

분산 실시간 시스템에서 신뢰성 향상을 위한 운영체제 영향 요소 분석

구현우^o 홍영식
동국대학교 컴퓨터공학과
{hwgoo^o, hongys}@dgu.ac.kr

Analysis of OS Timing Factor for Improving Reliability in Distributed Real-Time Systems

Hyun-Woo Goo^o Young-Sik Hong
Dept. of Computer Engineering, Dongguk University

요 약

실시간 시스템은 논리적 정확성뿐만 아니라 시간적 정확성을 요구한다. 시간적 정확성을 만족시키기 위해 실시간 시스템의 설계자는 작업들의 스케줄 가능성에 대한 연구를 진행해야만 한다. 그리고 스케줄 가능성 분석을 위해 프로그램들에 대한 실행 시간의 예측이 필요하다. 프로그램 또는 작업들의 실행 시간 예측을 위한 방법으로 측정과 정적 분석이 연구되었다. 측정 및 정적 분석은 비용 및 확장성에 문제점을 지니고 있고 실시간 시스템의 발전을 따라가지 못하여 분석 결과의 정확성 및 신뢰성이 만족스럽지 못한 경우가 발생한다. 본 논문에서는 정적 분석을 단일 실시간 시스템이 아닌 분산 실시간 시스템에 적용할 수 있는 확장된 정적 분석 도구의 개발에 초점을 둔다. 특히, 확장된 정적 분석 도구의 개발을 위해 운영체제에서 발생하는 작업 실행 영향 요소 분석 과정을 설계한다. 실시간 시스템에서 시간적 정확성을 만족하기 위해 스케줄링 기법이 가장 중요하고 이러한 스케줄링 기법은 운영체제 영향 요소 분석 대상 중 가장 중요한 요소이다. 이에 따라 스케줄러의 동작 과정의 정적 분석 및 우선 순위에 따른 작업의 큐 대기 시간 예측을 통해 원시 프로그램에서 자동적으로 예측된 실행 시간의 정확도와 신뢰도를 높인다.

1. 서 론

실시간 시스템의 가장 큰 특징은 시스템에서 요구되는 논리적 정확성, 뿐만 아니라 시간적 정확성을 필요로 한다는 것이다. 시간적 정확성을 만족시키기 위해 스케줄 가능성(Schedulability)에 대한 연구가 필요하였고 지금까지 많은 연구가 진행되어 왔다. 스케줄 가능성 분석은 실시간 시스템 설계에 있어 가장 중요하고 어려운 작업이다. 스케줄 가능성 분석을 위해 시스템에서 운용되는 작업의 실행 시간 분석 및 예측이 선행되어야 한다. 1)

과거 작업의 실행시간에 대한 분석은 작업 자체의 최악시간 만을 고려하였다. 하지만 보다 정확한 실행시간의 예측을 위해서는 운영체제 및 통신 부하를 예측하여 확장된 정적 분석 기법 및 동적 분석 기법이 필요하다. 이에 본 논문에서는 분산 실시간 시스템에서 운영체제에 의해 발생하는 부하의 분석을 통해 신뢰성 있는 분산 실시간 시스템 설계가 가능케 한다. [1,6]

특히, 운영체제에 의해 발생하는 부하를 예측하기 위해서는 커널 모듈의 동작 시간 및 작업의 큐 대기시간 예측이 필요하다. 본 논문에서는 커널의 스케줄링 모듈

의 동작 시간 및 작업의 대기시간 예측을 통해 운영체제 부하를 예측하고 이를 작업의 정적 분석 기법에 추가하여 보다 정확하고 신뢰성 있는 분산 실시간 시스템의 실행 시간을 분석한다.

본 논문의 2장에서는 정적 실행 시간 분석 및 작업들의 큐 대기 시간 예측을 위한 관련 연구를 살펴보고 3장에서는 실시간 분석 도구의 전체 구조를 살펴보고 4장에서는 작업의 실행 시 발생하는 운영체제에 의한 영향 요소 중 스케줄링에 관한 영향 요소를 살펴본다. 특히 스케줄링을 담당하고 있는 스케줄 모듈을 중점적으로 정적 분석한다. 5장에서 실행 시간 분석의 신뢰도를 높이기 위한 향후 연구 과제를 언급하기로 한다.

2. 관련 연구

실시간 시스템의 중요한 요소인 시간적 정확성을 만족하기 위해 실시간 작업 스케줄링 기법에 관한 많은 연구가 진행되어 왔다. 그와 병행하여 작업들의 스케줄 가능성의 분석을 위한 실시간 시스템에서 실행될 프로그램의 실행 시간 분석에 관한 연구 또한 활발히 진행되어 왔다.

