

RT-Mach에서 우선순위 반전을 줄이기 위한 모델 연구

최대수^o, 임종규, 이태현, 구용완
수원대학교 전자계산학과

The Study of Model for Reducing Priority Inversion in RT-Mach

Dae-Soo Choi^o, Jong-Kyu Im, Tae-Hyun Lee, Yong-Wan Koo
Dept. of Computer Science, The University of Suwon

요약

실시간 시스템은 서버와 같은 공유자원들에 대해서 보다 높은 우선순위 활동의 최악 블록 시간(worst case blocking time)을 결정해야 한다. 현재까지 분산 시스템을 위한 여러 서버 모델들이 제안되었으며 이러한 서버모델들에 우선순위 규약을 적용하여 실시간성을 높이고 있다. 본 논문에서는 RT-Mach에서 실시간성을 향상시키기 위한 새로운 모델을 제시하여 우선순위 반전 문제를 해결하기 위해 동적 스레드 관리자(dynamic thread manager)를 제안하였다. 기존의 여러 서버 모델들과 달리 서버마다 동적 스레드 관리자를 두어 작업자 그룹내에서 유휴 스레드(idle thread)와 바쁜 스레드(busy thread)를 판별하여 적절한 스레드를 선정, 조정하고 서버로부터의 요구에 대해 최상위 우선순위를 할당하여 서버간의 우선순위 반전을 줄이고 실시간성을 향상시키고자 했다.

1. 서론

분산 실시간 시스템은 항공기 제어, 통신 시스템, 군용 시스템, 공장 자동화, 로봇 공학 등 다양한 분야에 응용되고 있다. 실시간 시스템(Real-Time System)이란 외부의 비동기적인 사건에 대하여 예측 가능한 시간 내에 응답을 보장할 수 있는 시스템이다[2]. 반면에 이러한 목적에 맞는 엄격한 시간 요구사항을 만족하기 위해서는 실시간 시스템이 예측 가능하고 신뢰성 있는 환경을 제공해야만 한다. 그러나 분산 실시간 시스템에서 다양한 실시간 활동들의 엄격한 시간 요구를 만족시키는 것은 시스템의 분산된 특성 때문에 점차 복잡해지고 있다. 따라서, 사용자에게 예측가능하고(predictable) 신뢰성 있는(reliable) 분산 실시간 컴퓨팅 환경을 제공하는 Mach커널의 실시간 버전이 개발되었다[1].

예측가능하고 분석가능한 실시간 작업의 스케줄링을 위해서는 높은 우선순위(high priority) 작업이 낮은 우선순위(low priority)작업을 언제나 선점(preemption)할 수 있도록 하는 우선순위(priority)에 기반을 둔 선점 스케줄링 기법이 사용되어야 한다. 그러나 실시간 작업들이 공유자원을 사용할 경우 이들의 동기순서화를 위하여 사용되는 도구들에 의해 높은 우선 순위 작업이 낮은 우선순위 작업에 의해 블록되는 경우가 발생할 수 있다. 이러한 현상을 우선순위 반전(Priority Inversion)이라 한다[2]. 만일 이러한 우선순위 반전의 기간이 비결정적(nondeterministic)으로 길어지는 무제한 순위반전(unbounded priority inversion)은 실시간 시스템의 스케줄 가능성(schedulability) 및 예측가능성(predictability)을 떨어뜨린다. 따라서, 이러한 순위반전기간을 최소화하여 실시간 시스템의 성능을 높이는 연구가 필요하다.

RT-Mach에서는 여러 실시간 서버 모델들이 연구되었는데 이는 예측가능한 서비스 시간을 보장하기 위한 틀을 제공한다. 서버들의 구조와 우선순위 관리는 서버 모델에서 중요하다. 서버모델은 서버에서 스레드들의 수와 요구들을 처리하기 위한 스레드 생성시간에 의해 단일 스레드 서버 모델, 작업자 모델, 동적 서버 모델로 구분된다[3]. 이러한 서버모델들이 갖는 문제점 중 하나가 실시간 처리에 있어서 우선순위 반전 현상이다. 이러한 우선순위 반전을 줄이기 위한 연구가 많이 행해졌지만 부분적인 우선순위 허용등으로 인해 실시간

시스템에는 적합하지 않다. 본 연구에서는 기존 서버모델에 동적 스레드 관리자를 두어 다른 서버로부터 발생하는 요구뿐만 아니라 로컬에서 발생한 요구도 적절한 스레드에게 할당하여 우선순위 반전을 줄일 수 있다.

