이중 부하제어 서비스 연결을 이용한 실시간 원격 프로시저 호출의 성능 향상 기법

이 정 훈, 강 미 경
제 주 대 학 교 전 산 통 계 학 과
{jhlee, mkkang}@venus1.cheju.ac.kr

A performance enhancement scheme for real-time RPC based on dual controlled load service connections

Junghoon Lee, Mi-Kyung Kang
Dept. of Computer Science and Statistics, Cheju National University

요 약

본 논문에서는 부하제어 서비스 클래스와 같이 결정적인 수준의 보장을 지원할 수 없는 연결 구조를 이중화하여 실시간 RPC의 종료시한 만족도를 개선하는 기법을 제안하고 그 성능을 평가한다. 제안된 기법은 RPC의 여유시간에 따라 요청 혹은 응답 메시지 전송에 있어서 종료시한 만족가능성이 높은 연결을 선택하도록 함으로써 각 RPC 트랜잭션에 여유시간에 따른 우선순위를 부여할 수 있으며 분할 기준치의 효율적인 선택에 의해 성능의 향상을 기할 수 있다. 이를 위해 할당된 대역폭에 따른 최적의 분할 기준치를 통계적으로 추정하여 실험 결과와 비교하였다. SMPL을 기반으로 수행된 모의실험은 제안된 기법이 실험에서 주어진 인자값에 대해 실시간 RPC의 종료시한 만족도를 개선할 수 있으며 통계적으로 0.3 % 이내에서 최적의 분할 기준치를 추정할 수 있음을 보인다.

1. 서론

인터넷 사용자와 서비스 제공자의 급증에 따라 다양한 서비스들이 출현하게 되었으며 사용자와 서비스들간의 상호작용 형태 또한 다양화하고 있다. 이러한 경향에 있어서 두드러진 특징 중의 하나는 멀티미디어나 트랜잭션 시스템 등과 같은 실시간 응용(real-time application)의 출현이라 할 수 있으며 실시간 서비스의 필요성이 증가함에 따라 네트워크 상에 분산된 응용들간에 실시간 상호작용 혹은 메시지 교환을 효율적으로 지원하는 방식의 필요성이 대두되고 있다[1]. 실시간 응용이 교환하는 메시지는 주어진 종료시한(deadline) 이내에 전송이 완료되어야 한다는 시간제약 조건(time constraint)을 갖는데 이를 만족시키기 위해서는 응용의 시간제약 요구나 서비스의 질(Quality of Service) 요구에 따라 응용의 수행에 필요한 자원을 할당하여야 한다. 자원의 사전 할당에 의해 일정 수준의 종료시한 만족도(deadline meet ratio)를 보장하게 되는데 보장 과정은 네트워크나 운영체제 등 하부 플랫폼의 특성에 따라 최선 노력(best-effort) 수준에서부터 결정적인(deterministic) 수준까지 제공될 수 있다. 현재 인터넷의 하부구조는 결정적인 수준보다는 최선 노력 수준으로 실시간 응용을 지원할 수 있으며 부하제어(controlled load) 서비스는 자원할당 방식의 한 예로서 실시간 연결에 대해 용량제어 기능을 제공하여 경로상의 요소들의 부하를 적정하게 유지할 뿐 아니라 자원을 사전에 할당함으로써 일부 요소가 과부하 상태가 될지라도 연결 설정 과정에서 승인된 서비스를 제공할 수 있도록 한다[2].

분산 실시간 시스템에서 프로세스들은 서로 다른 노드에 위치하여 시스템이 제공하는 특정한 상호작용에 의해 서로 협력하여 주어진 작업을 수행하게 되는데 이러한 상호작용은 메시지의 교환을 수반한다. 각 프로세스는 메시지 교환에 있어서 원격 프로시저 호출(RPC: Remote Procedure Call)이나 분산 객체 연산(Distributed Object Operation) 등 다양한 통신 프리미티브를 이용할 수 있는데 RPC는 분산 시스템에서 서비스 요청과 수행에 있어서 가장 기본적인 통신 프리미티브로서 대부분의 RPC 구현은 지역 환경에서의 프로시저 호출과 동일한 시맨틱을 제공하도록 설계되었다[3]. RPC와 같은 통신 프리미티브들의 시간제약 조건을 만족시키기 위해서는 우선적으로 시간제약 조건을 표현할 수 있어야 하며 시스템은 이에 의해 프리미티브들이 생성하는 메시지들을 응용의 요구에 맞추어 스케쥴하여야 한다. 실시간 메시지 교환의 한 예로서 실시간 CORBA(Common Object Request Broker Architecture)는 동적 CORBA 환경에서 분산 객체들간의 상호 작용에 있어서 종단간 시간제약 조건을 지원하도록 설계되었다[4].

