

결함 감내 지원을 위한 적응형 자율 미들웨어 서비스

송은혜 김은경 이재은 김지영 김윤희

숙명여자대학교 컴퓨터과학과

{grace, kimek, jaeun0215, wldud5, yulan}@sookmyung.ac.kr

Adaptive and Autonomic Service Middleware for Fault Tolerance

Eunhye Song, Eun-kyung Kim, Jaeun Lee, Jeuyoung Kim,
Yoonhee Kim

*Dept of Computer Science, Sookmyung Sookmyung Women's
University

요 약

유비쿼터스 환경은 서비스 이동과 자원 상태 변화 등 실행 환경 변화가 빈번히 발생한다. 이러한 가변하는 실행 환경에 적응하고 가용자원의 부족등으로 발생하는 결함까지 복구할 수 있는 적응형 자율 미들웨어 서비스를 통해 서비스의 계속성 및 고가용성을 보장해야 한다. 이 논문에서는 유비쿼터스 환경에 따라 자율적으로 적응하는 서비스 미들웨어에서 결함 감내 서비스를 지원하는 방법을 제시하였고 프로토타입 Wapee(Web-Service based Application Execution Environment) v0.1을 통해 실제 환경에서 적용가능성을 확인하였다.

1. 서론

다양한 서비스들이 연동되는 유비쿼터스[1] 컴퓨팅 환경에서 요구되는 서비스들의 종류가 다양화되고 그 기능이 복잡해짐에 따라, 하나의 서비스로 사용자의 다양한 요구사항을 만족시켜주기 힘들게 되었다. 또한 같은 종류의 서비스라도 사용자가 필요로 하는 컨텍스트(Context)와 다양한 환경적 요인들의 변화에 따라 그 기능이 달라져야 한다. 따라서 최근에는 사용자의 컨텍스트에 따라 적합한 서비스를 동적으로 선택하여 이용하기 위한 방법들이 활발히 연구되고 있으며, 여러 단위 서비스들을 결합하여 보다 복잡하고 다양한 유비쿼터스 서비스(이하 U-서비스)를 제공하기 위한 노력도 활발히 진행되고 있다.

이렇듯 컨텍스트 기반의 다중 서비스가 결합되는 유비쿼터스 컴퓨팅 환경에서는 끊임없는 서비스를 보장하기 위하여 서비스 이동성 지원과 동적인 자원의 할당, 서비스 결함 감내 등을 통한 고가용성을 지원하는 적응형 자율 미들웨어가 필수적으로 필요하다. 그 중 유비쿼터스 미들웨어 기술인 웹 서비스[2]와 그리드[3] 미들웨어는 공개 표준 기술의 사용

이라는 관점에서는 유틸리티 컴퓨팅보다 한 발 앞서 있으나 서비스의 계속성과 유연한 이동성 및 결함에 대한 복구지원, 상호 보안, 지능적인 자원 분배, 응용을 위한 QoS(Quality of Services)지원 등의 서비스 고가용성은 효과적으로 지원하지 못하고 있다.

본 논문에서는 컨텍스트 및 사용자 요구사항의 변화, 그리고 예상하지 못했던 서비스 오류 발생 시 적절한 복구 지원과 대체 서비스를 제공을 통한 고가용성 지원을 고려한다. 이는 작업 수행 중이나 오류가 탐지되면 동적으로 작업 환경을 재구성하여 보다 유연한 런타임 서비스 환경을 제공하고 안정적으로 서비스의 계속성과 자료의 일관성을 보장하였다.

또한 워크플로우형태의 작업 실행 환경을 지원하고, 그리드 환경에서 작업을 실행하기 위해 필요한 RSL(Resource Specification Language)의 자동 생성과 오류 발생 시에 동적으로 재구성하기 위한 RSL 리플리케이션 서비스를 제시한다. 마지막으로, 필요한 데이터 및 파일을 FTP와 리플리케이션 관리를 이용하여 작업 실행에 필요한 곳에 위치시키고 그리드 환경 상에 존재하는 어플리케이션들의 모니터링 서비스를 제공한다.

