한다. 따라서 기업들은 지속적인 시스템 고도화 작업을 통하여 시장의 요구에 부분적으로 대응하거나, 차세대 시스템 형태의 새로운 정보 시스템을 구축하는 방향으로 대처하고 있다. 최근 들어 금융권을 기점으로 차세대정보 시스템 구축 프로젝트가 활기를 띠며 불을 이루고 있는 것도 이에 대한 반응이라 볼 수 있다.

이러한 차세대 시스템 구축 프로젝트는 시장 대응력을 향상시키기 위한 효율성(efficiency)과 민첩성(Agility)을 제공해야 한다. 여기에 더해 소스 코드의 재사용성(Reusability)을 통한 비용 효율성까지 고려해야 한다. 따라서 이러한 요건들을 대응하기 위한 새로운 아키텍처 체계가 필요하게 되었고, 현재 서비스 지향 아키텍처(SOA)는 효과적인 대안책으로 각광받고 있다. 서비스 지향 아키텍처가 그 특성상 비즈니스 유연하고 유지보수성이 뛰어난 동시에 소스 코드 재사용이 가능한 비즈니스 지향성을 기반으로 고립된 시스템들 간의 장벽을 제거함으로써 비즈니스 민첩성을 제공할 수 있기 때문이다.

하지만 서비스 지향 아키텍처는 이러한 많은 장점에도 불구하고, 서비스 재사용성을 위한 비즈니스 로직 모듈화 및 분산처리로 인하여 애플리케이션 아키텍처 환경을 매우 복잡하게 만들기 때문에 엔터프라이즈 IT 관리 방식에 대한 큰 변화를 초래한다. 다시 말해 엔터프라이즈 IT 관리 방식이 전통적인 인프라 구성요소(서버, 네트워크, 스토리지 등) 중심에서 애플리케이션 중심으로 전환되어야 한다.

본 기고에서는 최근 대대적인 정보 시스템 개편에 나서고 있는 차세대 프로젝트의 사례를 들어 대규모 프로젝트에서 적용 가능한 애플리케이션 중심의 관리 방안과 중요성이 점점 더 커지고 있는 애플리케이션 성능 모니터링에 대해서 소개하고자 한다.

Section 1

애플리케이션 관리(Application Management)

애플리케이션 관리는 IT에 종사하고 있는 사람들에게는 친숙한 용어이다. 그러나 본 용어에 대해 정확하게 알고 있는 사람은 엔터프라이즈 IT 관리에 종사하는 사람들 속에서도 찾아내기가 쉽지 않다. 우선 애플리케이션 관리에 대한 정의는 매우 다양하다. 온라인 백과사전인 위키피디어(Wikipedia)에서는 '애플리케이션 관리'는 '거버넌스, 개발, 운영 영역을 통해서 애플리케이션 라이프사이클을 지속적으로 관리하는 프로세스'로 정의하고 있다. ITIL(IT Infrastructure Library)에서 정의한 용어도와 유사하다. 비즈니스 요구로부터 최종 소프트웨어 폐기에 이르는 전체 애플리케이션 라이프사이클

을 관리하는 프로세스로 정의한다. 보다 현실적인 접근을 위해서 IT 관리 솔루션 시장에서 통용되고 있는 애플리케이션 라이프사이클 관리(Application Lifecycle Management) 솔루션 라인업으로 매핑하면 아래와 같이 분류해 볼 수 있다.

1. 거버넌스(Governance)

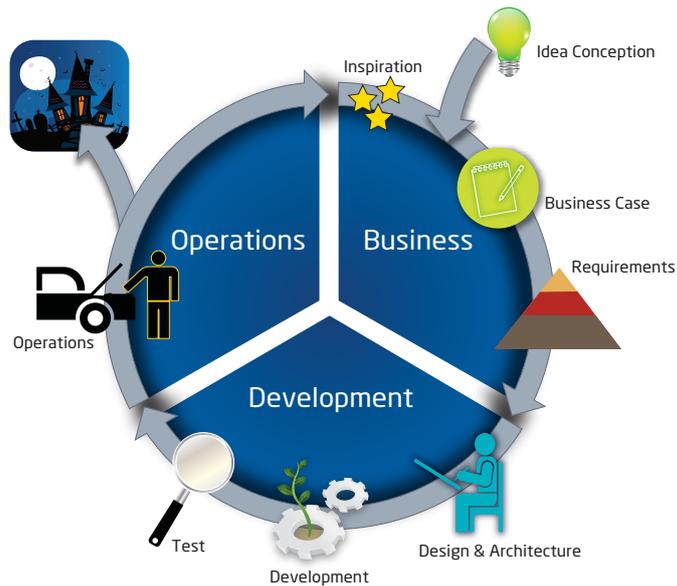
- 프로젝트 및 포트폴리오 관리(Project and Portfolio Management)

