방향재설정과 스케쥴링을 통한 지니시스템의 부하 분산 기법

고필훈, 김혜영, 송의성, 윤태명, 황종선
고 려 대 학 교 컴 퓨 터 학 과
(phko,khy)@disys.korea.ac.kr

Load Distribution mechanism using Scheduling and Redirection scheme for Jini

Pil-Hoon Ko , Chong-Sun Hwang
Dept. of Computer Science, Korea University

요 약

지니(Jini)는 썬 마이크로 시스템즈에서 제안하고 있는 확장성 있고 견고함을 가진 분산 시스템을 구축하기 위한 구조이다. 지니(Jini) 서비스들은 하나 또는 그 이상의 자바 언어 인터페이스와 클래스들에 의해 정의된다. 이것은 하드웨어나 소프트웨어의 구현에 상관없이 지니 서비스를 사용하는 기기들 사이에서 동적인 상호작용을 생성시키고 네트워크를 통해 서비스를 전달한다. 일단 기기들이 특정한 하드웨어에 구애 받지 않고 각종 디바이스를 통해서 네트워크에 접속이 이루어지면 시간과 장소에 상관없이 각종 서비스를 받을 수 있게 된다.

본 논문에서는 스케쥴링과 방향 재설정을 통해 지니에서의 부하를 분산 시키는 방법으로 보다 효율적인 서비스를 받을 수 있는 모델을 제시한다..

1.서 론

컴퓨터 네트워킹이 발전하면서 적은 비용으로 높은 성능을 낼 수 있는 분산시스템이 발달하게 되었고 컴퓨터 보급이 확산되면서 사용자들은 네트워크 접속에 대해 보다 쉬운 설정방법을 필요로 하게 되었다. 시스템이 특별한 별도의 드라이버를 필요로하지 않고 사용자는 원하는 장치나 소프트웨어 같은 새로운 서비스를 네트워크를 확장하기 위해 쉽게 추가할 수 있으며 시스템은 다른 서비스를 접근하거나 그것들을 사용하기 위해 쉽게 탐색을 할 수 있어야 한다. 자바와 자바 RMI 기술에 기반을 둔 지니는 이런 것들을 실현할 수 있는 기술이다.지니는 여러 장소에 있는 장치와 소프트웨어 컴포넌트 그룹을 연결할수 있고 (federating) 각각 서로간에 서비스를 사용할 수 있다.

지니에서 중요한 개념은 서비스이다. 서비스는 계산, 자바스페이스 같은 저장소, 다른 사용자들간의 통신채널, 소프트웨어 필터, 하드웨어 장치 등 아주 다양하다. 지니 서비스 그룹은 서로 상호작용하는 것 끼리 모이며 이것은 커뮤니티라 불린다. 커뮤니티의 모든 서비스들은 서로간의 정보를 알고 있고 각자 사용이 가능하다. 보통 서비스가 제공되려면 서비스는 서비스 관리자를 찾아야 하며 서비스관리자와의 통신을 통해 서비스 등록을 하게 된다. 클라이언트는 서비스를 이용하기 위해서 룩업 서비스를 찾아서 자신이 원하는 서비스의 상태를 명세하여 요청을 하여야 한다.

본 논문에서는 클라이언트가 서비스를 재 요청할 시에 스케줄러에 의해 할당을 받고 과부하시 클라이언트 방향 재설정을 통해 시스템의 부하를 효율적으로 줄이는 서비스 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2 절에서는 지니의 개요 및 관련 연구에 대해 다루고 3 절에서는 기본적인 시스템 모델을 나타낸다. 4 절에서는 제안된 스케줄링 기법과 변형된 모델을 나타내고 5절은 시뮬레이션을 통해서 제안된 기법의 효율성을 검증한다. 마지막으로 6 절에서는 결론 및 향후 연구방향에 대해 다룬다.

2. 지니의 개요 및 관련 연구

2.1 지니의 목표

1) 네트워크 플러그 앤 플레이
사용자는 네트워크를 통해 서비스에 접속할 수 있어야하고 그것을 사용할 수 있어야한다.

2) 하드웨어와 소프트웨어를 구별안함
사용자는 자신이 원하는 서비스가 소프트웨어인지 하드웨어인지 알 필요가 없다.

3) 즉각적인 네트워킹
서비스가 네트워크에 접속될 때 사용 가능해야 하며 클라이언트에 의해서도 사용가능 해야 하고 다른 서비스에 의해서도 가능해야 한다.