전통적인 실행 시간 분석 기법은 실시간성 분석을 위해 원시 프로그램 단위 요소의 흐름을 분석하는 흐름 분석과 하드웨어 영향 요소의 분석, 그리고 분석된 요소들의 실행 시간을 계산하는 단계로 나뉜다. 하드웨어 영향 요소에 대

본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었습니다.

한 연구는 캐시와 파이프라인에 대해 많은 연구가 진행되었었으며 Tree-Based Calculation, Path-Based Calculation 그리고 IPET-Based Calculation 기법을 이용하여 요소들의 실행 시간을 예측하였다. [2,3]

대부분의 연구는 원시 프로그램을 분석의 주요 대상으로 하고 있다. 프로그램의 컴파일 시간에 생성되는 중간 코드 또는 어셈블리 코드를 분석 대상으로 한다.

이러한 정적 분석 기법들은 몇 가지 문제점을 지니고 있다. 원시 프로그램 상에 반복문이 포함되어 있는 경우에 설계자 또는 사용자가 직접적으로 반복 파라미터를 입력하여야 한다. 그리고 반복 파라미터의 예측이 힘들다. 따라서 분석에 추가적인 시간 및 비용이 필요하고 분석 결과의 정확도를 보장하지 못하는 문제점을 가지고 있다. 그리고 분산 실시간 시스템에 적용은 운영체제 동작에 따르는 부하를 고려하지 않음으로 인해 정확하고 신뢰할 수 있는 실행 시간 분석을 보장하지 못한다.

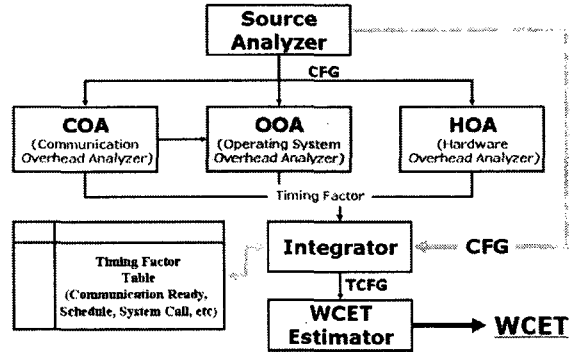
반복 파라미터의 직접 입력 문제를 해결하기 위해 분석 과정에서 표현식을 이용하여 실행 시간을 계산하는 방법이 연구되었다. [4, 5]

작업을 실행한 경우 운영체제에서 바로 실행되지 않고 작업들은 우선순위를 배정받고 큐에 저장된다. 이후 우선순위에 따라 큐에서 대기하고 프로세서에 다른 작업이 실행되지 않는 상태가 될 때 작업이 실제 실행된다. 일반적으로 리눅스 커널에서의 스케줄링은 우선순위에 따르고 우선순위가 같은 경우 라운드 로빈 방식으로 프로세서의 할당 시간을 공유한다.

스케줄러의 성능 향상 및 멀티 시스템에서의 효율적인 자원 할당과 같은 문제를 해결하기 위해 과거의 작업들의 특성을 고려한 작업의 큐 대기 시간을 예측하는 많은 연구가 진행되었다. 작업의 큐 대기시간 예측은 기존에 연구되어 왔던 실시간 예측 기법을 사용한다. 실시간 예측은 이전에 실행된 유사한 작업들의 과거 기록을 통해 현 작업의 큐 대기 시간을 예측하는 방법이다. 과거의 작업들의 특징을 몇 가지 요소로 구분하여 저장하고 현 작업의 특징을 추출하여 과거 작업들과의 유사도를 계산하여 유사한 몇 개의 작업을 선정하고 그 작업들의 실제 큐 대기 시간의 평균을 통해 현 작업의 큐 대기 시간을 예측한다. 큐 대기 시간을 예측하는 방법으로 평균 이외에 선형 회귀 방정식을 사용하기도 한다. 다른 기존에 연구된 기법들은 큐 대기 시간의 정확한 예측을 위해 많은 과거 정보를 가지고 있어야 하는 단점을 지닌다. 이러한 단점을 해결할 수 있는 새로운 실시간 예측 기법의 모색이 필요하다. [7, 8]

