

초대형 워크플로우 시스템에서의 워크케이스 생성 기법에 대한 구현 분석

안형진[○] 김광훈

경기대학교 전자계산학과 워크플로우 연구실
{zzanggu1107[○], kwang}@kyonggi.ac.kr

Analysis of Workcase Creation's Management Schemes for Very-Large Scale Workflow System

Hyungjin Ahn[○] Kwanghoon Kim

Dept. of Computer Science, Kyonggi University

요 약

기업 및 조직 내외에서 이루어지고 있는 대량 작업들을 처리하기 위하여 기존의 워크플로우 시스템보다 더욱 대형화된 처리를 지원하는 워크플로우 시스템의 등장이 요구되어지고 있다. 기업 인프라는 급격히 발전하고 있지만 그에 반해 워크플로우 시스템은 처리적인 측면에서 이를 뒷받침하지 못하고 있다. 현재의 시스템 환경의 한계를 극복하며 대량의 비즈니스 작업들을 처리해내기 위해서는 이에 대한 극복 방안이 마련되어야 한다. 본 논문에서는 극복 방안의 일환으로 다수 사용자들의 요청에 의해 발생하는 프로세스 인스턴스인 워크케이스에 대한 효과적인 생성 기법에 대해 제시하고 이러한 방법을 기반으로 하여 실제적인 구현 및 테스트를 통해 결과를 도출해내고자 한다.

1. 서 론

오늘날의 기업 및 조직 내외에서 이루어지는 비즈니스 프로세스 업무 처리 환경들은 갈수록 거대화되어 가고 있다. 특히 네트워크 및 인터넷 인프라의 비약적 발전에 의해 자동화 업무가 조직의 업무 처리에서 더욱 큰 비중을 차지하게 됨에 따라 기존의 워크플로우 시스템이 가지고 있던 기능보다 더 나은 기능을 가져야 한다는 인식이 대두되고 있다. 이와 같은 대규모의 작업들을 효율적으로 처리하기 위해 등장한 시스템이 초대형 워크플로우 시스템이다. 초대형 워크플로우 시스템은 대규모의 비즈니스 프로세스들을 처리해낼 수 있는 시스템이어야 하기 때문에 시스템 구축 시 고려되어야 할 점들이 많은데 그 중에서 시스템의 리소스를 얼마만큼 효율적으로 관리, 사용할 수 있는가는 점이 중요한 논의의 대상이 된다. 하드웨어의 발전 속도가 빠르다고는 하나 리소스의 관리 공간은 유한적이기 때문에 그로 인한 한계는 당연히 존재할 수 밖에 없다. 따라서, 본 논문에서는 워크플로우 프로세스 인스턴스인 워크케이스에 초점을 맞추어 초대형 워크플로우 시스템에서의 워크케이스 생성 기법에 대한 두 가지 방안을 제시하고 이 기법들을 토대로 구현을 통해 워크케이스 생성 반응 시간을 도출하여 어떠한 기법이 초대형 워크플로우 시스템에 적합한지를 기술하고자 한다. 2장은 관련 연구로서 워크케이스 기반의 초대형 워크플로우 시스템의 개요에 대해 설명하고, 3.1장에서는 초대형 워크플로우 시스템에서의 워크케이스 생성 기법들을 Static 방식과 Dynamic 방식 두 가지로 나누어 이에 대한 개념들을 설명한다. 3.2장에서는 이 두 가지의 기법들을 측정하기 위한 테스트 구현 환경에 대해 기술하고, 3.3장에서는 실질적인 테스트를 통해 각각의 워크케이스 생성 기법에 따른 반응 시간 측정 결과를 가지고 비교 분석을 한다. 마지막으로 4장에서는 결론 및 향후 연구 과제에 대해 기술한다.

2. 관련 연구

2.1. 워크케이스 기반의 초대형 워크플로우 시스템

워크케이스 기반의 초대형 워크플로우 시스템 구조는 모든 작업에 대해서 프로세스 인스턴스인 워크케이스가 주체로서 처리를 하게 되며, 요청한 작업의 수와 같은 수의 워크케이스가 증가하는 것이 특징이라 할 수 있다. 다음의 그림은 워크케이스 기반 초대형 워크플로우 시스템의 구조도이다.