2.2 지니의 구성요소

1) 서비스
서비스는 지니 아키텍쳐에서 가장 중요한 개념이다. 서비스는 사람이나 프로그램 혹은 다른 서비스에 의해 사용될 수 있는 개체로서 정의된다. 지니 시스템 구성원은 서비스 접근을 공유하기 위해 연합체를 형성한다. 그러므로 지니 시스템은 클라이언트와 서버의 집합으로써 생각되지 않아야 한다. 서비스들은 다른 서비스를 사용할 수 있고 한 서비스의 클라이언트는 다른 클라이언트에 의해 서비스가 될 수 있다.

2) 룩업 서비스
서비스들은 룩업서비스에 의해 발견되고 사용이 결정된다. 룩업서비스는 시스템에 대한 중심적인 메카니즘이며 시스템과 시스템의 사용자간의 접근 지점을 지원한다. 정확한 의미에서 룩업서비스는 서비스에 의해 지원되는 기능을 가리키는 인터페이스를 서비스를 구현하는 객체 집합에 매핑한다. 서비스는 발견과 결합이라불리는 프로토콜에 의해 룩업서비스에 추가된다.

3) 발견과 결합
지니의 발견과 결합은 네트워크에서 커뮤니티를 찾기위해 사용되는 프로세스이고 이들 커뮤니티를 지원하는 사용가능한 룩업 서비스를 찾는다.
발견 프로세스는 관심있는 잠정적 서비스 유저에게 룩업서비스 집합에 대해 알게하고 서비스 프록시를 얻는다.
4) 속성
속성은 서비스 프록시에 첨가된 자바 객체로써 서비스에 의해 지원되는 인터페이스에 의해 나타나지 않는 서비스들을 구분하기 위해 상대적인 특징을 기술한다.
서비스들은 서비스 프록시에 퍼블리시 될 때 이들 속성을 추가시키고 클라이언트들은 특정 속성 형태를 검색함으로써 룩업서비스를 찾을 수 있다.
지니 클래스 라이브러리는 서비스에 대한 정보를 지원하기 위해 많은 표준 속성 형태를 지원한다. 표준 지니 속성은 주소, 주석, 위치, 이름, 서비스정보, 서비스형태와 상태이다.
이들 형태를 잘 맞춤으로써 서비스는 자신에 대한 정보를 잘 정리된 형태로 퍼블리시할 수 있고 클라이언트들은 효과적인 검색을 할 수 있다.

2.3 방향 재설정
본 논문에서의 재방향 설정은 http 의 패킷 재쓰기를 통한 재방향 설정 기법에 근거한다. 서버에 과다한 패킷이 들어와 과부하가 발생할 경우에 패킷의 목적지 주소를 바꿔 줌으로써 서버의 부하를 분산시킨다. 서버 부하의 분산을 위해 재방향 설정을 이용한 접근은 많이 이루어져 왔다. 여러 재방향 기법들을 단계별로 정리하면 첫번째 단계는 재방향 프로세스를 활성화시키는 것과 활성화 결정 프로세스이다. 여기서 재방향 설정을 결정하는 프로세스가 활성화되는데 있어 활성화 촉진 메커니즘을 포함한다. 활성화 촉진 메커니즘은 다시 동기식과 비동기식으로 나뉘고 활성화 결정 프로세스는 중앙집중형과 분산형으로 나뉘게 된다. 두번째 단계는 재방향 설정 기법을 구현하는데 있어 상태 정보를 사용하는 것이다. 이것은 웹서버 부하정보와 도메인 부하 정보가 있을 수 있다. 세번째 단계는 재방향 정책을 실행하는 것이다. 이것은 재방향 설정되는 요청을 받는 재방향 설정 서버 결정도 포함된다. 그리고 재방향 설정되는 개체들은 도메인 전체일수도 있고 도메인 내의 어떤 개인적인 클라이언트일 수도 있다.

3. 지니 시스템 모델
기존의 지니 모델은 다음과 같다.

클라이언트가 룩업서비스에게 서비스요청을 하면 는 발견과 결합 프로토콜에 의해 저장하고 있던 프록시 객체를 업서비스에 추가된다. 클라이언트에게 반환하고 클라이언트는 이를 받은 후 RMI등의 통신프로토콜을 사용해 서비스 제공자와 통신을한다.

클라이언트는 찾는 서비스를 지정하기 위해 다음과 같은 ServiceTemplate형의 클래스를 사용한다.

```
Public class ServiceTemplate {
 Public ServiceID serviceID;
 Public java.lang.Class[] serviceTypes;
 Public Entry[] atributeSetTemplates;

 ServiceTemplate(ServiceID serviceID,
                 java.lang.Class[] serviceTypes,
                      Entry[] attrSetTemplates);
                                }
```

룩업서비스는 이 템플릿을 매칭 시켜 매칭되는 프록시를 클라이언트에게 반환하는데 이 때 ServiceMatches class 가 사용된다.

```
public class ServiceMatches implements Serializable {
  public ServiceMatches(ServiceItem[] items,
                  Int totalMatche          s) {
                                                ...}
  Public ServiceItem[] items;
  Public int totalMatches;
  }
```

클라이언트는 다운받은 프록시를 경유하여 서비스와 RMI 등의 private 통신 프로토콜로 통신한다.