본 논문은 2장 관련연구에서 우선순위 반전 문제와 RT-Mach시스템에 대해 기술한다 3장에서는 기존에 제안된 서버들의 문제점을 파악하여 본 논문이 제안한 서버 모델을 기술한다. 마지막으로 4장에서는 제안한 서버 모델의 평가 와 향후 연구 방향에 대해 기술한다.

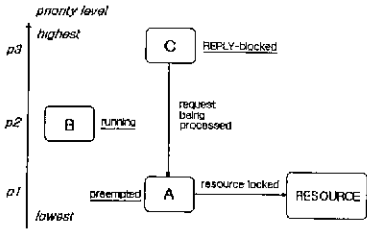
2. 관련연구

2.1 우선순위 반전과 해결책

우선순위 반전 현상은 프로세스 우선순위 결합과 프로세스들 사이의 자원공유에서 발생한다. 우선순위 반전의 두 시나리오를 [그림 1]과 [그림 2]에서 보았다.

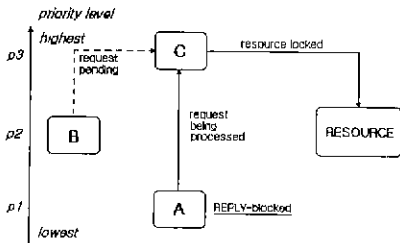
[그림 1]에서 자원은 3개의 프로세스들 중에서 가장 낮은 우선순위(p1)를 갖는 프로세스 A에 의해 사용중일 수 있다. 가장 높은 우선순위(p3)를 갖는 프로세스 C는 인터럽트를 얻어 즉각적으로 자원에 대한 접근을 요구한다 A에게 요구를 보내고 나서 응답이 올 때까지 블록된다 A는 스케줄을 얻어서 수행을 시작한다. A가 C로부터의 요구를 처리하고 있는 동안에 중간 우선순위(p2)를 갖는 프로세스 B가 인터럽트를 받아서 A를 선점하고 수행을 시작한다. 왜냐하면 B가 A보다 높은 우선순위를 갖기 때문이다 결과적으로 C는 B가 수행을 멈추고 A가 C의 요구에 대한 처리를 끝낼 때까지 기다려야만 한다. 사실상 높은 우선순위 프로세스 C는 낮은 우선순위 B 또는 A에 의해 블록 된다. 이러한 상황은 우선순위 반전의 한가지 예이다. 프로세스 B가 얼마나 수행시간이 소모될지 알 수 없기 때문에 임계 프로세스(C)가 트랜잭션을 완료하는데 필요한 시간을 예측할 수 없게 된다. 이것은 어플리케이션의 시간요구사항을 만족하기 어렵게 만든다 이러한 문제를 피하기 위하여 실시간 운영체제(RTOS)는 우선순위 상속의 개념을 이용한다 [그림 1]에서 RTOS는 가장 낮은 우선순위 프로세스(A)를 서비스를 요구하는 프로세스의 우선순위(C)까지 상승시킨다 우선순위 상속은 블록 프로세스(그림1에서 A)가 블록된 프로

세스(그림1에서 C)의 우선순위를 상속받음을 의미한다. 물론 블록된 프로세스가 높은 우선순위를 갖을 때에만 의미가 통한다.



[그림 1] 우선순위 반전 - low priority server

만약 서버가 어플리케이션에서 가장 높은 우선순위를 가지고 확장된 시간동안 수행한다면, [그림 2]에서 나타난 우선순위 반전 문제를 일으킬 수 있다. 자원은 단지 우선순위가 가장 높은 서버 C에 의해 서만 접근될 수 있을 때 가장 낮은 우선순위(p1)를 갖는 프로세스 A는 자원에 대한 접근을 C에게 요구한다. C가 수행중이고 A로부터의 요구를 처리하는 동안 중간 우선순위(p2) 프로세스 B는 같은 자원을 요구한다. 이때 B는 서버 C가 프로세스 A에 의한 처리 요구를 완료할 때까지 기다려야만 한다. 사실상 높은 우선순위 B는 가장 낮은 우선순위 A에 의해 블록 되는 경우이며 우선순위 반전의 또 다른 예이다. 우선순위 상속은 이러한 문제도 해결할 수 있다.



