

그룹통신 시스템을 이용한 복제 JavaSpace의 설계

문남두⁰, 이근웅, 구형서, 박양수, 이명준

울산대학교 컴퓨터정보통신공학부

(dooya⁰, daredevi, masker, yspark, mjlee)⁰@mail.uslan.ac.kr

Design of a Replicated JavaSpace using Group Communication System

Nam-Doo Moon⁰, Keun-Woong Lee, Hyeong-Seo Koo, Yang-Soo Park, Myung-Joon Lee

School of Computer Engineering & Information Technology, University of Ulsan

요 약

JavaSpace는 쉽고 단일화된 방식으로 분산 어플리케이션간의 객체공유, 중재 및 동적인 통신서비스를 제공한다. 현재 JavaSpace를 이용한 응용 프로그램들이 계속적으로 등장하고 있다. 기존의 JavaSpace는 중앙 집중적인 클라이언트/서버 방식으로 구현되었기 때문에 성능이 제한적이다.

본 논문에서는 그룹 참여자들간의 상태를 일관성 있게 유지시켜주는 그룹통신 시스템을 이용하여 분할 가능한 분산환경에 적합한 복제된 JavaSpace 시스템을 설계하였다. 복제된 JavaSpace 시스템을 이용하는 응용 프로그램들은 효율성과 확장성을 증대시킬 수 있다.

1. 서 론

최근 인터넷의 급속한 성장과 네트워크 기술의 발전으로 다양한 분야에서 분산 어플리케이션들이 계속적으로 등장하고 있다. 다수의 분산된 어플리케이션들은 객체공유, 중재(coordination) 및 동적인 통신 서비스를 요구한다. 이러한 요구사항을 만족시키기 위해 썬 마이크로시스템즈사는 쉽게 사용할 수 있는 JavaSpace 서비스[1] 명세를 제안하였다. 현재 병렬/분산 컴퓨팅 환경에서 JavaSpace를 이용한 응용 프로그램들이 계속적으로 개발되고 있다. 그러나, 기존의 JavaSpace는 중앙 집중적인 클라이언트/서버 방식으로 구현되었기 때문에 성능이 제한적이다.

본 논문에서는 그룹 참여자들간의 상태를 일관성 있게 유지시켜주는 그룹통신 시스템인 Jgroup[2, 3]을 이용하여 분할 가능한 분산환경에 적합한 복제된 JavaSpace 시스템을 설계하였다. 복제된 JavaSpace 시스템을 이용하는 응용 프로그램들은 효율성과 확장성을 증대시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 제안되어 있는 JavaSpace 모델과 인터페이스를 기술하고, 3장에서는 분할 가능한 분산환경에 적합한 그룹통신 시스템을 이용한 복제 JavaSpace 시스템을 제안한다. 4장에서는 복제 JavaSpace의 내부 구현을 기술하고, 5장에서는 관련연구에 대하여 소개한다. 끝으로 6장에서는 결론과 앞으로의 연구과제를 기술한다.

2. JavaSpace 모델과 인터페이스

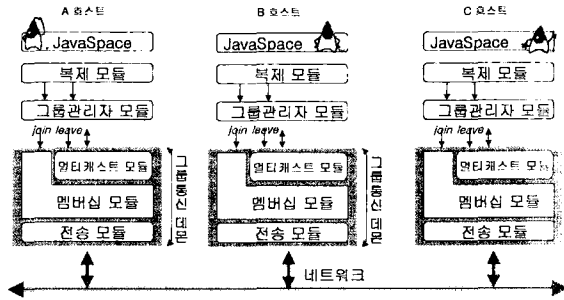
썬 마이크로시스템즈사는 JavaSpace 모델과 인터페이스를 정의하였다.[1] JavaSpace는 Java 기술에 기초한 클라이언트/서버와 같은 네트워크 자원들간의 객체공유, 중재 및 동적 통신기능을 쉽고 단일화된 방식으로 서비스를 제공한다. JavaSpace 기술은 LINDA 모델[4]과 매우 유사하며, 튜플(tuples) 대신에 엔트리(Entry) 인터페이스를 구현한 인스턴스를 관리한다. JavaSpace 서비스는 [표 1]과 같이 기본적으로 7가지의 API를 제공한다.

