

컴포넌트의 비동기적 교체를 이용한 동적 재구성 기법

송인준^o 김영필 김경운 유혁

고려대학교 컴퓨터학과

{ijsong^o, ypkim, kykim, hxy}@os.korea.ac.kr

Dynamic Reconfigurable Mechanism using Asynchronous Replacement of Component

Injun Song^o Youngpill Kim Kyeongyun Kim Hyuck Yoo

Dept. of Computer Science and Engineering, Korea University

요약

최근 내장형 시스템의 보급과 발전으로 휴대용 기기에서 다양한 애플리케이션들을 이용하고 있다. 특히, 핸드폰과 스마트폰과 같은 정보 통신 기기들은 과거의 전화 기능 외에도 개인 정보 관리 및 게임 등 다양한 기능을 수행한다. 또한 차세대 무선 통신 기술로써 SDR(Software Defined Radio) 등의 연구가 이루어지고 있다. 이렇게 소형의 휴대 기기에서 다양한 기능과 서비스를 제공하기 위해서는 시스템의 구조가 동적으로 재구성 되어야 할 필요성이 증대되고 있다. SDR 시스템의 경우, 하나의 장치에서 두 개 이상의 신호 처리(Signal Processing) 기법이 소프트웨어적으로 제공되어야 한다. 이러한 유연성(Flexibility) 및 재구성(Reconfigurability)을 요건들을 만족시키는 시스템을 위해서는 운영체제 단계에서 동적인 재구성 기법을 제공해주어야 한다. 본 논문에서는 컴포넌트의 비동기적(Asynchronous) 교체를 이용하여 기존의 동적 재구성 기법의 지연 시간을 줄이는 방법을 설명하며 실험을 통해 그것을 증명하도록 한다.

1. 서론

휴대폰이나 PDA와 같은 내장형 기기들의 이용률이 급증하면서, 그 용도의 범위 또한 넓어지고 있다. 특히 소형의 휴대 장치를 이용하여 전화, 개인 정보 관리, 문서 작업, 게임, 비디오 플레이 등의 다양한 용도로 활용하고 있다. 하지만 내장형 장치는 프로세서(Processor)의 능력과 주기억 장치 및 보조 기억 장치의 용량이 일반 PC 보다 작다. 이러한 하드웨어 자원 부족으로 인하여 다양한 서비스를 제공하는데 문제가 발생하고 있다.

제한된 하드웨어 자원을 이용하여 다양한 서비스와 응용 프로그램을 제공하기 위해 동적 재구성 기법에 대한 연구가 진행되고 있다. 동적 재구성을 위한 운영체제의 연구는 K42 운영체제의 Online Reconfiguration 기법[1]과 MMLite 운영체제의 Mutation 기법[2] 및 Kea 운영체제의 동적 재구성 기법[3] 등이 있다. 특히, SDR 시스템[4]은 다양한 무선 통신 표준에 대한 신호 처리 소프트웨어를 시스템의 메모리에 적재해야 한다. 이때 시스템의 제한된 자원을 효율적으로 관리하기 위해 소프트웨어 컴포넌트의 동적 재구성 기법이 필요하게 된다. Vanu와 같은 회사는 HP iPAQ PDA에 신호 처리 소프트웨어의 동적 재구성 기법을 제공하여 SDR 시스템을 구현하였다.[5] 이렇게 컴포넌트의 동적 재구성 기법은 시스템의 유연성(Flexibility)과 제한된 자원에 대한 효율성을 제공한다.

그러나 현재까지의 동적 재구성 기법은 재구성 단계에

서의 지연 시간이 길어지게 된다는 단점이 있다.[6] 이를 위해 동적 재구성 기법의 지연 시간을 줄이기 위한 방법이 필요하다.

본 논문에서는 동적 재구성 기법의 지연 시간을 줄이기 위한 기법을 제안하고자 한다. 이를 위해 비동기 방식의 컴포넌트 교체를 제안하고 그것을 실제로 실험을 통해 증명해본다.

논문의 구성은 다음과 같다. 먼저 2장에서는 간략한 배경 설명과 관련 연구에 대해 알아보고 3장에서는 제안된 동적 재구성 기법에 대해서 설명하고 이를 실험해 본다. 4장에서는 결론 및 향후 계획을 설명한다.

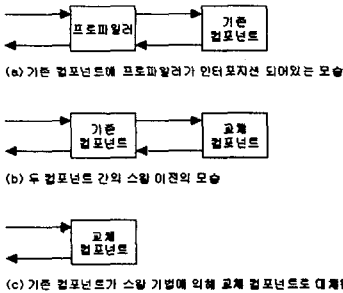
2. 배경 설명 및 관련 연구

컴포넌트의 재구성 기법은 크게 정적인 재구성 기법과 동적인 재구성 기법으로 나뉜다. 시스템의 실행 중에 그 행동을 변화시키기 위해서는 동적인 재구성 기법이 필요하며, 본 논문에서는 동적인 재구성 기법의 재구성 시의 지연 시간을 줄이는 기법을 제안하고 구현을 위한 기본 플랫폼으로 M3K를 사용하였다.