2. 개발(Development)

- 요구사항 정의/관리(Requirements Definition/Management)
- 형상 관리(Software Change and Configuration Management)
- 테스트 관리(Test Management)

3. 운영(Operations or Maintenance)

- 릴리즈 관리(Release Management)
- 운영 관리(Operations Management)
- 애플리케이션 성능 모니터링(Application Performance Monitoring)



이미지 출처 “www.netsoltech.com”

전통적인 의미의 애플리케이션 관리는 SDLC(Software Development Lifecycle Management)와 같은 개념으로 이해되어 왔는데, 이러한 접근방식은 애플리케이션 관리를 개발영역으로만 한정하게 된다. 따라서 애플리케이션 관리에 대한 주요 관심사가 각 개발 영역에서 사용되고 있는 포인트 솔루션들의 기능과 프로세스 연계에만 집중되는 단점이 존재한다. 따라서 애플리케이션 관리 환경을 도입하려는 조직에게 시장 대응력을 향상 시키기 위해서는 개발 영역과 더불어 거버넌스 영역과 운영 영역을 포함한 통합적인 관리 체계는 반드시 필요하다. 즉 거버넌스 영역에서 개발 대상이 되는 비즈니스 요건을 검토하여 시스템 구축이 필요할 경우에는 프로젝트 성과에 대한 사전 평가를 포트폴리오 관리를 통해서 시뮬레이션 하게 된다. 프로젝트 착수 이후부터는 범위, 일정, 예산 등에 대한 사업관리들을 프로젝트 관리를 통해서 수행하게 된다. 반면 운영 영역에서는 개발 영역에서 구축된 시스템들을 이관하고 운영하기 위한 각종 운영 관리 절차들을 수립하고 시스템 성능 및 가용성 현황을 모니터링하기 위한 애플리케이션 성능 모니터링 활동을 수행하게 된다.

애플리케이션 성능 모니터링 (Application Performance Monitoring, 이하 APM)

대다수의 기업들은 과거 1990년대 중반부터 애플리케이션 성능 모니터링에 대한 필요성을 체감하여, 엔터프라이즈 IT 운영 관리의 일환으로 애플리케이션들을 모니터링하기 시작했다. 초기 접근 형태는 애플리케이션을 개별적인 모니터링 대상으로 보는 게 아니라 단지 인프라의 한 구성요소로서 판단했다. 따라서 주로 로컬 시스템에 국한된 애플리케이션 관련 성능 정보들을 수집하여 모니터링 하였다. 즉 에이전트에서 주기적으로 대상 시스템에 대한 성능 정보들을 수집한 다음, 솔루션 벤더 또는 사용자에게 의해서 정의된 로컬 리소스 사용률(resource consumption), 수행시간(latency)과 관련된 임계치(threshold)들과의 비교 분석을 통해서 성능 문제들에 대응하는 수준이었다,

이러한 방식은 애플리케이션 코드가 한곳에 집중되어 있고 다른 애플리케이션과의 경계(Tier)가 분명한 정보 시스템 환경에서는 효율적이지만, 모듈화, 분산환경을 지향하는 대규모 시스템 환경에서는 많은 제약사항이 따른다. 즉 성능 모니터링을 위해서는 많은 에이전트가 필요하고, 또한 각 에이전트에서 수집된 방대한 정보들간의 연계성 및 추적성도 하기 때문에 기존의 애플리케이션 성능 모니터링 방식에 대한 변화가 불가피 하였다.

여기서 대규모 프로젝트 상에서 통상적으로 직면하게되는 애플리케이션 성능 모니터링 관련 주요 이슈 및 대응방안을 살펴보면 아래와 같이 5가지로 요약된다.

