

가용성 향상을 위한 분산 워크플로우 모니터링 서비스

홍형석⁰ 배성용 김광훈 백수기
경기대학교 전산학과
(hshong, sybae, kwang, skpaik)@kuic.kyonggi.ac.kr

A Distributed Workflow Monitoring Services with High Availability

Hyeong-Seok Hong⁰ Seong-Yong Bae Kwang-Hoon Kim Su-Ki Paik
Dept. of Computer Science, Kyonggi University

요 약

최근 들어 비즈니스 프로세스의 자동화에 대한 관심이 높아지면서 워크플로우에 대한 연구와 워크플로우 시스템에 대한 개발이 활발히 진행 중에 있다. 그러나 현 워크플로우 시스템을 비즈니스 업무에 실질적으로 적용하기에는 아직 부족한 점이 많다. 비즈니스 업무 자체는 유동적이고, 업무의 양이 많기 때문이다. 따라서, 워크플로우 시스템을 분산시켜 과부하를 줄이고, 변화하는 포인트를 체크하여 그에 맞게 적용해야 한다. 이러한 분산 워크플로우 시스템에서는 모니터링 서비스의 역할이 그만큼 중요하다. 기존의 기능인 워크플로우 흐름을 제어뿐 아니라 분산된 시스템을 관리하는 기능이 추가 되어야 한다. 워크플로우 시스템의 한 요소인 모니터링 서비스는 시스템이 분산되면서, 그 역할도 분산되어야 한다. 모니터링의 분산은 분산된 객체들에 대한 빠른 정보의 수집과 신뢰성있는 서비스의 제공이 그 목적이다. 본 논문은 이에 분산 워크플로우 시스템에서의 모니터링 서비스를 구현하였다.

1. 서 론

워크플로우의 개념이 대두된 후, 워크플로우에 대한 연구가 활발히 진행되었다. 실질적인 비즈니스 업무의 흐름을 원활히 할 수 있는 방법에 대한 연구가 그 대부분을 이루었다. 그것은 워크플로우의 정의 즉 BPR(Business Process Reengineering)을 통해 비즈니스 업무를 자동화하는데 효율적으로 적용하기 위해 어떻게 정의 할 것인가에 대한 연구이다. 그러나 최근 들어, 실질적인 워크플로우를 적용하는 시스템의 개발이 활발히 진행중이다. WfMC(Workflow Management Coalition) 와 OMG(Object Management Group)에서는 WfMS(Workflow Management System)에 관한 표준을 정하고, 그에 맞는 시스템을 개발하기를 권고하고 있다. 이러한 표준에도 시대의 흐름에 발맞추기 위해 표준이 객체기반으로 변화하고 있다. 객체 기반의 워크플로우 시스템을 개발하기 위해서 고려해야 할 것은 그 시스템 구조이다. 시스템 구조가 중앙 집중형태로 설계 된다면 워크플로우 시스템과 같이 많은 인스턴스를 발생 시키는 시스템은 과부하가 일어나는 것은 당연하다. 따라서 워크플로우 시스템에서는 수 많은 인스턴스를 처리하기 위해 분산 환경을 갖추어야 한다.

이렇게 워크플로우 시스템, 정확히 말하면 워크플로우 엔진이 분산되어 있다면 흩어져 있는 요소에 대

해 모니터링은 워크플로우 시스템의 상태에 태를 관장할 수 있어야 한다. 이 부분이 모니터링이 분산환경에 있어서 중요한 역할을 차지하는 이유이다. 기존의 모니터링 서비스에서 실행 중인 워크플로우 인스턴스의 흐름을 제어하는 기능에 더해 시스템의 상태 뿐 아니라, 그 시스템에서 진행되고 있는 인스턴스들의 상태를 파악하여 서비스를 제공하여야 한다. 또한 그 구조 역시 변화가 있어야 할 것이다. 분산의 기준은 규모가 큰 객체에 대한 대비이기 때문에 모니터링 서비스 역시 분산 구조로 흩어져 있는 정보를 수집하는데 보다 빠른 시간으로 제공해야 한다. 모니터링 서비스의 분산은 모니터링 서비스의 가용성을 향상 시키는 결과를 초래한다. 이에 본 논문은 분산워크플로우에서의 모니터링 서비스의 기능과 구조가 기존의 모니터링 서비스와 어떻게 변화가 이루어져야 하는 지에 대해 기술하고, 그것을 설계 및 구현 하고자 한다. 2 장에서는 분산환경에서의 모니터링이 지녀야 할 기능에 대해 기술하고, 그 다음 장에서 그 기능을 분산환경에 맞게 정의 하였다. 4 장에서는 그 기능을 구체화 하였고, 마지막으로 결론을 맺었다.