2.1 운영체제 컴포넌트의 동적 재구성 기법

운영체제의 유연성을 더하고 재구성 가능한 환경을 주는 기법에는 여러 가지 연구가 있다. [그림1]에서 보듯이, 동적 재구성의 기법에는 인터포지션(Interposition) 기법과 스왑(Swap) 기법이 있다.[1] 이 두 가지 기법은 각 컴포넌트의 상태가 활성화 되지 않은 상태를 가정하

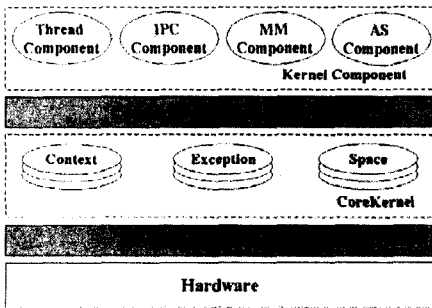
고, 인터포지션과 스왑 기법을 실제로 행하기 위한 작업이 복잡하다는 단점이 있다.



[그림 1] 동적 재구성성을 위한 (a)인터포지션 기법과 (b),(c)스왑 기법

2.2 M3K 커널의 개요

M3K는 멀티미디어에서 요구하는 실시간 특성을 지원할 수 있는 것을 목표로 제작된 커널로, [그림2]와 같은 구조를 갖는다. 하드웨어를 추상화 하는 문맥(Context), 주소(Space), 예외(Exception) 컴포넌트로 구성된 코어커널(CoreKernel)과 이들이 제공하는 인터페이스를 통해 커널 서비스를 제공하는 쓰레드(Thread), IPC, 메모리 관리(MM), 주소 공간(AS) 컴포넌트로 구성된 커널 컴포넌트(Kernel Component)가 있다.[7][8][9]



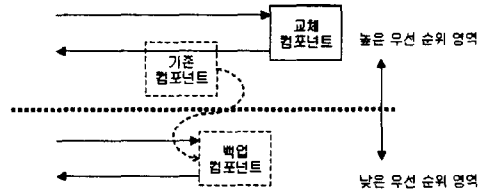
[그림 2] M3K 커널의 구조

3. 컴포넌트의 비동기적 교체

위에서 언급한 동적 재구성 기법은 기능을 추가하거나 구성 요소를 교체하는 과정이 복잡하고 교체 연산 전의 준비단계가 길다는 단점이 있다. 위의 스왑 기법은 포워딩(Forwarding), 블록(Block), 전승(Transition) 등 세 단계가 있다. 만약 블록 상태에서 활성화중인 쓰레드의 처리 시간이 길어진다면 스왑 기법의 요청은 늦어지고, 교체될 컴포넌트의 서비스를 이용하기 위한 지연 시간이 길어진다. 이러한 기존의 동적 재구성 기법은 교체될 컴포넌트의 빠른 응답성을 요구하는 시스템에서는 적합하지 않게 된다.

3.1 비동기적 교체

본 논문에서는 교체될 새로운 컴포넌트의 교체 시간을 빠르게 하기 위해 블록 상태의 지연을 줄이는 재구성 기법을 제안한다. 이 기법은 [그림3]에서 보듯이, 스왑 연산이 시작되면 백업 컴포넌트로 수행되고, 이 백업 컴포넌트는 낮은 우선순위를 갖고 수행된다.

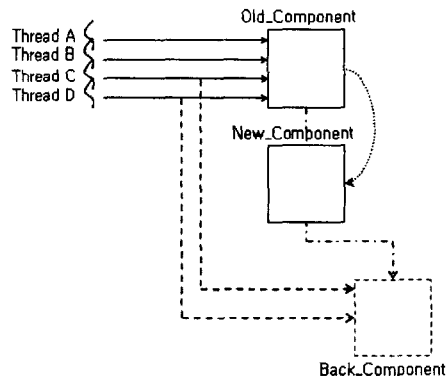


[그림 3] 권장화된 재구성 기법

3.2 실험

본 논문에서 제안한 컴포넌트의 비동기적 교체를 이용한 동적 재구성 기법의 성능을 측정하고, 기존의 방식의 결과와의 비교를 하였다. 실험에 사용된 하드웨어는 인텔 펜티엄 2.4GHz 프로세서와 512MB의 메모리를 장착하고 있다.