1. 최종 사용자 모니터링 End-user experience Monitoring

최종 사용자 관점에서의 주요 서비스에 대한 성능(응답시간) 및 가용성 현황은 구축대상 시스템에 대한 고객만족도와 직결되는 것으로서 통상적으로 프로젝트 분석 단계에서 비기능 요구사항 형태로 정의 된다. 이들 요구사항들은 목표 달성 여부를 정량적으로 파악할수 있게끔 실제 측정이 가능한 수치 형태로 제공되어야 한다. 그러므로 목표는 기존 운영되고 있는 정보 시스템에서 주요 서비스에 대한 성능 수치를 측정하여 지표화 하거나, 경쟁사 시스템과의 비교 분석을 통해서 수립되어야 한다. 하지만 두 방법 모두 구축 대상 시스템과의 아키텍처가 동일하지 않고 업무 내용도 차이가 날 수 있기 때문에 신뢰도가 높은 수치를 뽑아내기에는 제약사항이 따르게 마련이다. 따라서 반드시 비즈니스 현업 부서와의 충분한 협의를 통해서 절충안을 만들어야 한다.

본 프로젝트의 경우도 대부분의 유사 프로젝트가 그러하듯이 프로젝트 분석 단계에서 주요 서비스에 대한 성능(응답시간) 및 가용성 관련 요구사항이 정의되지 않아서, 테스트(성능 테스트) 단계에서 요구사항 분석 및 목표 수립 작업에 착수했다. 하지만 분석 대상이 되는 현행 시스템의 년중 피크치가 연말 기준이라 측정 자체가 불가능한 상황에다가, 현재 기준치를 가지고 피크 보정을 하기에는 논리적인 기준치가 없을 뿐더러 현행 시스템의 성능수치가 너무 좋지 않은 상황이었다. 여기에 더해 비즈니스 현업 부서의 이해 부족으로 인해 목표 결정에 수많은 난관이 있었다.

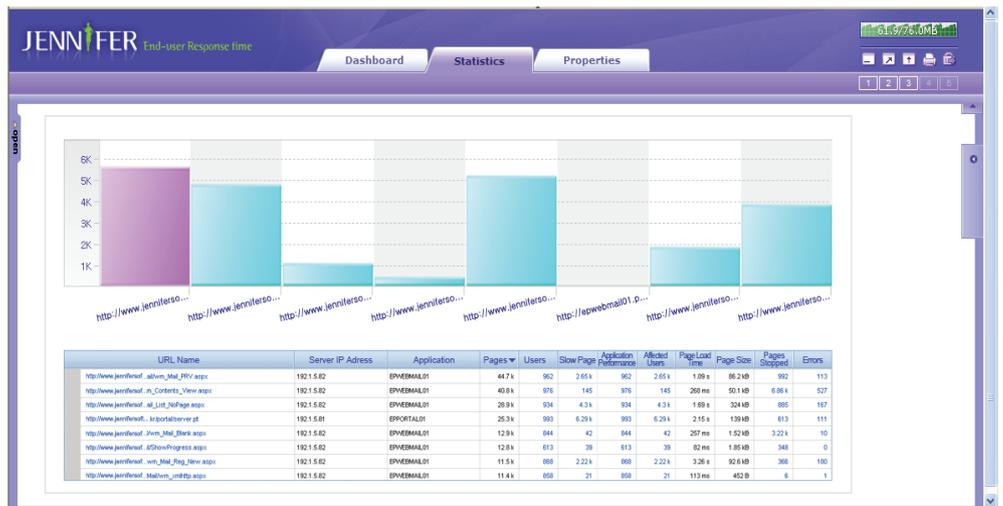
주요 서비스에 대한 성능(응답시간) 및 가용성 관련 요구사항이 이러한 여러 난관을 거쳐서 정의되면, APM 솔루션을 통하여 아래와 같은 2가지 방식을 통해서 모니터링된다.

- **액티브 에이전트(Active Agents)** - 액티브 에이전트 방식은 사전 정의된 시간 간격을 두고 가상 트랜잭션(synthetic transaction)을 생성하여 실제 사용자의 트랜잭션을 재현한다.
- **패시브 에이전트(Passive Agents)** - 패시브 에이전트 방식은 실제 사용자의 트랜잭션을 데스크톱 또는 서버에서 측정하여 성능을 검증한다.