이 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 관련 연구를 소개하고, 3장에서는 본 논문에서 제안하는 시스템의 전반적인 구조와 오류 감내 기법을 소개하며, 4장은 프로토타입 구현과 성능을 평가하며, 5장에서 결론과 향후 연구를 밝힌다.

2. 관련 연구

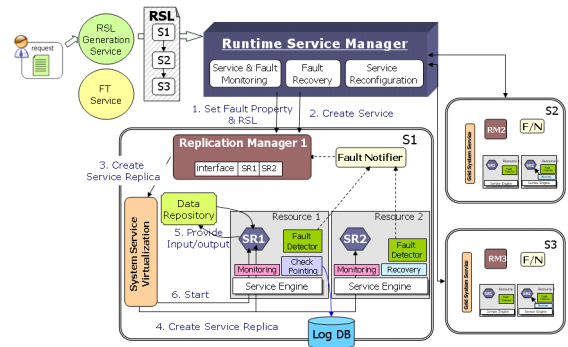
기존 웹 서비스 오류 표준 기술로는 SOAP 메시지 처리 중 발생하는 오류에 대한 자세한 내용을 기술할 수 있도록 여러 요소를 지원하는 SOAP(Simple Object Access Protocol)에서의 Fault[4]부분, 문제 확인과 오류 관리를 지원하는 웹 서비스 오류 메시지를 명세화 한 WSRF(Web Service Resource Framework)의 WS-BaseFault[5]와 분산 컴퓨팅 기술로 CORBA(Common Object Request Broker Architecture)표준에 포함된 오류 감내 기법으로 감지, 통보, 분석 메커니즘을 제공하는 FT(Fault Tolerance)-CORBA[6], ping 메커니즘과 referral component을 통해 네트워크 등 접속 오류를 감지하고 컴포넌트의 상태를 파악하여 자율적으로 새로운 인스턴스를 사용자에게 제공함으로써 오류를 극복하는 DCOM(Distributed Component Object Model)의 Fault Tolerance[7] 등이 있다.

그러나 WS-BaseFault와 SOAP Fault는 웹 서비스 오류 메시지에 대한 정의와 오류 메커니즘을 통해 오류를 처리하나 시스템 오류나 어플리케이션 상의 오류 감내는 존재하지 않는다. FT-CORBA와 DCOM과 같은 오류 감내 기술을 적용한 관련연구로는 GEMS(Grid Enactor and Management Service)[8], Armor(Adaptive Reconfigurable Mobile Objects of Reliability)[9], MEAD (Middleware for Embedded Adaptive Dependability)[10]등이 있다. GEMS는 지속적인 메시지 폴링으로 결함을 감지하나 본 연구는 Armor, MEAD처럼 발생된 결함에 대한 상황을 실시간으로 분석하여 자율적으로 처리가 가능하도록 하였다. 또한 고가용성을 위해 Armor, MEAD처럼 실패된 작업을 처음부터 다시 시작하지 않고 오류가 발생된 부분부터 재시작 하도록 하여 서비스의 연속성을 높였고 오류에 대처할 동일 서비스가 없는 경우, 이를 대체할 수 있는 유사 서비스를 결정하여 제공하였다.

3. 고가용성 지원 시스템의 설계

본 논문의 전체 시스템은 유비쿼터스 서비스를

지원하기 위한 자율 미들웨어로 지리적으로 분산된 컴퓨터 시스템, 대용량 저장 장치 및 데이터베이스, 첨단 실험 장비 등의 자원들을 고속 네트워크를 통해 공유하고 관리될 수 있는 환경을 제공한다. 또한 폴트 리커버리 메커니즘을 적용한 결함 감내 서비스를 통해 고가용성을 지원한다. (그림1)은 전체 시스템 설계를 나타낸 것으로 런타임 서비스 관리를 통해 자신이 실행한 작업의 상태를 모니터링 할 수 있고, 실행 중인 작업을 제어할 수 있다. 또한 작업의 오류에 대해 자동 복구함으로써 작업의 효율성을 증대한다.