각 실험은 100회 반복 측정 후 평균값을 구한 것이다. [그림4]와 같이 실험에 사용된 컴포넌트는 교체 컴포넌트(New_Component)와 기존 컴포넌트(Old_Component), 백업 컴포넌트(Back_Component)이며, 기존 컴포넌트에 연산을 수행을 시도하는 쓰레드는 4개(Thread A - D)이다. 쓰레드의 수행 시간 및 지연 시간에 대한 측정은 프로세서의 TSC(Time Stamp Counter)를 이용하였다.



[그림 4] 측정 컴포넌트 모델

[표1]은 각 쓰레드들의 수행시간을 측정 한 값이다. 이 값들은 재구성 기법을 사용하지 않고 기존 컴포넌트에 연산을 수행시켰을 때의 수행시간의 측정값이다. 각 컴

포넌트의 수행 시간은 A<B<C<D 순서이다.

| 쓰레드 | A | B | C | D |
|------------|---------|-----------|-----------|-----------|
| 수행시간 (TSC) | 938,676 | 1,644,416 | 3,649,220 | 7,191,060 |

[표 1] 각 쓰레드들의 수행시간 측정

교체 컴포넌트는 기존 컴포넌트에 각 쓰레드들이 연산을 수행하는 중에 기존 컴포넌트와 동적으로 스왑 연산을 수행한다. [표2]는 교체 컴포넌트 응답시간이 기존의 포워딩, 블록, 전송 단계를 거치는 방식에서의 교체 컴포넌트의 응답시간과 본 논문에서 제안한 비동기 교체를 이용한 재구성 기법에서의 응답시간을 측정된 결과이다.

| 수행 방법 | 기존의 재구성 | 비동기 교체를 이용한 재구성 |
|------------|---------|-----------------|
| 응답시간 (TSC) | 640,996 | 285,612 |

[표 2] 교체 컴포넌트의 응답시간 측정

기존의 재구성 기법의 경우, 교체 컴포넌트의 서비스가 호출된 시점부터 서비스를 끝낼 때까지의 응답시간의 평균값이 비동기 교체를 이용한 재구성 기법의 교체 컴포넌트의 응답시간의 평균값에 비해 오래 걸린다는 것을 알 수 있다. 측정 결과에서 볼 수 있듯이, 본 논문에서 제안한 컴포넌트의 비동기 교체를 이용한 재구성 기법의 경우가 기존의 기법에 비해 응답시간을 약 44% 감소시켰음을 볼 수 있다.

3. 결론 및 향후 계획

본 논문에서는 기존의 동적 재구성 기법에서 교체 연산 전의 준비 단계가 길고 교체 과정이 복잡하다는 단점을 해결하기 위하여 컴포넌트의 비동기적 교체를 이용한 동적 재구성 기법을 제안하였다. 본 논문이 제안하는 기법은 기존의 동적 재구성 기법의 포워딩, 블록, 전송 단계 중, 활성화 중인 쓰레드의 종료를 위한 블록 단계의 지연 시간을 줄임으로써 교체 컴포넌트의 응답시간을 줄였다. 하지만 백업 컴포넌트로 인한 메모리 사용량의 증가라는 단점을 가지고 있다.

향후, 본 연구 내용을 한층 발전시켜서, 메모리 사용량을 줄이고자 하는 연구와 SDR 시스템에 적용에 대한 연구를 진행할 계획이다.

참고 문헌

- [1] C. A. N. Soules, J. Appavoo, K. Hui, et al, "System Support for Online Reconfiguration", *USENIX 2003 Annual Technical Conference*
- [2] Johannes Helander, Alessandro Forin, "MMLite: A Highly Componentized System Architecture", *Eight*

- ACM SIGOPS European Workshop, September 1998*
- [3] Alistair Craig Veitch, "A Dynamically Reconfigurable and Extensible Operating System", Thesis of Ph.D. University of Columbia, July 1998
- [4] SDR Forum <http://www.sdrforum.org/>
- [5] J. Chapin, V. Bose, "The Vanu Software Radio System", 2002 Software Defined Radio Technical Conference, *San Diego, November 2002*
- [6] 송인준, 김영필, 유혁, "휴대 기기에서의 SDR을 위한 운영체제 구조와 재구성 기법", *추계 정보과학회 논문집 2003*
- [7] 김영호, 고영웅, 이재용, 유혁, "멀티미디어 마이크로 커널 M3K에서 프로세스간 통신 구현 및 성능 분석", *정보과학회 논문지, 2002*
- [8] 양순섭, 고영웅, 조유근, 신현식, 최진영, 유혁, "컴포넌트 기반 커널을 위한 프레임워크", *춘계 정보과학회 학술대회 논문집, 1999*
- [9] 김영호, 고영웅, 유혁, "M3K에서 IPC 컴포넌트 설계 및 구현", *추계 정보과학회 학술대회 논문집, 2000*