액티브 에이전트 방식은 실제 사용자의 트랜잭션을 에뮬레이션 하기 위하여 가상 환경내에서 가상 트랜잭션을 생성하여 성능(응답시간) 및 가용성을 측정 한다. 하지만 실제 사용자 환경과의 차이로 인해 측정 결과에 대한 신뢰도가 낮아질 확률이 매우 높다. 따라서 낮은 신뢰도를 극복하기 위하여 추가 비용이 들더라도 실제 사용자들이 사용하고 있는 각 ISP 망 별로 별도로 가상 환경을 구축해야 한다.

패시브 에이전트 방식은 데스크톱, 서버 또는 어플라이언스(Appliance)에서 실제 사용자가 수행하는 트랜잭션들을 측정하고 지표화 한다. 따라서 액티브 에이전트의 단점 즉 측정 결과에 대한 낮은 신뢰도는 극복할 수 있지만, 측정된 결과치가 비즈니스 서비스와의 매핑이 용이하지 않다는 숙제를 떠안고 있다.

최근에는 두 가지 측정 방식의 장점을 유지한채 단점을 보완한 하이브리드 방식이 대세일 것으로 보이는데, 이 방식은 실제 사용자 트랜잭션의 응답시간과 가용성 정보를 사용자 PC에 설치된 HTTP 쿠키 또는 액티브 에이전트와 패시브 에이전트간의 연결 정보를 이용하여 측정하고, 지표화하게 된다.



2. 애플리케이션 컴포넌트 상세 모니터링 Application component deep-dive monitoring

앞서 소개한 바와 같이 최근의 엔터프라이즈 애플리케이션 아키텍처의 주요 특징은 시장의 역동성과 복잡성에 대한 대응책으로 전사 IT 표준을 통한 동질화, 효율성, 민첩성을 제공하고, 기존 투자된 자산에 대한 재사용성을 위해서 서비스 지향 아키텍처를 채택하고 있다는 것이다. 이러한 환경에서는 애플리케이션 비즈니스 로직을 독립적인 서비스로 설계하고 노출되는 컴포넌트들을 분산 배치하는 형태를 띄게 된다. 그러므로 아키텍처 구성은 매우 유연해지는 반면 시스템의 복잡도가 높아지기 때문에 성능 장애 사항을 조기에 대응하거나 근본 원인(Root Cause)을 알아내기가 매우 어려워지게 된다.

이러한 시대적 상황에 힘입어 많은 기업들은 비즈니스 서비스와 연관된 애플리케이션 컴포넌트에 대한 상세 모니터링 요구사항들을 제시하게 되었고, 이에 대한 대응책으로 APM 솔루션이 태동하게 된 것이다. 초기에 APM 솔루션들은 Java EE, .NET 미들웨어내에 에이전트를 설치하여 바이트 코드 인스트루멘테이션(Byte code instrumentation)과 같은 프로파일링 기법을 통하여 애플리케이션 상세 모니터링을 수행 하였으나, 현재는 미들웨어 뿐만 아니라 비즈니스 서비스와 관련된 모든 애플리케이션 컴포넌트들을 분석하기 위한 통합 모니터링 체계로 진화 하고 있는 상황이다.

서비스 지향 아키텍처를 지향하고 있는 본 프로젝트의 애플리케이션 아키텍처도 엔터프라이즈 서비스 버스(Enterprise Service Bus, 이하 ESB), 비즈니스 프로세스 관리(Business Process Management, 이하 BPM)등의 여러 분산 처리 기술들을 통하여 분산 컴퓨팅 플랫폼을 목표로 하고 있기 때문에, 각 비즈니스 서비스의 구간별 처리 현황 및 성능 장애 분석 및 장애 조기 대응을 위하여 애플리케이션 컴포넌트 상세 모니터링 기술은 필수적이었다

통상적으로 애플리케이션 컴포넌트 상세 모니터링을 하기 위해서는 분석대상이 되는 개체 즉 클래스 또는 패키지를 개별적으로 설정해줘야 하는데, 개체 수가 많아지게 되면 분석 대상이 되는 애플리케이션 자체에 영향을 줄 수 있기 때문에 적절한 설정 기준이 필요 하다.일반적으로는 시스템 운영 시 기본 옵션으로 사용하다가 성능 장애시 상세 분석을 위해서 분석하고자 하는 개체를 개별적으로 설정하는 다이나믹 프로파일링 방식을 추천한다.