3. 실시간성 분석 도구

[그림 1]은 분산 실시간 시스템을 위한 실시간성 분석 도구의 전체적인 구조를 나타낸다. 소스 분석기는 컴파일러의 진단부에 해당하는 작업을 진행하여 CFG(Control Flow Graph)를 생성한다. CFG는 원시 프로그램의 동작 의미를 가지는 문장별 흐름의 정보를 표현한다. 소스 분석기의 추가적인 동작은 통신 부하 분석기 및 운영체제 부하 분석기에 사용될 통신, I/O 그리고 메모리 할당에 해당하는 문장 및 시스템 콜을 분석하여 저장한다. 파악된 문장별 흐름을 그래프 형태로 나타내어 영향 요소와 관련되어 저장



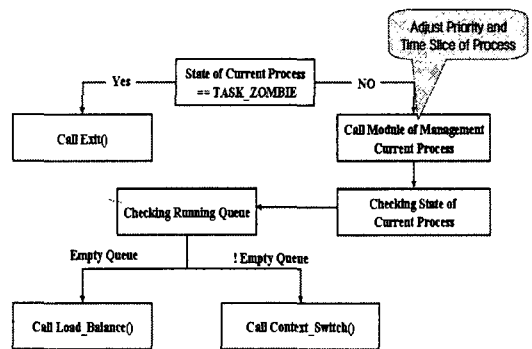
[그림 1] 실시간성 분석 도구

된 정보와 함께 다음 단계 분석기의 입력으로 사용한다. 통신, 운영체제 그리고 하드웨어 영향 요소 분석기는 원시 프로그램에서 추출된 영향 요소들의 시간 영향 요소(Timing Factor)를 분석하여 TCFG (Timing Control Flow Graph)를 생성한다. 본 연구에서 실행 시간 계산의 단계는 관련 연구에서 언급한 Tree-Based Calculation을 기반으로 하여 실행 시간을 예측한다.

통신 부하 요소 분석(COA : Communication Overhead Analysis)는 어플리케이션과 커널에서 통신을 위한 준비 시간의 분석을 선행하여 선행 예측 테이블을 작성한다. 선행 예측 테이블은 통신에 필요한 모듈 호출의 시작을 위한 바인딩 및 소켓 생성 네트워크 모듈의 최장 인스트럭션 경로와 각각의 실행 시간을 담고 있다. [1]

4. 운영체제에 의한 영향 요소

작업 실행 시 운영체제에 의해 발생하는 영향 요소는 스케줄링 기법에 의존적이다. 리눅스 커널을 대상으로 영향 요소를 분석하는 경우 우선순위와 할당 받은 프로세서를 사용할 수 있는 시간 단위(Time Slice)의 크기에 따라 작업의 실행시간이 달라진다. 그리고 또 다른 중요한 영향 요소는 큐 대기 시간이다.



[그림 2] 스케줄링 모듈 작업 처리 순서

[그림 2]는 스케줄링 모듈의 대략적 동작 과정을 보여준다. 현재 실행되고 있는 작업이 할당 받은 Time Slice를 소

비한 후 작업의 상태와 작업 큐를 확인하고 작업 상태가 종료된 상태가 아니면 우선순위와 Time Slice를 조정한다. 그리고 실행 큐에 다른 작업 있는지를 확인하여 다음 실행할 작업과 Context Switch를 실행한다. 따라서 스케줄링 모듈의 작업 실행에 필요한 최소한의 시간은 큐를 관리하기 위한 모듈의 실행 시간과 Context Switch 실행 시간이 추가되어야 한다.

[표 1] 스케줄 모듈 분석 결과

| Suboperation | Timing Factor (clock) |
|------------------|---|
| Schedule | 1962+(Queue_Handle+Context Switch+recalc_task_prio) |
| Queue_Handle | enqueue_task + dequeue_task |
| Context_Swith | 137 + (switch_mm + switch_to) |
| recalc_task_prio | 355 + effective_prio |
| enqueue | 84 + list_add_tail |
| dequeue | 53 + list_del |
| switch_mm | 209 + clear_bit + set_bit |
| switch_to | 19 |
| effective_prio | 78 |
| list_add_tail | 40 |
| list_del | 38 |
| clear_bit | 18 |
| set_bit | 18 |

[표 1]은 리눅스 커널 2.6.0 버전의 스케줄링 모듈의 실행시간을 정량적 정적 분석 방법을 통해 분석한 결과를 보여준다. 정량적 정적 분석 방법은 모듈 함수의 어셈블리 코드를 이용하여 각 인스트럭션의 흐름을 그래프 형태(CFG: Control Flow Grape)로 표현하고 시작점에서 종료점까지 최장 인스트럭션 경로를 추출한다. 분석된 스케줄링 모듈을 살펴보면 현재 작업의 할당된 Time Slice를 다 사용하거나 종료가 되면 Schedule 모듈이 호출되고 큐에 남아 있는 작업이 있는지 여부를 확인한 후(Queue_Handle) 대기 중인 작업과 교환 작업(Context Switch)을 시행하게 된다. 각 하위 모듈의 시간 요소는 최소한의 최악 Clock 수를 앞선 스케줄링 모듈 분석 방법을 이용하여 구한다.