본 논문에서는 연성 실시간 시스템을 구축함에 있어서 객체들간의 실시간 상호작용을 지원하기 위해 부하제어 서비스 클래스의 이중화된 그룹 연결을 기반으로 실시간 RPC의 종료시한 만족도를 개선하는 기법을 제안하고 성능을 평가한다. 인터넷과 같은 최선 노력 수준으로 실시간 응용을 지원하는 하부구조에서 RPC 메시지들의 종료시한 만족도를 개선하기 위해서는 메시지의 종료시한을 고려한 스케쥴링이 필요하여 종료시한이 촉박한 RPC 메시지는 종료시한에 여유가 있는 메시지 보다 신속히 전송이 완료되어야 한다[5]. 그러나 하부 네트워크가 단순히 선입선출 방식으로 전송한다면 우선순위를 부여할 수 없다. 네트워크의 부하가 낮을수록 메시지의 전송 시간은 단축되므로 이중화된 네트워크 연결을 통해 두 연결의 부하나 대역폭을 다르게 유지하고 낮은 부하의 연결을 통해 촉박한 시간제약 조건을 갖는 RPC 메시지를 전송한다면 종료시한에 따른 우선순위를 부여할 수 있게 되어 종료시한 만족도를 개선이 가능하다.

2. 실시간 RPC 스케쥴링 기법

2.1 실시간 RPC

전형적인 실시간 RPC 트랜잭션은 클라이언트로부터 서버로 호출의 전달, 서버에서의 수행, 서버로부터 클라이언트로 결과의 전달 등 순차적인 과정으로 구성된다. 클라이언트의 호출 과정은 시간제약 조건, 목적지 서버 주소 등과 아울러 수행을 원하는 프로시저에 대한 정보, 이에 관련된 입력 인자 등을 하부 네트워크를 통해 서버로 전송한다. 요청을 받으면, 서버는 메시지에 명시된 입력 인자를 기반으로 요청된 프로시저를 수행하는데 동시에 여러 개의 요청이 도착한 경우 요청들은 서버 큐에 삽입되어 추후에 수행되도록 한다. 서버 큐의 대기 정책은 서버가 수행되는 운영체제 의해 결정되며 이에 가능한 정책으로는 FCFS, EDF(Earliest Deadline First) 등이 있다[6]. 요청에 대한 수행이 완료되면 서버는 서비스를 요구한 클라이언트에게 계산 결과를 전송하는데 이러한 RPC 트랜잭션 과정이 모두 종료시한 내에 완료되어야 RPC의 시간제약 조건을 만족시킬 수 있다.

종료시한을 만족시키기 위해서는 메시지 교환에 필요한 대역폭을 사전에 할당받아야 하는데 이를 위해서는 RPC를 교환하는 노드들간에 그룹 연결이 설정되어 있어야 한다. 그룹 연결을 설정함에 있어서 부하제어 서비스는 연결에 개입된 네트워크 요소들의 자원을 할당받는다. 이때 상이한 경로를 이용하여 두 개 이상의 연결을 설정할 수 있을 뿐 아니라 각 경로의 대역폭도 다르게 할당받을 수 있다. 특정 경로를 통한 연결이, 응용이 요구하는 대역폭을 모두 할당할 수 없는 경우도 발생하며 이 경우 충분한 대역폭을 할당받기 위해서는 또다른 그룹 연결을 설정하여야 한다. 결국 RPC를 교환하는 노드들간에 다수의 그룹 연결을 설정하여 대역폭을 할당받는다. 하나의 RPC 트랜잭션은 하나 이상의 노드가 개입된 일련의 수행 과정을 포함하므로 트랜잭션 수행 중에 종료시한을 초과할 수 있는데 이 경우 네크워크 큐에서나 서버 큐에서 사전에 종료시한 초과 여부를 판단하여 종료시한이 초과한 트랜잭션을 기각시켜야 한다.

2.2 이중 연결 상의 스케쥴링 기법

각 노드에 있어서 요청 혹은 결과 메시지가 생성되었을 때, 두 연결 중 하나를 선택하여 전송되어야 하는데 분배하는 기법으로서 균등 분할(even partition) 방식과 여유시간 분할(slack partition or chop partition) 방식을 고려할 수 있다. 균등 분할 방식은 메시지를 두 연결에 똑같은 확률로 할당하여 충분한 네트워크 부하에서 메시지의 전송시간을 반으로 감소시킨다. 반면 여유시간 분할 방식은 각 RPC 여유시간에 따라 분배하는데 여유시간은 일반적으로 작업의 종료시한과 실행시간 사이의 차이로서 정의되며 작업의 시간제약 조건을 만족시키기 위해서는 늦어도 여유 시간 이내에 작업의 수행이 시작되어야 한다. 균등 분할 기법은 각 연결 큐의 길이와 전송시간을 평균적으로 같게 하여 두 연결의 RPC 부하를 동일하게 유지하므로 단일 그룹 연결로 모든 대역폭을 할당받는 경우와 동일한 특징을 갖는다. 그러나 각 RPC 메시지는 다른 여유시간을 갖기 때문에 촉박한 종료시한을 갖는 RPC가 생성한 메시지는 신속히 전송되어야 하는 반면 여유시간이 충분한 RPC에서 생성된 메시지는 전송시간이 여유시간 범위 내에서 연장되어도 종료시한을 만족시킬 수 있다.