[그림 2] 우선순위 반전 - high priority server

우선순위 상속규약(priority inheritance protocol)은 낮은 우선순위의 요구를 서버가 수행할 때 높은 우선순위의 요구가 대기하는 기간을 줄이기 위하여 서버의 우선순위를 대기하는 요구들의 우선순위중 가장 높은 우선순위로 대체한다. 우선순위 핸드오버 규약(priority handoff protocol)은 높은 우선순위의 요구 메시지를 낮은 우선순위의 서버가 수행중일 때 중간 우선순위의 요구가 이를 선점하는 경우를 막기 위하여 요구의 우선순위를 서버가 갖는다. 우선순위 상한 규약(priority ceiling protocol)은 우선순위 상속 규약의 문제점인 교착상태와 연쇄블록킹을 해결하기 위하여 모든 임계구역에 순위를 부여하고 어떤 작업이 자신의 임계구역에 들어가고자 할 때 자신의 순위가 다른 작업에 의해 수행중인 임계구역의 순위보다 높아야 한다는 조건을 우선순위 상속 규약에 추가한 방법이다[2]

2.2 Real-Time Mach

Mach는 CMU(Carnegie Mellon University)에서 개발한 마이크로 커널(microkernel)을 기반으로 하는 개방형 운영체계의 시초이다. 마이크로 커널은 IPC, 단기 스케줄링, 메모리 관리 같은 필수적인 서비스만을 갖는 운영체제이며, 개방형 구조로서 마이크로 커널과 서버로 구성된다. 각 서버(server)는 메시지 전달(message passing) 방식을 이용하여 마이크로 커널을 접근한다. 커널 주소공간에서 실행되는 코드는 최소화하고 응용 프로그램 수준의 서버에서 실행되는 코드는 최대화 한 것이다.

RT-Mach는 CMU의 ART(Advanced Real-Time Technology)그룹에서 개발한 것으로 RT-Mach의 목적은 사용자에게 예측가능하고

신뢰성있는 분산 실시간 환경을 제공하는 Mach커널의 실시간 비전을 개발하는 것이다. RT-Mach의 특징으로는 스레드 모델에 시간 속성을 추가한 실시간 스레드 모델, 통합된 실시간 스케줄러의 제공, 커널은 단지 하드웨어 자원을 관리하기 위한 수단만을 제공하며 사용자 모드 TASK들이 그 수단을 이용해 정책을 구현하는 스케줄러에서 policy/mechanism의 분리, 다양한 실시간 동기화 메카니즘, 메모리 상주 객체들의 지원을 들 수 있다[1].

RT-Mach에서는 5가지 동기화 모델을 선택적으로 사용한다. KM(Kernelized Monitor)는 스레드가 커널내의 임계영역내에 있을 때 선점할 수 없다. 그래서 우선순위 반전의 기간은 임계영역의 크기로 한정된다. BP(Basic Priority)는 대기중인 스레드가 스레드의 우선순위를 기반으로 하여 큐에 대기된다. BPI(Basic Priority Inheritance)에서 높은 우선순위 스레드는 낮은 우선순위 스레드가 수행중인 임계영역에 요구를 할 때 임계영역은 높은 우선순위를 상속받는다. PCP(Priority Ceiling Protocol)는 임계영역의 우선순위 상한이 다른 스레드들의 임계영역보다 높지 않다면 스레드의 수행은 블록된다. RCS(Restartable Critical Section)은 임계영역이 높은 우선순위 스레드에 의해 중지되고 높은 우선순위 스레드가 작업을 완료한 후 낮은 우선순위 스레드는 재시작 한다[6].