3. 애플리케이션 성능 관리 데이터베이스 Application performance management database

대규모 프로젝트를 통해서 구축되는 시스템들은 점차적으로 시스템 규모가 더욱 더 커질 뿐만 아니라, 최근 비용 효율성으로 인하여 유행처럼 번지고 있는 클라우드 컴퓨팅 도입으로 인해 관리 대상서버 대수도 기하급수적으로 증가하고 있다. 결국 이러한 시대적 조류는 APM 솔루션에서 처리해야 하는 정보량의 급증과 더불어 성능 분석 및 성능 이슈 해결을 위한 데이터 분석 작업량도 배가 시키기 때문에, 애플리케이션 성능 관리 데이터베이스에 대한 중요도를 더욱 더 가중시키고 있다.

여기서 APM 솔루션을 통한 성능 분석 및 성능 이슈 해결 과정을 예를 들어 데이터 분석 작업이 어떻게 진행되는지 알아보기로 하자.

운영 담당자는 최종 사용자 모니터링(End-user Experience Monitoring) 데이터 수집 과정에서 특정 서비스에 대한 응답시간이 지연되고 있는 것을 확인하였다. 운영 담당자는 해당 성능 이슈에 대한 근본 원인 분석(Root cause Analysis)을 위해서 사용자 정의 트랜잭션 프로파일링(Use-defined transaction profiling)을 수행하였다. 여기서 사용자 정의 트랜잭션 프로파일링은 특정 서비스에 포함된 최상위 서비스 요청(Request)을 기준으로 거래 추적을 실시하고, 응답시간 지연을 일으키는 최종 병목 지점을 제공해 준다. 그 후 운영 담당자는 애플리케이션 컴포넌트 발견 및 모델링(Application component discovery and modeling) 기능이 제공하는 토폴로지 뷰(Topology View)를 통해서 해당 병목 지점이 발생했던 서버가 A 데이터 센터에 위치한 특정 서버인 것을 확인한 후 해당 서버에 대한 WAS 구성 정보를 확인한다. 최종적으로 운영 담당자는 애플리케이션 컴포넌트 상세 모니터링(Application component deep-dive monitoring) 기능을 통해서 최종 병목 지점이 특정 메소드와 관련되어 있는 것을 확인하고 관련 개발자에게 통보하여 코드를 개선작업을 요청하게 된다.

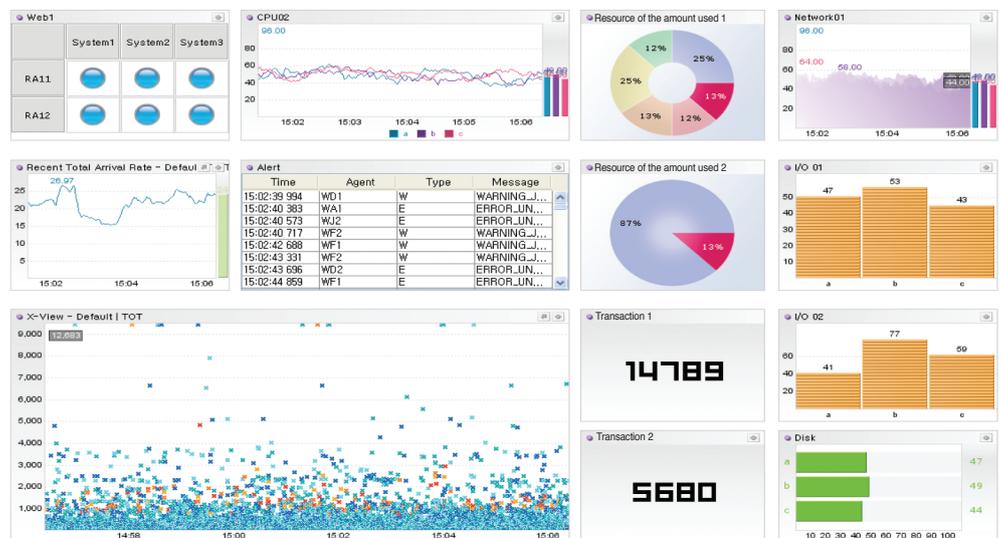


위의 예제에서 소개한 바와 같이 성능 분석 및 성능 이슈 해결을 위해서는 다량의 데이터 분석 작업이 필요하기 때문에, 효율적으로 데이터를 분석하고 처리하기 위한 방안은 반드시 필요하다. 더군다나 국내 엔터프라이즈 아키텍처 환경에 클라우드 컴퓨팅을 도입하려는 많은 움직임이 포착되고 있는 상황에서는 관리 대상 서버 대수가 적게는 수십 대에서 많게는 수백 대를 넘어설 수 있기 때문에 APM 솔루션에 대한 대용량 처리 방안은 현 시점에서 반드시 구현되어야 한다.