여유시간 분할 방식은 각 RPC 메시지를 트랜잭션의 여유시간에 따라 각 연결에 분배하며 RPC의 여유시간을 계산하는데 필요한 수행시간은 서버에서의 수행시간과 네트워크 전송 시간의 합으로 구성된다. 서버에서의 수행시간은 사전에 예측이 가능하거나 최소한 최악의 수행시간에 대해 계산할 수 있는데 반해 인터넷과 같은 네트워크에서의 수행시간은 정확한 예측이 불가능하므로 메시지의 길이에 의해 결정되는 순수한 전송시간으로 설정한다. 결국 같은 RPC에 속한 요청과 응답 메시지는 같은 연결을 통해 전송된다. 여유시간 분할 방식은 RPC 트랜잭션들의 여유시간 분포가 사전에 알려져 있다는 가정 하에 한 연결의 부하를 낮게 유지하여 여유시간이 촉박한 메시지들을 전송하고 또다른 연결에는 종료시한 만족에 있어서 비교적 여유가 있는 메시지를 전송한다. 이를 위해 그림 1에서 보는 바와 같이 분할 기준치(chop value) p를 기반으로 네트워크를 선택하는데 여유 시간이 B. Gao 등의 연구에서처럼 S_{min}에서 S_{max}까지 균일하게 분포한다고 가정한다면 여유시간 분포상 0부터 p 범위에 해당하는 메시지들은 그룹 연결 1을 통해 전송하고 나머지 $(1-p)$에 해당하는 메시지들은 그룹 연결 2를 통해 전송하도록 분배한다[7].

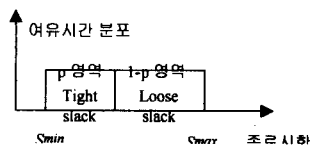

그림 1. 여유시간 분할

여유시간 분할 방식은 분할 기준치의 값에 의해 그 성능이 좌우된다. 만약 시스템의 인자들이 변하지 않는

정적인 특성을 갖고 있다면 사전에 다양한 모의실험이나 환경 실험을 수행하여 인자들과 최적의 분할 기준치에 대한 관계를 통계적인 모델에 의해 구한 후 시스템 운영 중에 이 모델에 의해 분할 기준치를 설정할 수 있다. 반면 시스템의 인자들이 동적으로 변화한다면 분할 기준치를 이에 따라 같이 변경하여야 하는데 이러한 동적 적응을 위해서는 네트워크 상에 조정자(coordinator) 노드가 필요하다. 조정자 노드는 네트워크 상의 다른 노드들로부터 주기적으로 RPC 부하와 같은 인자들과 현재의 종료시한 만족도와 같은 상태 정보들을 수집하여 새로운 분할 기준치를 설정하고 이를 방송에 의해 다른 노드들에게 알려 분할 기준치를 갱신하도록 한다.

3. 성능 평가

본 절에서는 SMPL을 사용한 모의실험을 통하여 제안된 RPC 스케쥴링 기법의 성능을 측정한 결과를 제시하고 분석한다[8]. 모의실험에 있어서 주요한 가정은 다음과 같다. 시스템에서 모든 RPC 트랜잭션들은 동일한 중요도를 가지며 종료시한 만족도가 가장 중요한 시스템 성능 요소라고 가정한다. 또한 각 RPC들은 다른 RPC 트랜잭션과 독립적으로 수행되며 또한 서버는 EDF 정책에 따라 수신된 요청을 스케쥴하고 수행한다. 모의실험은 10 Mbps를 할당받은 이중의 그룹 연결에 기반하고 있으며 실험의 단순화를 위해 추가적인 분할이나 재조립 과정이 없이 하나의 메시지를 통하여 전송될 수 있도록 요청 메시지의 평균 길이는 500 바이트, 반면 응답 메시지의 평균 길이는 1000 바이트로 설정하여 지수 분포를 따르도록 하였다.