3. 본론

3.1 실시간 서버 모델의 문제점

RT-Mach에서는 여러 가지 서버 모델이 제안되었다. 각각의 서버 모델은 실시간 자원의 관점에서 문제점을 고려해 보면 다음과 같다. 첫째, 단일 스레드 서버 모델(single thread server model)은 서비스를 제공하는 서버 스레드를 한 개로 구성하는 방식이다. 서버 스레드가 하나의 요구를 수행하는 동안 다른 요구는 메시지큐에 삽입된다. 우선순위 반전 대책 규약을 모두 사용하는 Mach IPC가 주어진다 해도 근본적으로 높은 우선순위 실시간 요구의 오랜 대기는 피할 수 없다. 또한 서버가 수행 도중 I/O 통과 같은 커널 서비스를 요구하는 경우 서버는 블록되므로 후속 요구를 위한 처리를 시작하지 못한다. 둘째, 동적 우선순위 작업자 모델(dynamic priority worker model)은 정해진 개수의 서버 스레드로 작업자 그룹을 형성하고 우선순위 반전 대책 규약을 사용하여 일의의 사용 가능한 서버 스레드에게 요청을 할당하는 방식이다. 모든 스레드들이 낮은 우선순위 요구를 처리하고 있을 경우, 높은 우선순위의 요구가 낮은 우선순위 요구를 기다려야 하는 문제가 발생한다. 셋째, 정적 우선순위 작업자 모델(static priority worker model)은 정해진 개수의 서버 스레드로 작업자 그룹을 구성하고 각 스레드에 각기 다른 한정된 우선순위 요구만을 서비스하도록 함으로서 낮은 우선순위 요구들에 의한 서버들의 목적을 방지하는 방식이다. 그러나 이는 우선순위 단계를 모두 피해야 하며 우선순위 단계가 많아질 경우 상대적으로 작업자 스레드의 개수가 많아져야 하는 오버헤드가 있다. 마지막으로 동적 서버 모델(dynamic server model)은 새로운 요구가 들어올 때마다 새로운 스레드를 생성한다. 이 방식은 가장 높은 선점가능성을 가지며 우선순위 반전 문제를 일으키지 않지만 실시간 요구 발생 후 서버 스레드를 생성할 경우가 있을 수 있고, 또한 서버를 생성하거나 소멸시키는 데 별도의 시간과 노력이 필요하므로 시간적 오버헤드와 시간 예측이 어렵다. 이 모델의 가장 중요한 문제는 스레드가 동적으로 생성되기 때문에 서버의 최대 자원 사용량을 측정하기 어렵다는 것이다. 따라서, 본 모델은 실시간 서버에는 적합하지 않기 때문에 RT-Mach에서는 단일 스레드 서버 모델과 작업자 모델만 이용한다.

동적 우선순위 작업자 모델은 우선순위 반전을 어느 정도 허용하면서 오버헤드를 줄이고, 정적 우선순위 작업자 모델은 우선순위 반전이 일어나지 않도록 함으로서 선점가능성을 높이는 데 목적이 있다. 두 모델의 특성을 결합한 연관 우선순위 작업자 모델(associative priority worker model)이 제안되기도 했다[7].

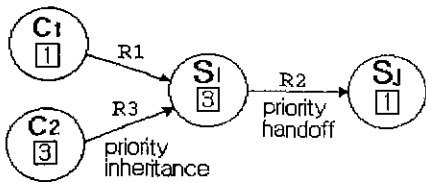
본 모델은 한 서버에 있는 스레드가 우선순위 범위를 갖고서 그 범위에 속한 요구가 들어왔을 때 요구를 처리하게 된다. 그러나 상기 모델에는 치명적 오류가 발생할 가능성이 있다. 만약 클라이언트 요구들의 우선순위가 하나의 서버 작업자 스레드 우선순위 범위에 접

중적으로 요구될 경우 그 범위를 갖는 스레드 요구들은 그 스레드에 계속적으로 사상(mapping)되어 큐길이를 증가시킨다. 반면 다른 우선순위 범위를 갖는 스레드는 요구가 없어서 계속적으로 유휴상태로 있을 경우가 발생한다. 이러한 경우 유휴 스레드를 갖으면서도 요구들이 계속적으로 대기하게 되어 요구들이 데드라인을 위반할 수 있으므로 실시간성에 위배된다.