[표 1] JavaSpace 인터페이스

메소드	설 명
<i>write</i>	주어진 엔트리(entry)를 JavaSpace의 저장소에 저장한다.
<i>read</i>	주어진 템플릿(template)과 매칭되는 엔트리를 읽는다.
<i>readIfExists</i>	주어진 템플릿과 매칭되는 엔트리를 읽는다. 매칭되는 엔트리가 없다면 블록(block)되지 않고 null을 반환한다.
<i>take</i>	주어진 템플릿과 매칭되는 엔트리를 읽고, JavaSpace 저장소에서 제거한다.
<i>takeIfExists</i>	주어진 템플릿과 매칭되는 엔트리를 읽고, JavaSpace 저장소에서 제거한다. 매칭되는 엔트리가 없다면 블록(block)되지 않고 null을 반환한다.
<i>notify</i>	주어진 템플릿과 매칭되는 엔트리가 JavaSpace에 저장되면 지정된 객체에 그 사실을 알려준다.
<i>snapshot</i>	원래 엔트리의 snapshot을 포함하는 새로운 엔트리 객체를 반환한다.

3. 복제된 JavaSpace 시스템 구조

(그림 1)는 복제 JavaSpace 시스템 구조와 Jgroup 그룹통신 시스템과의 관계를 도식적으로 보여준다.



(그림 1) 복제 JavaSpace 시스템 구조

시스템의 주요 컴포넌트는 다음과 같다.

- 복제된 JavaSpace : 썬 마이크로시스템즈사의 JavaSpace와 동일한 인터페이스를 제공하며, 내부적으로 그룹통신 시스템을 사용하여 다수의 호스트에 동일한 상태를 갖는 복제된 JavaSpace를 유지할 수 있다.
- 복제 모듈 : 그룹관리자 모듈의 일부이며, 그룹과의 연결 및 멤버들간의 일관성 유지를 위해 복원작업 등을 내부적으로 수행한다.
- 그룹관리자 모듈 : 그룹관리자는 서버 객체(혹은 프로세스)에 그룹통신 기능을 지원하며, 여러 개의 레이어(layer)로 구성된다. 그룹관리자와 그룹통신 데몬 사이의 상호작용을 지원하는 레이어, 상태 전이 레이어와 그룹메소드 호출 레이어가 있다.
- 그룹통신 데몬 : 그룹 멤버십 관리, 멀티캐스팅, 실패 탐지 등의 기본적인 그룹통신 기능을 제공하며, 멤버들간의 상태를 일관성 있게 유지하기 위해 메시지를 교환한다.

3.1 환경설정

그룹통신 데몬은 환경설정 파일을 읽어 들인 후, 초기화작업을 수행한다. 환경설정 파일은 (그림 2)와 같이 XML 형태로 도메인, 호스트 정보, 멀티캐스팅 주소와 버퍼 크기 등 다양한 정보를 포함한다.

```
<GroupCommunication version="1.0">
  <Domain name="pslab.ulsan.ac.kr">
    <Mcast> 239.1.2.3 </Mcast>
    <BufferSize> 4096 </BufferSize>
    <Host>ndmoon</Host>
    <Host>avenue</Host>
    <Host>soft</Host>
  </Domain></GroupCommunication>
```

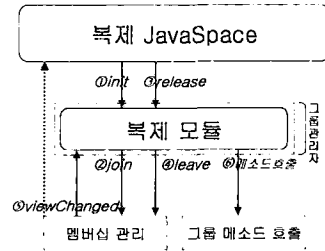
(그림 2) XML 환경설정 파일

호스트 정보는 처음 실행될 때 참조되고, 수행 시간

에는 동적으로 재구성된다.

3.2 복제 모듈

복제 JavaSpace는 (그림 3)과 같이 복제 모듈의 init 메소드를 호출하여 그룹의 멤버로 참여한다. 복제 JavaSpace가 참여하는 그룹명은 명령행 인자로 전달될 수 있으며, 생략되면 디폴트 그룹의 멤버로 참여한다. 멤버십 변경 이벤트를 복제 모듈에서 받으면 이벤트에 따라 적절한 복원작업을 수행한다.