4. 제안된 스케쥴링
스케쥴링의 기본 아이디어는 다음과 같다.
첫 번째로 스케쥴러는 정적 스케쥴링을 하며 각각 시간 T_N 동안 스케쥴링을 수행한다.
두 번째는 네트워크에서 각 클라이언트들은 원하는 룩업서비스에 할당이되고(단, 이경우 룩업서비스의 주소를 알고, TCP 기반의 유니캐스트를 사용하는 경우임) 보통 T_N 동안 계속 할당이 된다고 가정한다. 단, 여기서 리싱문제는 고려하지 않는다고 가정한다. T_N 후에 네트워크상의 각 클라이언트는 정적 스케쥴러에 의해 다시 스케쥴링되고 다른 룩업서비스에 할당이 된다. 여기서 할당이란 클라이언트가 서비스 이용을 끝내고 다시 서비스를 필요로할 시 이루어진다고 가정한다.
세번째는 네트워크상의 클라이언트는 클라이언트가 맨 처음 룩업서비스를 찾기위해 UDP기반의 멀티캐스트를 보내는 경우를 제외하고 일반적으로 자신이 원하는 룩업서비스에 접근을 하려고 노력한다. 만약 원하는 룩업서비스가 과부하 상태이면 클라이언트는 시간 T_r 동안 방향재설정을 통해 다른 룩업서비스에 할당이 된다. 이를 통해 클라이언트에 의해 생기는 부하가 다른 룩업서비스에 분산이 되면서 부하를 분산시킬 수 있다.
스케쥴러는 부하 예측 , 정적 스케쥴링 , 부하분산 및 클라이언트 방향 재설정의 단계를 따른다.

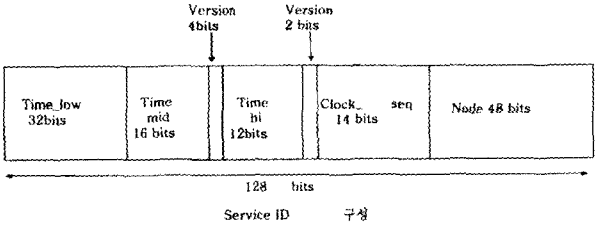
4.1 부하 예측
서비스관리자는 썬 마이크로 시스템의 reggie 를 사용하고 이것의 로그 파일을 통해 부하정보를 얻는다. 부하예측을 하기 위해 접근 기록들이 관리되고 시간에 따라 네트워크상의 접근 수를 계산한다. 시간에 따라 서로 다른 가중치를 기록에 부여한다. 여러 시간의 히스토리 정보를 가질수록 변이가 적어지고 더 정확한 정보를 얻게 된다.

4.2 스케쥴링
스케쥴링 방식은 다음과 같다. 스케쥴러는 부하예측을 한 후에 모든 룩업 서비스를 접근 수에 따라 내림차순으로 정리한다. 그 다음 스케쥴러는 이 순서대로 스케쥴링을 한다. 네트워크상에서 스케쥴을 할 때 현재 부하를 결정하기

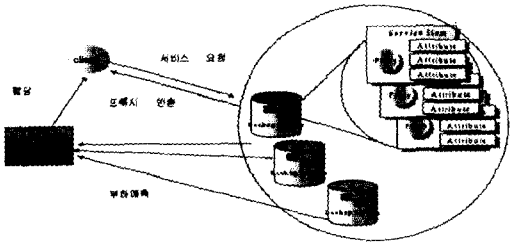
위해 모든 룩업 서비스를 체크한다.

4.3 부하 분산 및 클라이언트 방향 재설정
각 룩업서비스는 그룹을 지을 수 있으며 커뮤니티를 이룰 수 있다. 클라이언트는 룩업서비스가 그룹을 이루고 있는지 단일 서비스인지 알 필요가 없기 때문에 투명성을 가진다.
스케쥴링 하에서 룩업 서비스의 부하는 거의 균등하게 된다.하지만 순간적인 과도한 접근 들으로 일시적으로 과부하 상태가 될 수 있다. 이 문제를 해결하기 위해 본 논문에서는 일시적으로 다른 룩업 서비스에 접근 방향을 재설정하기 위해HTTP 방향 재설정에 근거를 둔다. [3] 룩업 서비스가 미리 정의된 임계값을 넘어설 때 스케쥴러는 클라이언트에게 현재의 과부하 룩업 서비스로부터 방향 재설정을 지시한다. 클라이언트는 적은 부하의 룩업서비스에T ,동안 접근을 하게 되고 그 후에 다시 원래의 룩업 서비스로 돌아오게 된다.