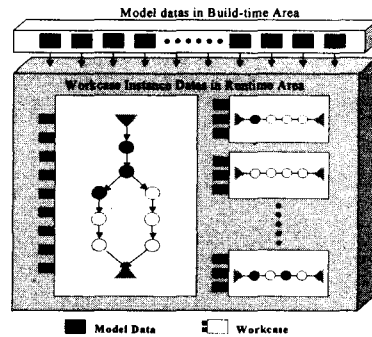


그림 1. 워크케이스 기반 워크플로우 시스템 구조도

워크케이스는 액티비티를 워크케이스 내부적으로 포함하는 데이터로서 보고 있으며 작업을 처리하기 위한 인터페이스들을 외부에 제공한다. 요청된 작업의 처리는 워크케이스가 제공하는 인터페이스를 참조하여 프로세스 관련 데이터, 액티비티 데이터, 관련 데이터들을 가지고 이루어진다. 워크케이스 기반의 초대형 워크플로우 시스템에서는 작업 처리 주체가 되는 워크케이스들이 작업 요청 수와 같은 수로 시스템에 존재하게 되는데, 본 논문에서는 이러한 대량의 워크케이스 발생으로 인해 나타나게 되는 병목 현

상 및 시스템 복잡도 증가에 따른 부하를 줄이기 위한 방안으로서 워크케이스의 효과적인 생성 기법에 대해 기술하고자 한다.

3. 본 론

3.1. 워크케이스 생성 기법

3.1.1. Static 워크케이스 생성 기법

Static 워크케이스 생성 기법은 요청된 작업을 처리하는데 필요한 워크케이스의 모든 액티비티 데이터들이 워크케이스 생성 시부터 모두 인스턴스화 되는 방법이다.

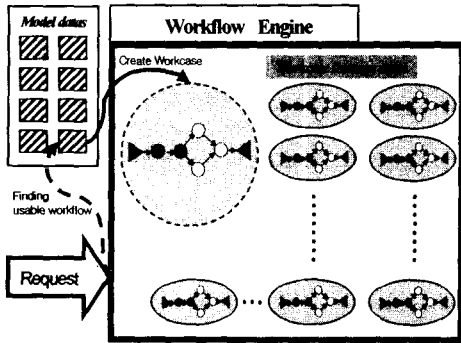


그림 2. Static기법의 워크케이스 생성 개념도

위의 그림 2에서 보는 바와 같이, Static 기법을 사용하게 될 경우 워크케이스는 포함되어진 모든 액티비티들을 오브젝트화 함에 따라 '워크케이스 + 워크케이스가 가지는 모든 액티비티 오브젝트 데이터들' 만큼의 크기를 가지는 워크케이스가 생성 초기부터 존재하게 된다. 따라서, 요청된 작업들의 처리 초기 시점부터 시스템의 리소스 사용에 큰 부담을 안겨 주게 된다. 이는 곧 워크플로우 시스템의 병목 현상을 빠르게 발생시키는 결과를 가져오게 되므로 초대형 워크플로우 시스템에서의 리소스 관리적인 측면에서 볼 때 바람직한 방법이라고 보기 어렵다.

3.1.2. Dynamic 워크케이스 생성 기법

Dynamic 워크케이스 생성 기법은 워크케이스의 시작 액티비티를 기점으로 액티비티를 컨트롤 플로우의 순서에 따라서 시퀀스하게 생성해 나가며 처리하는 방법이다. 다음의 그림 3은 Dynamic 기법을 이용한 워크케이스 발생에 대해 보여주고 있다.