그리고 작업의 실행시간에 운영체제 영향의 정도를 예측하기 위해서는 작업이 큐에 얼마나 대기하여야 하는 것도 중요한 분석 요소 중 하나이다. 앞서 설명과 같이 기존의 큐 대기 시간 예측 방법은 과거의 작업 정보를 저장하여야 하는 문제와 과거의 작업들과 현 작업과의 유사도를 검사하는 부하를 가지고 있는 단점이 있다. 이에 본 논문에서는 확률적인 큐 대기시간 예측 방법을 제안하고자 한다.

과거의 연구 방법들은 일반적인 분산 시스템을 대상으로 하고 있다. 분산 실시간 시스템으로 환경이 확장되면 작업의 실행에 가장 큰 영향을 미치는 요소는 우선순위이다. 즉, 우선순위가 높으면 많은 Time Slice를 할당 받게 되고

또한 다른 작업들을 선점할 수 있는 권한을 가지게 되므로 우선순위에 따른 큐 대기 시간을 확률적으로 예측 가능하다. 만약 과거의 정보를 최소한 사용하고 확률적 예측을 통해 기존의 큐 대기 시간을 예측하는 연구들 큐 대기 시간을 예측하기 위해 필요한 부하를 감소시킬 수 있을 것이다.

스케줄러에서 발생할 수 있는 영향 요소와 실시간 예측을 통해 얻어지는 큐 대기 시간을 작업의 실행시간에 반영 신뢰성 있는 실시간성 분석 도구 개발이 가능하다.

5. 결론 및 향후 연구

본 논문에서는 분산 실시간 시스템을 위한 실시간성 분석 도구의 설계에 있어 운영체제 의해 발생하는 부하를 분석하기 위한 모델 설계를 논하였다. 특히 작업의 실행에 가장 큰 영향을 미치는 운영체제의 요소인 스케줄러를 중심으로 영향 요소를 분석하였고 작업 큐 대기 시간 예측을 위한 확률적 접근 방법을 제안하였다.

특히, 작업 큐 대기 시간 예측을 위한 확률적 접근 방법에 대한 구체적인 실험을 통해 예측 오류율을 최소화할 수 있는 기법 설계 및 구현을 향후 연구과제로 남긴다.

분산 실시간 시스템에서 단순한 작업 자체의 실행시간 예측만을 고려하지 않고 주위 환경의 영향 요소인 통신, 운영체제 그리고 하드웨어 영향 요소를 추가하여 보다 정확하고 신뢰성 있는 분산 실시간 분석 도구 개발을 진행한다.

6. 참고문헌

- [1] Y. S. Hong, H.W. Goo, Moon H. Kim and C.H. Chang "Estimation of the communication time for distributed real-time applications in TMO-Linux", UKC2005, August, 2005
- [2] Ermedahl, Andreas, "A Modular Tool Architecture for Worst-Case Execution Time Analysis", Doctoral Thesis, Uppsala University, 2003
- [3] J. Engblom, A. Ermedahl, M. Sjoedin, J. Gubstafsson, and H. Hansson. "Worst-case execution-time analysis for embedded real-time systems", Journal of STTT, vol. 4, p. 437-455, no. 4. 2003
- [4] G. Bernat and A. Burns. "An approach to symbolic worst-case execution time analysis". In Proc. 25th Workshop on Real-Time Programming, Palma, Spain, May 2000.
- [5] B. Lisper. "Fully automatic, parametric worst-case execution time analysis". In Proc. of 3rd Int. Workshop on WCET Analysis, pages 99-102, July 2003.
- [6] K. H. (Kane) Kim, "APIs for Real-Time Distributed Object Programming, Computer", IEEE Computer, v.33 n.6, p.72-80, June 2000.
- [7] W. Smith, V. Taylor, and I. Foster, "Using Run-Time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance", In Proceedings of the IPPS/SPDP, pages 202-219, 1999