4.4 스케쥴러 컴포넌트
스케쥴러는 로그파일에서 각 룩업 서비스의ID 와 클라이언트의 접근 정보를 얻어낸다. 지니에서 서비스 ID 는 전역적으로 유일하기 때문에ID 가 동일한 둘 이상의 룩업서비스가 존재하지 않는다. 서비스 ID구성은 다음과 같다

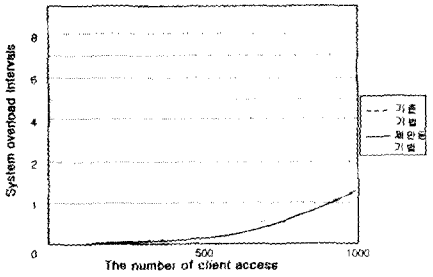


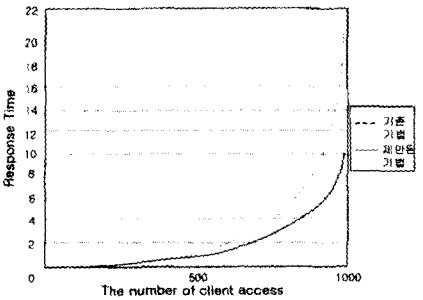
스케쥴러가 클라이언트에게 할당을 해주면 클라이언트는 그ID 로 유니캐스를 통해 서비스를 요청하게된다.이를 통해 변경된 시스템 모델을 다음에 나타내었다



제안된 시스템 구성

5. 시뮬레이션 결과





스케쥴러 객체를 통해 수집한 데이터를 가지고 2 차회귀분석을 통해 결과를 나타내었다. 첫 번째는 과부하가 걸린시간간격이 제안된기법이 사용자접속이 늘어날수록 기전의기법보다 완만한 상승곡선을 이루는 것을 볼 수 있다. 두 번째는 사용자 접근에 따른 응답 시간을 나타내는데 제안된 기법이 사용자 접근이 어느 정도 이상이 되면 응답시간 상승이 완만한 반면 기존의 기법은 점점 가파른 상승곡선을 그리며 높은 응답시가능ㄹ 보이고 있다. 위의 그래프를 통해 클라이언트접근 수가 증가할수록 제안된 기법이 효율적으로 부하를 줄이고 있음을 알 수 있다.

6. 결론 및 향후 연구 방향
우리는 지니 시스템에서 부하분산을 위해 스케쥴링과 HTTP의 Redirection 에 근거하여 지니에 적용할 수 있는 부하분산기법을 제안하였다. 기존에 없는 컴포넌트를 추가함으로써 생기는 계산과 통신 오버헤드에 기인해 클라이언트의 접근증가 초기에는 더 많은 부하를 초래하지만 접근수가 증가할수록 효율적으로 부하를 낮추는 것을 알았다
향후 과부하에 사용되는 임계값을 수학적 접근 방법을 통해 최적으로 얻는 것이 바람직하다고 생각된다.
지니는 자바가 내장되지 않은 기기에도 지니 기술을 적용시키는 서로게이트 기술과 J2ME 와 더불어 앞으로 홈네트워킹과 무선 이동통신환경에서 보다 혁신적이고 견고한 서비스를 제공할 것이다.

참고 문헌
[1] Jini Technology:http://www.sun.com/jini
[2] Edwards W.K. "Core Jini",Pretice-Hall,Inc.2000
[3] 문석현,문창주, 백두권 "IRSJ: 지니기반인터페이스공유서비스" ,고려대학교 ,2000
[4] R.Fielding, J Gettys,J. Mogul, H.Frystyk, and T. Berners-Lee Hypertext transfer protocol - http/1.1. In RFC2068,Jan . 1997
[5] Sing Li, "Professional Jini" , Wrox, 2000
[6] Isaac E Sutedja, Nicholas Vun, Hsu Wen Jing, A.Fasbendr ,Ericsson Research/Cyberlab Applications of jinni technology in wireless mobile computing environment IEEE 2000
[7] Vittorio Ghini,Fa bio Panzieri,Marco Roccetti, "Client-centered Load distribution: A Mechanism for Constructing Responsive Web Services " IEEE Trans. On Computer 2001
[8] V.Cardellini, M. Colajanni, and P. Yu. Redirection algorithms for load sharing in distributed webserve r systems.In Proc.of the 19 thIEEE International Conference on Distributed Computing System,Austin, Texas,June 1999