(그림 1) 서비스 자율 미들웨어 구성

3.1 작업 실행 관리 서비스

사용자가 작업을 요청하면 런타임 서비스 관리는 사용자의 요구사항에 적합한 유효 자원을 할당하고 작업을 실행한다. 이때 그리드 환경에서 VO로 구성되어 있는 다종의 자원에 등록되어 있는 서비스들은 서비스 풀을 통해 생성·합성·관리·모니터링 된다.

그리드 환경 상의 자원들은 사용 권한과 인증을 설정하고 인증을 받은 사용자에게만 자원을 지원한다. 즉, 그리드 환경의 다수의 자원을 이용하기 위해 사용자는 원격 자원에 대한 접근이 필요로 하며 한번의 인증으로 그리드 환경의 자원을 모두 이용할 수 있어야 한다. 런타임 서비스 관리는 그리드 환경에서 작업을 실행하기 위해 제출해야 하는 RSL(Resource Specification Language)파일을 자동 생성하여 사용자 중심의 시스템을 지원하고 한번 인증으로 여러 자원을 모두 사용할 수 있는 인증서를 제공·관리함으로써 편리성을 높였다.

3.2 모니터링 서비스

런타임 서비스 관리는 실행 중인 작업과 VO 상 자원들의 상태를 감시하는 모니터링 서비스를 제공

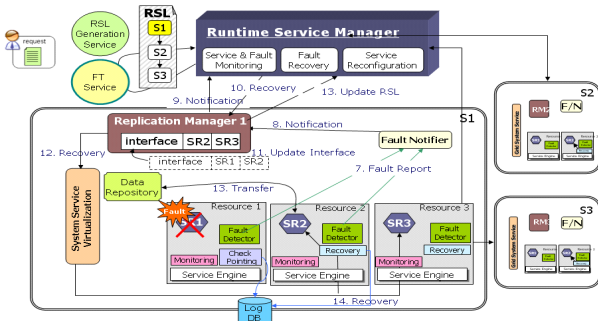
한다. 다양한 응용 어플리케이션을 지원함에 있어 실행 중인 작업에 대한 모니터링은 긴 작업 시간이 요구되는 응용에서는 매우 유용하다. 즉, 작업의 상태를 파악함으로써 오류를 감지하고 작업을 제어함으로써 자원을 보다 능동적으로 사용할 수 있다.

런타임 서비스 관리는 폴링(polling)기법을 이용하여 주기적으로 자원의 상태 변화를 감지하는 자원 모니터링도 제공한다. 이로써 작업의 실행 중 시스템 및 자원, 서비스 등의 결함으로 오류가 발생되면 이를 신속히 감지 할 수 있으며 유용한 자원에 대해 신속하게 맵핑하여 작업 시간의 효율성을 높인다.

3.3. 오류 감내 서비스

본 논문은 결함 감내를 지원하는 적응형 자율 미들웨어 서비스를 지원하기 위해 오류 감지 및 복구 서비스를 제공한다. 런타임 서비스 관리는 모니터링을 이용하여 작업의 오류를 감지하고 리플리케이션 관리자(Replication Manager)를 통해 감지된 오류를 복구함으로써 지속적인 서비스를 제공한다. 이때 오류에 대한 능동적인 서비스를 제공하기 위해 서비스 변화를 수용할 수 있는 서비스 업데이트와 작업에 필요한 데이터 및 파일의 신속하고 정확한 마이크로제이션과 리플리케이션이 동반된다.