2. 모니터링 서비스의 기능 정의

일반적으로 워크플로우에서 모니터링이라 함은 현재 실행 중인 워크플로우의 이벤트를 추적하고, 보고하는 능력을 말한다. 예를 들어 한 프로세스의 생성자가 실행 중인 프로세스 인스턴스에 대한 수행을 모니터 하는데 사용된

다. 따라서 모니터링은 워크플로우의 상태에 대한 정보를 탐색하는 것에 대한 검색, 수행 중 정보에 접근하고, 변화를 표현할 수 있는 기능을 제공하여야 한다. 이러한 방식에는 PUSH, PULL 방식을 수용할 수 있다. 즉 객체에서 모니터링 요소에 전달하는 방식과 모니터링 요소에서 객체를 접근하여 정보를 얻는 방식이다.

2.1 기존 모니터링 서비스의 기능

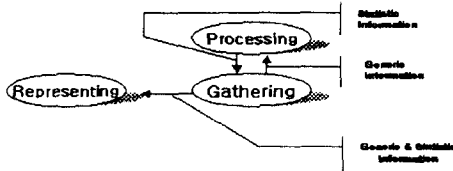


그림 1. 모니터링의 기능

모니터링 서비스는 그림 1과 같이 정보의 수집, 가공, 표현 등 크게 3가지 기능으로 나눌 수 있다.

- 워크플로우 엔진에서 동작하는 객체, 빌드 타임에서 정의된 워크플로우가 저장된 데이터베이스의 정보등을 수집하여 서비스하는 기능
- 워크플로우 엔진에서 종료된 인스턴스를 가공하여 통계정보를 추출하기 위한 기능
- 서비스된 정보를 보다 이해하기 쉬운 정보로 제공하기 위한 그래픽적 표현 기능

2.2 분산 워크플로우 시스템의 모니터링 기능

2.1절에서 제공되는 기능 이외에 분산 워크플로우 시스템에서는 부가되는 기능이 추가되어야 할 것이다. 시스템이 분산되어 있으므로 시스템을 관리하는 운용자는 분산되어진 정보, 즉 각 시스템의 위치, 시스템 내의 객체에 대한 정보, 또한 시스템간의 Load Balancing에 대한 정보를 수집해야 한다. 따라서, 기존 모니터링의 기능이 큰 범주 3가지는 고정적이나 그 내용에서 변화가 있어야 한다.

- 정보 수집 기능은 워크플로우 객체에 대한 정보와 시스템의 분산 정보, 시스템들 간의 상호 작용에 대한 정보를 수집하는 기능이 추가되어야 한다.
- 가공 기능은 운용 관리자로 하여금 분산 시스템의 원활한 흐름을 제어하기 위한 시스템의 수행도를 분석하는 기능이 추가되어야 한다.
- 표현 기능은 시스템의 Load Balancing에 대한 정보를 실시간으로 보여 줄 수 있는 그래프가 필요하다.

3. 모니터링 서비스의 분산 구조

일반적으로 모니터링의 구조를 보면, 어떠한 객체의 상태를 수집해 서비스를 하는 형태로 되어있다. 그러나, 분산 워크플로우 모니터링 서비스는 현재 실행되고 있는 엔진이 분산됨에 따라, 모니터링의 대상이 규모가 확대됨에 따라 모니터링 서비스의 구조 역시 변경이 되어야 한다. 이에 모니터링 요청에 대한 빠른 응답시간과 분산 정보의 관리, 예기치 못한 상황

으로 모니터링 서비스를 제공하지 못했을 상황에 대비해 분산의 구조를 가져야 한다. 워크플로우 시스템의 분산을 고려해 보면, 표 1과 같은 분산 형태가 있을 수 있다.