3.2 서버간의 우선순위 반전

RT-Mach는 운영체제의 여러 기능을 일반적으로 단일 서버로 구현하지 않고 여러 계층의 서버로 구성하여 운영체제 서비스를 제공한다. 다중 서버로 구성되어 있기 때문에 서버에서 다른 서버에게 요구를 요청할 수 있는데 이 경우 단일 서버 모델에서와 같은 우선순위 반전 현상이 발생한다.

[그림 3]에서 우선순위 상속과 우선순위 핸드오프를 모두 제공하는 서버 모델을 보자. 우선순위가 1인 클라이언트 C₁이 서버 S_i에게 서비스를 요구한다. 서버 S_i는 이를 수행하던 중 이에 관련된 동시적 서비스를 S_j에게 요구하고 대기상태가 된다. 이때 서버 S_j는 우선순위 핸드오프에 의해 1이 되고 서버 S_i는 우선순위 3인 클라이언트 C₂의 요구메시지가 대기하므로 3이 된다. 이러한 상황에서 실질적 서비스를 수행하고 있는 서버 S_j는 우선순위가 1이고 대기 클라이언트는 우선순위가 3이므로 우선순위 상속과 핸드오프를 모두 제공하였어도 심각한 우선순위 반전이 일어나게 된다. 작업자 모델에서는 다중 스레드된 서버로 이러한 현상을 줄일 수 있지만 클라이언트의 개수가 많을 경우 같은 현상이 발생한다. 즉, 기존의 서버 모델들의 서버간 요구에서 우선순위 반전현상은 우선순위 상속과 우선순위 핸드오프를 이용하더라도 해결되지 않는다[4][5].



[그림 3] 우선순위 반전

이에 대한 해결방법 중 동시적 요구와 비동시적 호출의 조합에 의한 방법이 있다[4]. 이는 클라이언트가 서버에게 동시적인 요구를 하고 서버가 다른 서버를 호출하고 블록되는 현상을 방지하기 위하여 비동시적으로 서비스를 요청하는 방법으로 우선순위 반전현상을 개선하고자 한 방법이다. 그러나 이 방법은 단조형(Monolithic) 커널 방식과 같은 개념으로 Mach 서버의 CSP(Communicating Sequential Processes)적 개념의 장점과 간단성을 해치게 된다.

3.3 제안한 서버 모델

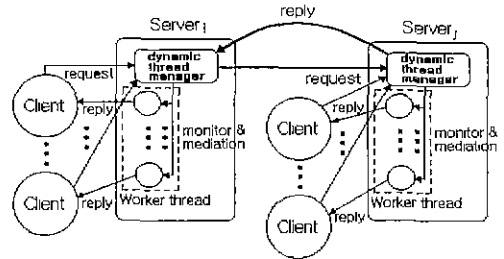
본 연구에서는 연관 우선순위 작업자 모델의 문제점 제거 및 서버간 호출 발생시 피호출 서버의 블록 발생을 예방하여 마감시간을 만족시키는 모델을 제시하였다.

제안된 모델은 각 서버마다 동적 스레드 관리자를 두어서 모든 서버에 대한 요청을 통합 관리한다. 동적 스레드 관리자는 다음과 같은 두가지 기능을 갖는다. 첫째, 클라이언트에 의해 서버가 직접 호출되었을 때 요청을 받아 정적으로 우선순위 범위에 따라 적절한 작업자 그룹내 스레드를 선정하고 둘째, 다른 서버로부터의 요청시에도 블록예방을 위해 요구를 받아 가장 높은 우선순위로 스레드에 할당하여 처리하는 역할을 담당한다. 상기 두가지 과정에서 이미 작업자 그룹내 해당하는 스레드가 사용중인 경우, 자신보다 우선순위가 하위 레벨에 있는 작업자 스레드가 유휴한 상태인가를 판단하여 유휴한 경우 그곳에 본 요청을 할당하게 한다. 이때 작업자 그룹의 스레드 우선순위는 우선순위 상속 규약에 따라서 클라이언트의 요청 우선순위를 일시적으로 상속받게 된다. 자신의 작업자 그룹 스레드가 아닌 하위 스레드에서 처리된 요청이 완료시 작업자 그룹의 스레드는 원래의 우선순위로 환원된다. 이러한 처리는 연관 우선순위 작업자 모델의 취약점을 해결하여 스케줄 가능성 및 데드라인을 만족할 수 있게 된다.