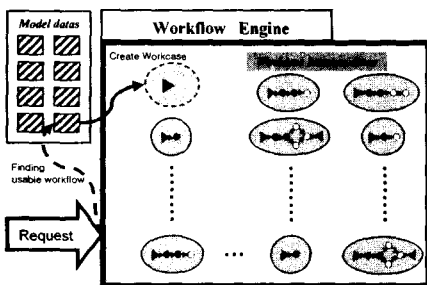


그림 3. Dynamic기법의 워크케이스 생성 개념도

Dynamic 기법은 Static 기법과는 달리, 플로우에 따른 시퀀스한 액티비티의 생성을 통해 처리해 나가는 방식이기 때문에 작업 처리 초기 시점부터 워크케이스들의 생성에 따른 시스템 리소스 사용량이 Static 방식에 비해서 크게 증가하지 않게 된다. 따라서, Dynamic 워크케이스 생성 기법은 Static 기법과 비교하여 워크플로우 시스템의 과부하를 훨씬 지연시킬 수 있다는 이점이 가장으로서 효율적인 워크케이스 생성 및 리소스 관리가 가능하다고 예측할 수 있다.

3.2. 구현

본 논문에서는 앞서 설명한 두 가지의 워크케이스 생성 기법들을 기반으로 반응 시간 결과를 측정할 수 있는 테스트 환경을 구현하고 있다. 테스트는 e-chautauqua 초대형 워크플로우 엔진에 접속하여 해당 클라이언트의 세션을 얻어 내고, 세션을 통해 사용 가능한 워크플로우를 획득하고, 작업 요청을 하게 되면 그에 따른 워크케이스가 생성되어 작업을 시작할 수 있게 된다. 이 때, 워크케이스 생성 시 앞서 설명했던 Static 워크케이스 생성 기법과 Dynamic 워크케이스 생성 기법의 두 경우로 나누어 테스트하게 된다. 구현 환경에 대해서는 e-chautauqua 엔진은 EJB 기술을 사용하여 구현하였으며, EJB를 지원하는 애플리케이션 서버 중 BEA사의 Weblogic 8.1버전을 사용하여 엔진 컴포넌트들을 디플로이 시킨다. 또한 테스트 클라이언트의 제작은 자바 개발 IDE인 Jbuilder X를 사용하였다. 다음의 그림은 e-chautauqua 엔진의 실행을 위한 Weblogic 구동 화면을 보여주고 있다.

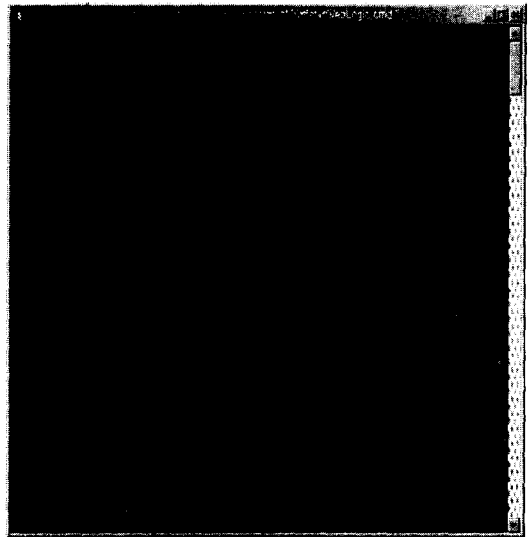


그림 4. e-chautauqua 엔진 실행 화면

아래의 그림 5는 테스트 클라이언트 구현 화면으로서 Jbuilder X에서 제작된 GUI 화면을 나타내고 있으며, 워크케이스 생성 기법들의 테스트를 위한 구현 개발 환경에 대한 정리 내용은 다음의 표 1과 같다.