4. 사용자 정의 트랜잭션 프로파일링 User-defined transaction profiling

최근 서비스 지향 아키텍처 도입을 검토하고 있는 고객들은 APM 솔루션과 더불어 End-To-End 구간에 대한 비즈니스 트랜잭션을 프로파일링 하고 모니터링 할 수 있는 비즈니스 트랜잭션 관리 (Business Transaction Management) 솔루션에 대해서 많은 관심을 가지고 있다. BTM 솔루션은 비즈니스 트랜잭션 모니터링 (Business Transaction Monitoring), 애플리케이션 트랜잭션 프로파일링 (Application Transaction Profiling), 사용자 정의 트랜잭션 프로파일링 (User-defined transaction profiling) 등으로 불려왔는데, 과거에는 APM 진영과는 별도로 진행되다가 최근에 들어서는 가트너, 포레스터와 같은 시장 조상 기관들에 의해서 APM 솔루션의 확장 개념으로 통합되고 있는 추세이다.

이처럼 BTM 솔루션이 시장에서 많은 관심을 끌고 있는 이유는 분명하다. 즉 분산 컴퓨팅을 지향하는 서비스 지향 아키텍처의 경우 비즈니스 트랜잭션 관점에서의 추적을 위하여 관련 인프라 및 애플리케이션들간의 처리 흐름 및 애플리케이션 토폴로지에 대한 동적 매핑 기술이 필요하다. 하지만 현재 시장에서 널리 활용되고 있는 SMS/NSM, APM, DB Monitoring 솔루션들만 가지고는 이러한 요건에 대해서 대응할 수가 없는 실정이다.



서비스 지향 아키텍처 환경을 지향하고 있는 본 프로젝트에서도 특정 비즈니스 트랜잭션을 처리하기 위해서는 여러 계층으로 분리된 비즈니스 티어를 거쳐서 처리가 되어야 한다. 하지만, 비즈니스 트랜잭션에 이슈가 발생하거나 성능이 저하될 경우, 근본 원인에 대한 분석을 위해서는 해당 트랜잭션이 어느 티어에서 문제가 되었고, 해당 티어와 관련된 시스템, 네트워크, 애플리케이션 서버, 데이터 베이스 서버 등에 대한 성능 상태 정보가 필요하다. BTM 솔루션은 성능 이슈를 발생시킨 티어에 대한 근본 원인을 별도로 분리(isolate)하여 분석 함으로써 장애 발생부터 문제 해결까지 소요되는 시간을 최소화 할 수 있고, 과정 자체를 단순화 시킬 수 있는 것이다.

현재 국내 IT 환경에서는 BTM 솔루션이 종단간 거래 추적 시스템(End-to-End Transaction Tracking) 또는 전사 애플리케이션 트랜잭션 관리 시스템(Enterprise Application Transaction Management) 이라는 명칭으로 몇몇 업체들에 의해 시장에 소개되고는 있지만, 아직까지 APM 솔루션만큼 보편화 되기에는 상당히 많은 시간이 소요될 것으로 보인다.

4. 애플리케이션 컴포넌트 디스커버리 및 모델링 Application component discovery and modeling

애플리케이션 상세 모니터링 및 사용자 정의 트랜잭션 프로파일링을 위해서는 해당 트랜잭션과 연관되어 있는 인프라 및 서비스 컴포넌트에 대한 구성 정보가 필요하다. 예를 들어 수십 또는 수백대의 서버로 구성되어 있는 로그인 트랜잭션이 느려질 경우, APM 솔루션은 본 트랜잭션과 관련된 서버 및 서비스 컴포넌트에 대한 상세 정보를 토폴로지 뷰를 통하여 직관적으로 제공해야 한다.

현재 IT 업계에서 애플리케이션 컴포넌트 디스커버리 및 모델링을 지원하는 솔루션들은 아래와 같은 2 가지 형태로 제공된다.