(그림2)는 작업 실행 중 오류가 발생했을 경우 자율적으로 변경된 내용을 적용하여 서비스를 재구성하는 흐름을 도식화 한 것이다. 오류 감지기(fault detector)에 의해 오류가 감지되면 오류 통보자(fault notifier)에게 통보된다. 감지된 결함 정보는 리플리케이션 매니저에게 전달되고 결함 상황에 대비하여 미리 생성해 둔 리플리카에 대한 정보와 감지된 결함 정보를 런타임 서비스 매니저에게 통보한다.



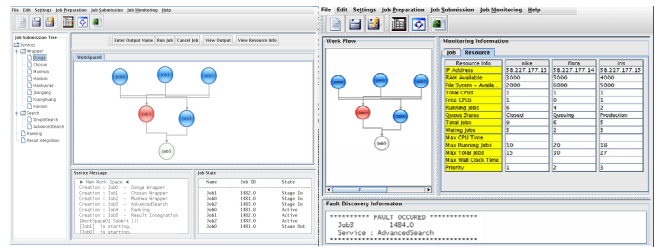
(그림 2) 런타임 서비스 재구성

런타임 서비스 매니저는 감지된 오류의 상태를 파악하고 미리 정의되어 있는 오류 속성을 비교 판단한다. 이를 바탕으로 해당 오류를 분류하고 그에

대한 복구방법을 선택 한 후 RSL을 변경하여 서비스를 재구성하고 자율적으로 업데이트한다. 이때 사용자에게 의해 입력 받았던 요구사항 등은 기존의 정보를 그대로 유지하며 데이터 저장소를 통해 이전에 진행 중이던 작업 정보를 제공받아 오류가 발생된 시점부터 재실행되고 사용자는 자율적으로 재구성된 내용을 보고 받을 수 있다.

4. 구현 및 성능 평가

WSRF(Web Service Resource Framework)기반의 GT4[11] 환경에서 구현된 본 시스템은 (그림3)과 같이 다양한 어플리케이션 타입을 지원하는 워크플로우 기반의 GUI 에디터를 통해 사용자에게 쉽고 편리한 환경을 제공한다.



(그림 3) Wapee v0.1

제공되는 작업 패턴은 순차적인 작업(Sequence), 다중작업(Multi), 분배작업(Split), 병합작업(Merge), 결합작업(Join)들로 잡(Job)마다 ID를 부여하여 작업을 구성하는 잡들의 의존성을 표현하였다. 이와 같은 작업 실행 환경을 제공하기 위하여 그리드 환경에서 작업 관리 및 스케줄링을 담당해주는 메타 스케줄러인 Condor-G[12]와 워크플로우 형태의 작업을 가능하게 해주는 DAGMan[13]을 이용하였다.

모니터링부분은 지역 스케줄러로부터 작업 상태 변화에 대한 정보를 받아 사용자에게 보고해 주는 것은 물론, MDS[14]를 통한 시스템 자원 정보도 제공한다. 그러나 MDS는 로컬 스케줄러에 대한 일부 정보만을 제공하므로 Ganglia[15]를 이용, 다양한 정보를 제공하여 효율적으로 자원 할당을 할당하였다.

작업 상태 모니터링은 작업의 상태를 UnSubmitted, StageIn, Pending, Active, Done, Failed, Suspended, StageOut로 표기한다. 작업 상태가 Failed 일 경우 모든 작업은 중지되고 진행 중이던 작업에 대한 정보는 해당 자원의 데이터 저장소에 저장되며 오류 분석자에 의해 오류가 분석된다. 사용자에게 가시적인 효과를 위해 서로 다른 색으로

작업 상태를 표현하여 작업이 진행됨에 따라 상태의 변화를 파악할 수게 구현하였다.

워크플로우를 제공하는 본 실험 환경에서는 사설을 검색하는 응용, 사용자가 입력한 단어가 포함된 데이터를 검색하는 응용, 입력한 값의 빈도수를 랭킹으로 표현하는 응용을 워크플로우 형태로 표현하고 순서대로 실행하였다. 이는 각 응용의 실행 결과 생성되는 결과파일들은 다음 응용의 입력파일로 사용되며 이전 작업이 완료되지 못한다면 다음 작업이 실행될 수 없음을 의미한다.