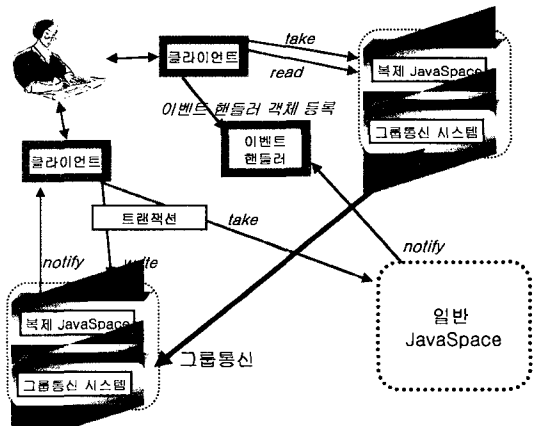


(그림 3) 복제 모듈

3.3 그룹통신

분산된 JavaSpace를 그룹으로 동작시키고, 멤버들간의 동일한 상태를 유지하기 위해 신뢰성 있는 그룹통신기능이 요구된다.

- atomic : 엔트리 삭제는 원자성을 보장받는다.
- all or nothing : 서비스 요청이 모든 JavaSpace 노드에 전달되는 것을 보장받는다.
- Jgroup[2, 3] 시스템 : 분할 가능한 네트워크 환경에서 객체그룹 패러다임과 JavaRMI 분산객체를 통합한 그룹통신 시스템이다. Jgroup 시스템은 분할가능 멤버십 관리, 상태전이와 그룹메소드 호출 등의 서비스로 구성되었다. 그룹메소드 호출은 내부 그룹메소드 호출과 외부 그룹메소드 호출로 구분된다.



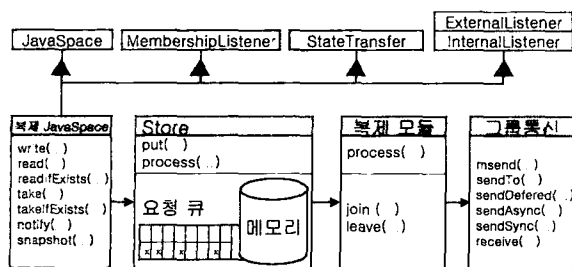
(그림 4) 복제 JavaSpace의 응용

클라이언트는 (그림 4)와 같이 투명하게 복제 JavaSpace를 이용하여 엔트리를 저장하고, 템플릿을 이용하

여 원하는 엔트리를 읽거나 가져오기를 수행할 수 있다.

4. 복제 JavaSpace와 Store 구현

(그림 5)는 그룹통신을 이용한 복제 JavaSpace의 구현과 메소드 호출 관계를 보여준다.



(그림 5) 그룹통신을 이용한 복제 JavaSpace

복제 JavaSpace는 JavaSpace 인터페이스를 구현한다. 클라이언트의 서비스 요청은 내부적으로 참조하고 있는 Store 객체에게 전달된다. 썬 마이크로시스템즈사의 JavaSpace 모델이 메모리 상에서 공간(space)을 관리하는 *transient-outrigger*와 영속성을 지원하는 저장소에 공간을 관리하는 *outrigger*로 구분되어 제공되는 것과 같이 두 가지 이상의 형태로 구현될 수 있다. 우선적으로 메모리 상에서 관리하는 모델을 고려한다면 클라이언트의 요청은 Store 객체의 `process` 메소드에 의해서 처리되는데 `write`의 경우는 메모리 저장소에 엔트리를 추가한다. `readIfExists/takeIfExists`의 요청은 메모리 상에 요구사항이 만족되는 엔트리를 찾는 경우, 그 결과를 반환하지만 만족하는 엔트리를 찾지 못한 경우 `null`을 반환한다. 반면 `read/take` 등의 경우에는 현재 메모리 상에 요구사항에 만족하는 엔트리가 있다면 그 결과를 반환하지만, 만족하는 엔트리를 찾지 못한 경우는 클라이언트의 요청이 해결되지 못했으므로 미해결 요청 큐에 임시 보관한다. 호출 쓰레드는 블록된다. 새로운 엔트리가 메모리에 추가될 때마다 요청 큐를 검색하여 요구사항이 만족되는 요청이 있는 경우 이를 처리하고 요청 큐에서 제거한다.