[표 1]

엔진:모니터	관리의 용이성	가용성	과부하	특징
1:1	o	o	x	자체 분산
N:1	o	x	o	분산 안 함
1:N	x	o	x	기능 분산
N:M	x	o	o	기능분산확장

1:1의 분산은 가장 단순하지만 모니터링의 특성상 가장 알맞은 분산 방법이다. 모니터링의 특징은 한 곳으로 통합하는 성질을 가지고 있기 때문이다. 잦은 엔진과의 통신을 고려할 때 관리의 효율성이 좋고 통신의 장애로 모니터링 서비스(모니터라 칭함)에 접근이 불가할 때 다른 모니터에 접근할 수 있으므로 가용성 면에서도 좋다. 또한 모니터가 유지하는 정보가 연결할 수 있는 레퍼런스만을 유지하기 때문에 과부하가 없다. 그러나, N:1의 경우 통신장애로 인한 가용성과 모니터 자체의 과부하를 피하지 못하고, 1:N의 경우 효율성이 떨어진다. N:M의 경우는 분산 구조 개념에 가장 알맞은 분산형태이지만, 모니터의 특성상 불필요한 관리가 요구된다. 이에 분산 워크플로우 모니터링 서비스는 엔진의 분산과 1:1로 분산하는 것이 가장 알맞다. 그림2는 1:1로 분산된 환경에서 클라이언트의 요청에 따른 정보의 수집을 나타낸다.

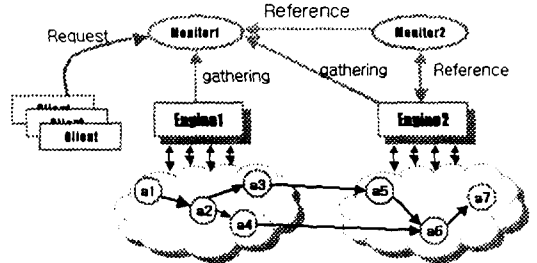


그림 2. 분산 모니터의 정보 수집

위 그림은 a1 ...a7의 액티비티를 포함하고 있는 프로세스 인스턴스가 두개의 엔진에서 동작하고 있다고 가정할 때, 한 워크플로우 클라이언트가 자신의 프로세스에 대해 모니터링을 요구하면, 같은 위치에 있는 엔진의 객체에 대한 정보와 다른 위치에 있는 엔진의 레퍼런스를 요구한다. 이 레퍼런스를 통해 객체에 대한 정보를 요청하여 수집한다. 워크플로우 엔진이 증가 하면 그만큼의 객체를 생성하는 양이 많아 진다는 의미이다. 이에 그림 2와 같은 방식으로 정보를 수집할 경우 한 모니터에 집중되는 것을 막고, 많은 객체의 수집에 효율성을 가져온다. 위의 구조는 이쁜 아니라 가용성면에서도 좋은 영향을 줄 수 있다. 그림 3은 가용성의 향상을 보여준다. 워크플로우 모니터링 서비스는 클라이언트가 자신이 원하는 정보를 얻기 위하여 모니터에 요청을 하게 된다. 그러나 클라이언트가 속해 있는 모니터가 자체의 결함 외에 통신의 단절이나 실패로 인해 서비스를 받지 못할 시 다른 모니터를 통해 정보를 요청할 수 있게 해준다. 그림 3에서는 클라이언트 2가 모니터2에 서비스를 요청했으나 연결이 단절된 경우를 가정한 것이다. 이때 연결 메커니즘을 통해 우선 순위가 있는 모

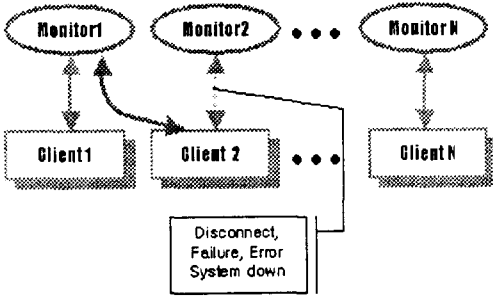


그림 3. 모니터의 가용성

터1에 서비스 요청을 하는 것이다. 규모가 큰 워크플로우 객체에 대해서도 정보를 수집하는데 효율성을 가져온다. 분산 모니터링 서비스는 클라이언트의 요청을 받는 모니터의 연결이 불안정할 경우에 다른 모니터를 연결하는 신뢰성의 이점을 갖고 있다. 그림 3은 한 클라이언트가 자신의 모니터에 연결을 하지 못할 경우의 가용성을 보여준다.