1. CMDB 기반의 ADDM(Application Discovery and Dependency Mapping)
2. SOA 토폴로지 뷰(Topology View)

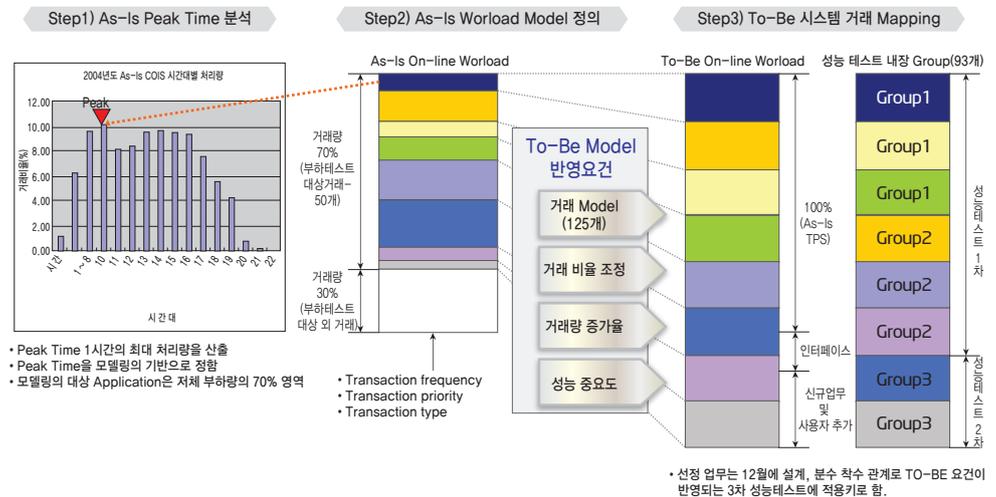
ADDM(Application Discovery and Dependency Mapping)은 IT 자산 관리 측면에서 시스템 관련 기본 인프라들과 애플리케이션 간의 복잡한 종속성을 시각화하여 비즈니스 가용성을 최적화하고 문제 해결 프로세스를 단축시킨다. 반면 SOA 토폴로지 뷰는 SOA 관련 메타 정보를 이용하여 컴포지트(Composite)를 구성하고 있는 서비스, 레퍼런스, 컴포넌트 간의 상관관계를 정적 또는 동적으로 시각화해 준다. 따라서 컴포넌트 간의 서비스 흐름 분석과 더불어 성능 이슈 분석 시 매우 유용하게 사용될 수 있다.

5. 애플리케이션 서비스 모니터링 Application Service Monitoring

차세대 프로젝트를 통하여 대대적으로 개편되는 정보 시스템 환경은 기존에 운영되고 있던 레거시 시스템이 처리하고 있던 업무량을 원활하게 처리 할 수 있어야 한다. 이와 더불어 향후 몇 년간의 업무 확장성을 고려하여 정보 시스템을 구축해야 하기 때문에 고객과 사전 협의된 목표 성능치 도달 여부를 반드시 검증해야 한다.

앞서 지적하였듯이 구축 대상 시스템의 목표 성능치는 개발 프로세스의 분석단계에서 요구사항 수집 단계를 통해 비기능 요건으로 정의되는데 애플리케이션의 기능 요건과는 달리 추적성이 떨어지는 경우가 비일비재하다. 즉 프로젝트 초기에 설정된 목표 성능치는 프로젝트가 진행되면서 비즈니스 환경이 바뀌어 새로운 요구사항이 추가되거나 기존 요구사항이 삭제 될 수도 있다. 따라서 프로젝트 후반부로 갈수록 목표 성능치에 대한 추적이 불가능한 경우가 많다. 이러한 특성 때문에 프로젝트 초기에 목표 성능치를 설정 할 때는 변경 사항에 대한 충분한 고민과 협의를 통해서 고려되어야 한다.

아래 그림은 목표 성능치를 설정하기 위해서 필요한 액티비티들을 단계적으로 보여 주고 있는데 이는 다음과 같다.