5. 관련 연구

- 썬 마이크로시스템즈사의 JavaSpace[1] : 메모리 상에서만 공간(space)을 관리하는 *transient-outrigger*와 영속성을 지원하는 저장소에 공간을 관리하는 *outrigger*로 구분하여 제공된다. 양쪽 모두 중앙 집중적인 클라이언트/서버 접근방식으로 구현되었으며, 성능이 제한적이다.
- IBM의 T Spaces[5] : 데이터베이스 기능을 제공하고, 튜플 스페이스에 기초한 네트워크 통신 버퍼 시스템이다. T Spaces 시스템은 인터넷 환경과 같이 이 기종 컴퓨터와 운영체제를 사용하는 응용 프로그램과 디바이스들간의 통신을 지원하고 무한히 확장할 수 있는

협력 클러스터링화 시키는 것을 목적으로 하고 있다.

- HARNESS를 이용한 JavaSpace[6] : HARNESS[7]는 가상 머신의 기능과 가상 머신을 구성하는 컴퓨터 및 네트워크의 동적인 재구성을 지원하는 실험적 메타 컴퓨팅 시스템이다. HARNESS 시스템의 모듈러 프로그래밍 환경에 대한 검증과 HARNESS 시스템에 유동적인 잠재기능을 제공하기 위한 JavaSpace를 구현하였다.

6. 결론

썬 마이크로시스템즈사의 JavaSpace는 중앙 집중적인 클라이언트/서버 방식으로 구현되었기 때문에 성능이 제한적이다.

본 논문에서는 그룹 참여자들간의 상태를 일관성 있게 유지시켜주는 그룹통신 시스템인 Jgroup을 이용하여 분할 가능한 분산환경에 적합한 복제된 JavaSpace 시스템을 설계하였다. 복제 JavaSpace 서비스는 그룹통신 기능을 이용하기 위해 `MembershipListener`, `StateTransfer`, `ExternalListener`와 `InternalListener` 등의 인터페이스를 구현한다. 복제된 JavaSpace 시스템을 이용하는 응용 프로그램들은 효율성과 확장성을 증대시킬 수 있다.

추후 연구과제로는 본 논문에서 기술한 시스템 설계를 바탕으로 복제 JavaSpace 시스템을 구현하는 것이다. 또한 기존에 개발된 다수의 그룹통신 시스템을 쉽게 플러그인 할 수 있도록 인터페이스를 제공하고자 한다.

참고문헌

- [1] Sun Microsystems, Inc. JavaSpaces™ Service Specification, Version 1.1, October 2000.
- [2] A. Montresor, R.Davoli, O. Babaoglu. Enhancing Jini with Group Communication. In Proceedings of the ICDCS Workshop on Applied Reliable Group Communication (WARGC 2001), April 2001,
- [3] A. Montresor. Jgroup Tutorial and Programmer's Manual, September 2000
- [4] N. Carriero and D. Gelernter. Linda in Context. Communications of the ACM, 32(4):444-458, 1989.
- [5] Tobin J. Lehman, Stephen W. McLaughry, and Peter Wyckoff. T Spaces: The Next Wave. In Proceedings of the Hawaii International Conference of System Science, January 1999.
- [6] Mauro Migliardi, Simon Schubiger, and Vaidy Sunderam. A distributed JAVASPACE implementation for HARNESS. Journal of Parallel and Distributed Computing, 60(10):1325-1340, 2000.
- [7] M. Migliardi and V. Sunderam. Heterogeneous Distributed virtual Machines in the Harness Meta computing Framework. In Proc. of the Heterogeneous Computing Workshop of IPPS/SPDP 1999, pages 60-73, S. Juan de Puerto Rico, April 12-16 1999. IEEE Computer Society Press.