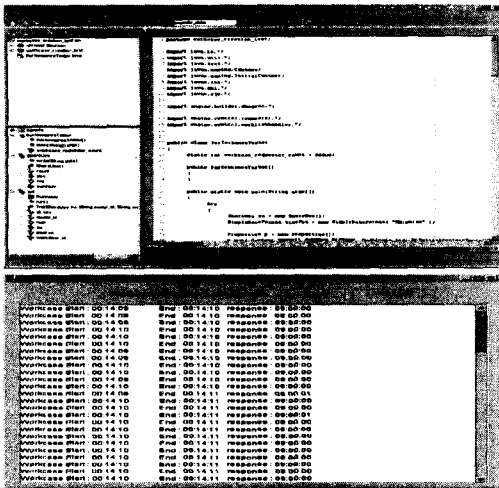


그림 5. 테스트 클라이언트 구현 및 실행 화면

표 1. e-chautauqua&테스트 클라이언트구현 개발환경

하드웨어 환경	
컴퓨터 기종	COMPAQ Proliant ML 530
CPU	Intel Xeon Pentium III 1.0Ghz Dual (x 2)
RAM	1024Mb (133MHz ECC SDRAM)
소프트웨어 개발 환경	
O / S	Window 2000 Server
EJB 애플리케이션 서버	Weblogic 8.1
데이터베이스	Oracle 8.1.7

3.3. 워크케이스 생성 반응시간 분석

다음의 그림은 Static과 Dynamic 생성 관리 기법을 바탕으로 작업 요청 단위 수당 워크케이스 생성에 걸리는 반응 시간에 대해 나타내고 있다.

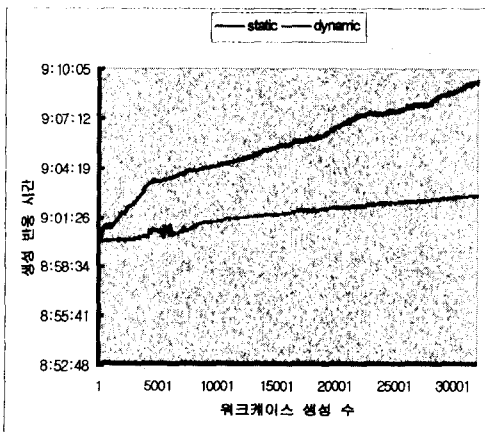


그림 6. 생성 반응 시간 측정 그래프

위의 그래프에서 검정색 선은 Static기법을 이용한 워크케이스 생성 반응에 대한 측정 결과를 보여주고 있으며, 반대로 적색 선은 Dynamic기법에 대한 워크케이스 생성 반응 측정 결과를 나타내고 있다. Static 기법을 이용한 워크케이스 생성 반응 시간 결과는 작업 요청 수 만큼의 워크케이스의 생성 수가 늘어감에 따라 반응 시간에 대한 그래프의 증가 폭이 커짐을 보이는 반면, Dynamic 기법을 이용한 생성 결과에서는 작업 요청의 수가 증가해도 워크케이스 생성 및 처리에 그다지 큰 시간을 소요하고 있지 있음을 알 수 있다. 따라서, 초대형 워크플로우 시스템에서의 워크케이스 생성은 Dynamic 기법을 통해 구성되는 것이 Static 기법에 비해 더 효율적임을 위의 결과를 통해 알 수 있다.

4. 결론 및 향후 연구 과제

지금까지 본 논문에서는 초대형 워크플로우 시스템에 입각하여 작업 요청에 의해 발생하는 워크케이스 생성 시에 Static 기법과 Dynamic 기법 두 가지의 워크케이스 생성 방법 중 Dynamic 워크케이스 생성 관리 기법이 Static 기법에 비해 적합하다는 것을 기술하였다. 그러나 워크케이스라는 것은 결국 작업을 처리하는 데 필요한 요소가 되는 리소스이기 때문에 어떠한 기법을 이용해 생성을 하든 대량의 워크케이스가 대량적으로 발생해 나갈 경우 그에 대한 인스턴스 풀링에 관한 지속적인 고려는 불가피하기 때문에, 향후에는 분산 클러스터링 기법이 추가된 워크케이스 생성 및 리소스 관리에 대한 연구가 추후 진행되어져 나갈 예정이다.

참 고 문 헌

1. 심성수, 김광훈, "대규모 워크플로우 시스템을 위한 EJB 기반 워크리스트 핸들러의 설계 및 구현", 한국인터넷정보학회 추계학술발표논문집, 2001. 11
2. Kim, K, "Architecture for very large scale workflow management systems", PhD Thesis, Computer Science Department, University of Colorado at Boulder, May 1998