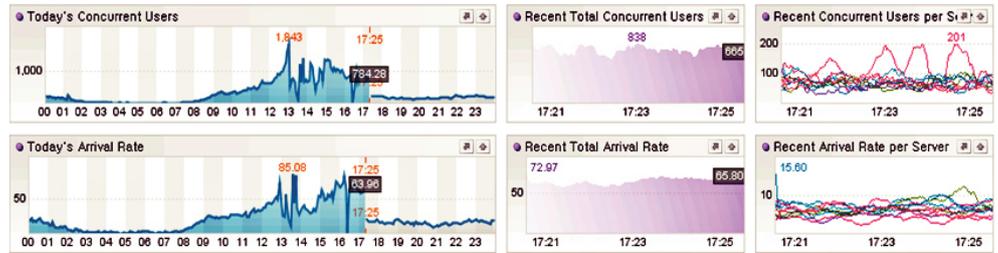


먼저 구축 대상 시스템의 부하량 모델(Workload Model)은 기존 시스템의 피크 일자의 부하량을 주기적으로 분석하여 수립된다. 그런 다음 성능 테스트 대상 거래 업무는 부하량 모델을 분석하여 거래량 대비 상위 70~80%를 점유하고 있는 대상 거래 업무들을 기준으로 선정된다. 여기서 선정 대상에서 제외된 나머지 거래 업무들은 고객과의 협의를 통해 업무 빈도, 업무 중요도, 업무 유형에 따라 다시 선정작업을 진행하여 최종 성능 테스트 대상 업무에 포함될 수 있다. 본 과정을 통해서 최종 결정된 성능 테스트 대상 거래 업무들은 TO-BE 모델 수립 시 계획된 거래 비율 조정, 거래량 증가율, 성능 중요도들을 반영하여 최종 성능 목표치로 수립된다.

일단 목표 성능치까지 설정되었으면 이를 검증하기 위한 테스트 환경 구축과 더불어 성능 테스트를 수행해야 한다. 한편 시스템 오픈 이후의 운영 상황과 거의 유사하게 재현 할 수 있는 성능 테스트 솔루션과 함께 애플리케이션 서비스 모니터링(Application Service Monitoring) 솔루션은 반드시 필요하다.

이유인즉은 애플리케이션 서비스 모니터링 솔루션은 성능 테스트 솔루션을 통하여 생성된 목표 성능치 도달 여부를 Java EE, .NET 기반의 미들웨어 레벨에서 손쉽게 파악할 수 있게 해준다. 만약 목표 성능치에 도달 하지 못할 경우에는 애플리케이션 컴포넌트 상세 모니터링 기능을 통하여 성능 개선 작업을 단순화 시켜 준다. 아래 그림은 제니퍼 솔루션에서 제공하고있는 애플리케이션 서비스 모니터링 화면으로 서비스 레벨의 성능 분석 시 유용하게 사용할 수 있다.

제니퍼 애플리케이션 서비스 모니터링 그래프



애플리케이션 서비스 모니터링은 여기에 더해 대상 시스템의 모니터링 결과를 주기적으로 수집하여 정보 시스템 투자 계획을 수립하기 위한 용량 계획 데이터도 제공해 준다.

Section 3

결론

우리는 지금까지 차세대 시스템 구축 프로젝트와 같은 대규모 프로젝트 하에서 적용 가능한 애플리케이션 중심의 관리 방안 특히 애플리케이션 성능 모니터링 측면에 대해서 알아보았다.

본 기고에서 소개했듯이 변화 무쌍한 비즈니스 요구사항에 대처하기 위한 대안으로 각광 받고 있는 서비스 지향 아키텍처는 아키텍처 복잡성으로 인해서 성능 관련 이슈들을 조기에 대응 하고 장애 원인을 빠른 시간에 찾기에는 많은 제약사항이 있다.

결론적으로 APM솔루션은 제한된 프로젝트 일정과 인력 범위 내에서 다양한 성능 모니터링 요건 및 성능 관련 이슈들을 효율적으로 대처하기 위해서 반드시 필요하다. APM솔루션이 프로젝트 성공을 보장하는 지름길이라고 보더라도 무방하지 않을까 생각해본다.

(참고 문헌)

1. "Dirk Kraefzig" Enterprise SOA
2. "Gartner Press" Market Scope for Application Lifecycle Management
3. "Jean-Pierre Garbani" The IT End User Experience Monitoring Software Market
4. "Jean-Pierre Garbani" Competitive Analysis: Application Performance Management And Business Transaction Monitoring

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 focusing on R&D to develop world class solution.