4. 모니터링 서비스의 설계 및 구현

앞 장에서 설명된 분산 모니터의 구조를 바탕으로 모니터를 설계하면 다른 모니터를 접근하기 위한 모니터 레퍼런스를 유지해야 하며, 또한 엔진의 레퍼런스를 유지 하여야 한다.

4.1 모니터 설계

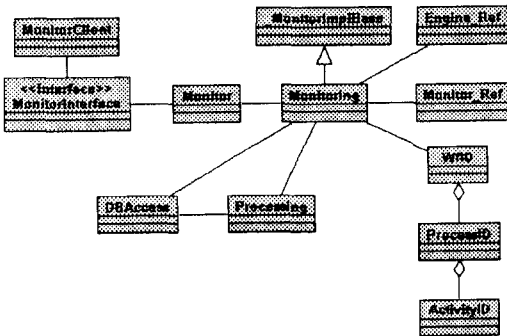


그림 4. 모니터링 서비스를 위한 클래스 다이어그램

모니터링 서비스는 클라이언트와 통신을 할 수 있는 인터페이스가 있으며, 연결을 담당하는 Monitor와 모니터링 서비스를 관장하는 Monitoring이 있다. 또한 엔진들의 객체정보를 갖는 Engine_Ref, 다른 모니터의 객체정보를 갖는 Monitor_Ref, DataBase의 정보를 추출하기 위한 ID를 포함하고 있는 WfID, ProcessID, ActivityID가 있으며, Database를 접근하기 위한 DBAccess, 데이터를 처리하여 통계정보를 산출하는 Processing 클래스 들이 포함된다.

4.2 구현

모니터링 서비스는 별개의 사용자 인터페이스를 지니고 있다. 이는 타 워크플로우 제품과의 구별을

주기 위험이고, 모니터링의 역할이 증대함에 따라 요구되는 사항이다.

본 논문에서 언급되는 분산환경에서의 모니터링 정보 즉, 분산된 시스템에 대한 수행도를 그래프로 보여준다. 그림 5은 4개의 분산된 워크플로우 시스템에 대한 수행도를 나타낸다.

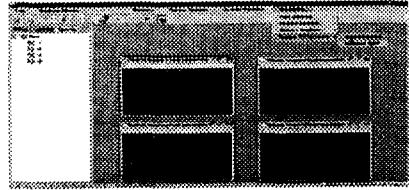


그림 5. 시스템에 대한 수행도

5. 결론 및 향후 발전 방향

분산 워크플로우에서의 모니터링 서비스는 모니터링 기능의 분산이 아니라 모니터 자체의 분산이 이루어 져야 한다. 이는 클라이언트의 빠른 요구 처리와 잦은 엔진의 요구를 막아주는 역할을 한다. 분산된 모니터의 정보는 워크플로우의 흐름에 대한 상태정보와 종료된 인스턴스의 통계 정보 뿐 아니라 분산된 시스템에 대한 정보를 서비스해야 하며, 시스템에 대한 수행도 역시 제공해야 한다. 현재 분산환경의 모니터링 서비스는 개발단계이며, 시스템의 과부하를 줄이면서 모니터의 분산이 이루어 지도록 고려해야 한다.

이러한 서비스를 바탕으로 앞으로 제공되어야 할 요소는 시스템이 분산되면 문제가 되는 네트워크 오류 등의 예기치 않은 오류가 발생을 한다. 이러한 오류에 대한 모니터링 기능 및 분석이다. 또한 환경이 변화함에 따라 변화되는 워크플로우를 관리하는 워크플로우 시스템에서의 모니터링 역시 고려해야 할 사항이다.

[참고문헌]

[1] Workflow Management Coalition Specification Document, Workflow Management Coalition Audit Data Specification, Document Number WfMC-TC-1015, 22 Sep 1998.
 [2] OMG BODTF RFP #2 Submission Workflow Management Facility Revised Submission, OMG Document Number bom/98-06-07
 [3] J.A.Miller, A.P.Sheth, K.J.Kochut and X.Wang CORBA-Based Run-Time Architectures for Workflow Management Systems Large Scale Distributed Information Systems Lab Department of Computer Science The University of Georgia.
 [4] Diimitrios Georgakopoulos and Mark Hornick "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure", Distributed and Parallel Databases, 3, 119-153(1995), GTE Laboratories Incorporated. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.