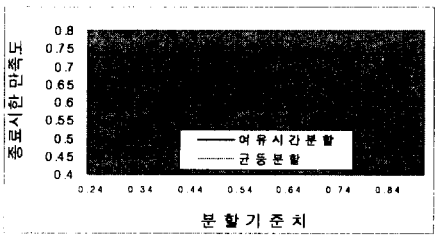

그림 2. 분할기준치에 따른 종료시한 만족도

그림 2는 5.5 Mbps와 4.5 Mbps를 할당받은 두 연결에 있어서 분할 기준치의 영향을 보인다. 실험에서 RPC의 평균도착간격 시간은 0.0008초, 평균 서비스 시간은 0.0001초, 평균 여유시간의 비율은 32로 설정하였는데 다른 인자값들을 사용한 경우에도 그림과 같은 형태의 종료시한 만족도 패턴을 보인다. 위 실험은 결국 분할기준치의 선택이 단일 10 Mbps 연결과 균등분할에 비해 종료시한 만족도를 개선할 수 있음을 보인다. 대역폭을 변화시켜가며 향상도를 측정한 결과 2에서 5 %까지의 향상을 기할 수 있다.

제안된 스케쥴링 기법에 있어서 주어진 네트워크 인자에 대해 최적의 분할 기준치를 설정하는 것이 시스템의 실시간 성능에 가장 중요한 과정인데 이는 실험에 의한 통계 분석에 의해 결정될 수 있다. 식(1)은 한 네트워크의 대역폭 B_1이 5.5 Mbps에서 8.5 Mbps까지 변화함에 따라 최적의 성능을 보이는 분할 기준치 c를 통계적으로 추정한 결과이다[9].

$$c = 0.0914B_1 + 0.07419 \qquad (1)$$

물론 다른 RPC 인자를 사용하면 이에 따라 상수값들이 변화한다[10]. 이와 아울러 그림 3은 실험에 의해 찾아진 최적의 분할 기준치와 식 (2)에 의해 추정된 값으로 분할 기준치를 설정하였을 때 첫번째 네트워크의 대역폭 할당량에 따른 종료시한 만족도를 보이고 있는데 성능의 차이는 최대 0.3 % 이내로서 통계적 분석에 의한 방식은 효율적으로 분할 기준치를 설정할 수 있다.

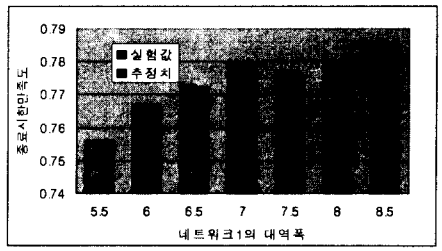

그림 3. 추정치와 실험값의 비교

4. 결론

본 논문에서는 이중의 부하제어 연결을 기반으로 하여 실시간 RPC의 종료시한 만족도를 개선할 수 있도록 RPC 트랜잭션의 여유시간에 따라 RPC 메시지들이 전송될 연결을 선택하는 기법을 제안하고 그 성능을 모의실험에 의해 측정하였다. 적정한 분할 기준치의 설정에 의해 단일화된 연결이나 균등분할 방식에 비해 RPC의 종료시한 만족도를 개선할 수 있음을 보였으며 이 분할 기준치는 통계적으로 효율적으로 계산될 수 있었다. 그러나 보다 다양한 RPC 인자와 대역폭이 다양하게 설정된 경우를 위하여 해석적 모델이 구해져야 한다.

5. 참고 문헌

[1] K. Arvind, Krithi Ramamritham and John A. Stankovic, "A local area network architecture for communication in distributed real-time systems," *JRTS*, Vol. 3, pp.115-147, May 1991.
[2] J. Wroclawski, *Specification of the Controlled-Load Element Service*, RFC 2211, September 1997.
[3] A. Birrel, B. Nelson, "Implementing remote procedure calls," *ACM TCS*, pp.39-59, 1984.
[4] D. Schmidt, D. Levine, S.Mungee, "The design and performance of real-time object request brokers," *Computer Communications*, pp.294-324, 1998.
[5] Marina Dao and Kwei-Jay Lin. "Remote procedure call protocols for real-time systems," *Proceedings of IEEE Euromicro Workshop*, pp.216-223, 1991.
[6] K. Ramamritham, J. Stankovic, "Dynamic task scheduling in hard real-time systems," *IEEE Software*, pp.65-75, July 1984.
[7] B. Gao, H. Garcia-Molina, "Scheduling soft real-time jobs over dual non-real-time servers," *IEEE TPDS*, pp.56-68, Jan. 1996.
[8] M. H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987.
[9] D. C. Frank, *Sas Applications Programming : A Gentle Introduction*, PWS Co., 1991.
[10] J. Lee, S. Kim, "Design of a real-time RPC scheduling scheme over dual networks," *ICACT*, pp.90-94, 2000.