Wapee의 모니터링 제공은 기존 응용 실행 시간보다 약 26.3%의 오버헤드가 존재하지만 오류 감내 지원에 있어 DAGMan[13]은 특정 자원을 사용해야 하는 작업에 있어 해당 자원이 존재하지 않는다면 rescue파일을 자동 생성하여 사용자가 재 실행해야 하나 Wapee는 모니터링을 이용하여 해당 자원을 바로 맵핑하며 자원이 존재하지 않아도 이를 어플리케이션 수준에서 자율적으로 추론하여 오류를 자동 복구한다. 또한 오류가 발생한 작업에 대해서만 작업을 재실행해줌으로써 시스템에 부담을 줄일 수 있고 자율적으로 오류 복구를 지원하여 서비스의 계속성을 보장할 수 있다.

5. 결론 및 향후 계획

다양한 U-서비스를 지원을 위해서는 끊임없는 서비스를 보장되고 서비스 이동성 지원과 적응형 동적 자원 지원, 서비스 결합 감내를 지원하는 자율 미들웨어가 필수적으로 필요하다. 이에 따라 본 논문에서는 개방형 표준을 고려하고 고가용성을 지원하는 적응형 자율 미들웨어 구조를 설계하고 프로토타입 구현을 통해 검증했다.

U-서비스 환경에서 고가용성을 지원하는 다양한 방법 중 본 논문에서는 결합 감내 서비스를 통하여 자율성을 지원하고자 프로토타입을 구현하였다. 차후에는 자율성을 지원하기 위한 폭넓은 방안을 제시해, 사용자에게 유연하고 투명한 실행 환경을 제공해야 할 것이다. 또한 다양한 어플리케이션이 지원되는 환경에서의 작업 실행을 위해 다양한 워크플로우 패턴 및 어플리케이션 타입을 지정해 어느 환경에서나 적응 가능한 에디터를 지원해야 한다.

참고문헌

[1] Weiser, M. "The computer for the 21st Century." Scientific American, Vol. 265, No. 3, pp.

94-104, September, 1991.

[2] OASIS <http://www.w3.org/2002/ws/>

[3] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International J. Supercomputer Applications, 2001.

[4] SOAP fault <http://www.w3.org/2000/xp/Group/>

[5] OASIS <http://docs.oasis-open.org/wsrf/>

[6] <http://www.omg.org/docs/formal/01-09-29.pdf>

[7] DCOM Fault <http://msdn.microsoft.com/>

[8] Satish Tadepalli, Calvin Ribbens, Srinid Varadarahan "GEMS: A Job Management System for Fault Tolerant Grid Computing", High Performance Computing Symposium, 2004

[9] Zbigniew Kalbarczyk, Ravishankar K Iyer, Long Wang, "Application Fault Tolerance with Armor Middleware" Internet Computing, March/April 2005 (Vol 9, No 2) pp 28-37

[10] P Narasimhan, C F Reverte, S Ratanotayanon and G S Hartman, "Middleware for Embedded Adaptive Dependability" IEEE Workshop on Large Scale Real-Time and Embedded Systems, Austin, TX, December 2002

[11] Globus Project, <http://www.globus.org/>

[12] James Frey, Todd Tannenbaum, Ian Foster, Miron Livny, and Steven Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids", Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10) San Francisco, California, August 7-9, 2001

[13] Condor DAGMan <http://www.cs.wisc.edu/condor/dagman/>

[14] K Czajkowski, S Fitzgerald, I Foster, C Kesselman "Grid Information Services for Distributed Resource Sharing", Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001

[15] Matthew L Massie, Brent N Chun, David E Culler, "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", Parallel Computing, Vol 30, Issue 7, July 2004