제안한 동적 스레드 관리자에 두가지 기능을 고려하면 실시간성을 높일 수 있으리라 사료된다. 첫째는 각 작업자 스레드마다 history정

보를 두어 하위레벨 스레드에 요청 사상시 각 스레드마다 몇회의 요구가 있었는지 조사하여 가장 적게 요구가 발생한 스레드에게 클라이언트의 요청을 사상하는 방법이며, 둘째는 작업자 그룹의 스레드에서 처리가능한 우선순위 범위를 정적으로 할당시 기존의 모델은 모두 일률적인 범위로 할당하였으니 이는 높은 우선순위 요구에게 불리한 작용을 한다. 이의 개선책은 낮은 우선순위 스레드의 범위를 넓게, 높은 우선순위 스레드의 범위를 좁게 할당하여야 할 것이다. 이것의 이유는 높은 우선순위 요청에 대해 넓은 범위 할당시 해당하는 스레드에 큐입될 수 있는 클라이언트 요청이 많을 수 있기 때문에 그만큼 스케줄가능성을 약화시키는 결과를 초래할 수 있다.

동적 스레드 관리자의 서버간 호출처리 기능은 다른 서버로부터의 요구에 대해서도 최상위의 우선순위를 부여하여 유휴한 스레드에 할당하거나 유휴한 스레드가 없을 경우 우선순위가 가장 낮은 스레드를 선정하여 처리토록 하는데 있다 이로써 서버요구에 대해 가장 빠른 응답을 보내준다



[그림 4] 제안한 서버 모델

4. 결론 및 향후 연구

실시간 시스템은 서버와 같은 공유자원들에 대해 보다 높은 우선순위 활동의 최악 블록 시간(worst case blocking time)을 결정해야 한다. 분산 시스템을 위한 여러 서버 모델들이 제안되었으며 이러한 서버모델들에 우선순위 규약을 적용하여 실시간성을 높이고 있다.

본 논문에서는 RT-Mach에서 실시간성을 향상시키기 위한 새로운 모델을 제시하여 우선순위 반전문제를 해결하기 위한 동적 스레드 관리자를 제안하였다. 서버마다 동적 스레드 관리자를 두어 연관 우선순위 작업자 모델의 단점을 해결하였으며 또한 서버로부터의 요구에 대해서도 최상위 우선순위를 할당하여 서버간의 우선순위 반전을 줄이고 실시간성을 향상시키고자 했다.

본 제안된 모델은 기존의 모델들과 비교 평가 및 RT-Mach상에 실제 구현할 필요가 있다.

5. 참고문헌

- [1] H. Tokuda, T. Nakajima, and P. Rao, "Real-Time Mach: Towards a Predictable Real-Time System", In *Proceeding of USENIX 1st Mach Symposium*, October, 1990
- [2] 이기용, "실시간 시스템에서의 순위반전 대책 규약들의 성능분석", 석사학위논문, 1994
- [3] T. Nakajima, T. Kitayama, and H. Tokuda, "Experiments with Real-Time Servers in Real-Time Mach", In *Proceedings of USENIX 3rd Mach Symposium*, 1993
- [4] 이영길, "서비체인에서의 우선순위 반전현상을 줄이기 위한 실시간 객체 기반 서비스 요청 구조 설계 및 구현", 석사학위논문, 1996
- [5] T. Kitayama, T. Nakajima, H. Arakawa and H. Tokuda "Integrated Management of Priority Inversion in Real-Time Mach", *IEEE Real-Time Systems Symposium*, December, 1993
- [6] H. Tokuda and T. Nakajima, "Evaluation of Real-Time Synchronization in Real-Time Mach", In *Proceeding of USENIX 2nd Mach Symposium*, 1991
- [7] 천경아, 박홍진, 김영찬, 전성희, "우선순위 계층 프로토콜을 이용한 실시간 서버의 모델 연구", 한국정보과학회, 1997