

스마트 TV환경에 적합한 Web Worker 기반의 그리드 컴퓨팅 시스템

김 현 식*, 조 근 식*

Grid Computing System based on Web Worker for Smart TV Environments

Hyunsik Kim*, Geunsik Jo*

요 약

스마트TV의 보급이 확대되면서 스마트TV의 여유 연산능력을 활용한 그리드 컴퓨팅 시스템에 대한 수요가 증가할 것으로 예측하고 있다. 그러나 스마트TV는 PC와 달리 다양한 플랫폼이 존재하고, 또한 사용자들의 스마트 TV 사용 패턴이 기존 PC 사용 패턴과 크게 다르기 때문에 PC기반의 그리드 시스템을 적용하기에는 어려움이 있다. 즉, 유휴 상태에서 동작하던 기존 그리드 에이전트가 스마트TV 환경에서는 동작할 기회를 보장 받을 수 없는 상황이 된 것이다. 본 논문에서는 이러한 문제를 극복하기 위하여 스마트TV에 적합한 자바스크립트 기반의 그리드 시스템 및 콘텐츠 시청 중에 남는 CPU자원을 활용하는 에이전트 스케줄링 정책을 제안하고 구현하였다. 그리고 기존 그리드 방식과 제안하는 방식을 비교 실험하여, 그 결과 기존의 그리드가 0.09%, 제안하는 방식은 평균 1.78%의 자원을 제공함을 보였다.

▶ Keyword : 그리드 컴퓨팅, 스마트TV, 자바스크립트, 웹 워커

Abstract

In recent years, there has been a popularity rose up on Smart TV (Smart Television) usage at home. Therefore, it is also have increase the demand on grid computing system. Smart TV has a variety of platform and usage compare to PC (Personal computer). Base on this, it is difficult to apply a traditional grid system on Smart TV. One major reason are concerning the small idle time compare to PC. To overcome this problem, this paper will propose a Javascript grid system and introducing a new scheduling policy that best suit for a smart TV. We have conduct an experiment

• 제1저자 : 김현식 • 교신저자 : 김현식

• 투고일 : 2011. 09. 28, 심사일 : 2011. 10. 05, 게재확정일 : 2011. 10. 13.

* 인하대학교 컴퓨터정보공학과(Dept. of Computer and Information Engineering, Inha University)

on the proposed method. The result provides an average of 1.78 percent, which is improved compare to the traditional method which is only provides an average of 0.09 percent.

▶ Keyword : Grid Computing, Smart TV, JavaScript, Web Worker

I. 서론

DisplaySearch의 발표 자료에 따르면 2014년까지 전 세계적으로 약 1억 2천만대 가량의 스마트TV가 각 가정에 보급될 것으로 예측하고 있다[1].

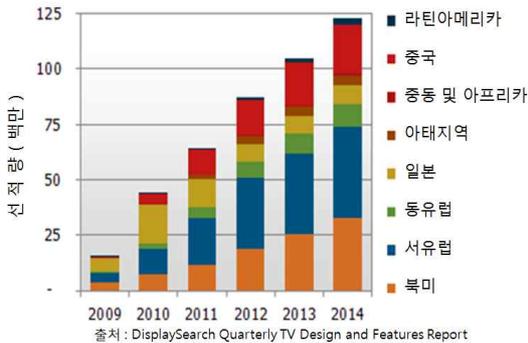


그림 1 스마트TV 보급 전망
Fig. 1. Smart TV shipment

스마트TV는 기존 TV와 달리 별도의 프로세서를 내장하고 있는 임베디드 컴퓨터로서 그 내부 구조는 일반 TV보다는 PC에 더욱 가깝다. 스마트TV의 프로세서는 콘텐츠를 시청하는데 충분한 성능을 갖고 있으며 항상 일정한 범위의 여유 연산능력을 남겨놓고 있다. 이 여유 연산능력을 그리드 컴퓨팅에 활용할 수 있다면 거대한 연산 자원의 확보가 가능할 것이다.

그러나 스마트TV를 이용한 그리드 컴퓨팅을 적용하기 위해서는 크게 두 가지 요소를 고려해야한다[2].

첫째, 스마트TV는 PC와 달리 하나의 단일 아키텍처로 제작되지 않는다. 제조사별로 상이한 CPU를 사용하며 따라서 PC에서와 같이 애플리케이션 방식의 그리드 에이전트를 구현하고자 할 경우 다양한 플랫폼간의 포팅 문제가 대두된다.

둘째, 스마트TV는 PC와 사용 패턴이 다르다. 모든 스마트TV에 그리드 에이전트를 포팅하는 데 성공했다 하더라도 스마트TV는 시청이 끝나면 전원을 끄기 때문에 유휴 시간 (Idle time)에 동작하도록 설계된 그리드 에이전트는 동작할 기회를 보장 받을 수 없는 상황이 된다.

본 논문에서는 위에서 언급한 고려사항들에 대해서 웹 브라우저의 자바스크립트를 이용하여 그리드 에이전트를 구현

하고 스마트TV에 적합한 그리드 에이전트 스케줄링을 제안함으로써 호환성 문제와 그리드 에이전트의 동작 기회 보장 문제를 해결할 수 있다. 따라서 본 논문의 목적은 다음과 같다.

첫째, 자바스크립트 기반의 그리드 에이전트 시스템을 제안하고 구현하여 플랫폼 독립성을 확보한다. 둘째, 그리드 에이전트의 스케줄링 정책을 유휴 시간에 동작하는 방식에서 콘텐츠 소비 중 남은 CPU자원을 활용하는 방식으로 변화시켜 PC와 상이한 사용 패턴을 보이는 스마트TV상에서 최대의 작동 기회를 끌어낸다.

본 논문의 구성은 2장에서는 그리드 컴퓨팅에 대해 소개하고 PC기반의 그리드 환경 중 하나인 SETI@HOME에 대해 소개한다. 그리고 그것이 스마트TV환경으로 옮겨갈 때 발생할 수 있는 문제점과 병렬 자바스크립트 모델 및 Web Worker에 대해서 소개한다. 3장에서는 스마트TV환경에 맞는 Web Worker기반의 분산 컴퓨팅 그리드 시스템을 제안하고 4장에서 실험을 통해 기존 그리드 시스템과 비교하고 5장에서 결론을 제시한다.

II. 관련 연구

2.1. 그리드 컴퓨팅의 정의

그리드 컴퓨팅(Grid Computing)은 지리적으로 분산된 네트워크 환경에 수많은 이기종 컴퓨터와 대용량 저장장치, 데이터베이스 시스템, 인공위성 등과 같은 다양한 자원들을 고속 네트워크로 연결하여 그 자원들을 상호 공유할 수 있도록 하는 차세대 컴퓨팅 환경이다[3]. 이용하는 자원에는 CPU(Computing Grid), 스토리지(Data Grid), 네트워크(Access Grid) 등이 있으며 본 논문에서 언급하는 그리드 컴퓨팅은 이 중 CPU자원을 이용하는 컴퓨팅 그리드를 말한다.

2.2. 그리드 컴퓨팅 분류

그리드 컴퓨팅을 달성하기 위한 방법에는 크게 Cluster Computing과 Volunteer Computing이 있다[4,5]. 그 중 Cluster Computing은 Local Network상에 여러 대의 과제 수행 전용 컴퓨터를 갖추고 병렬 연산을 수행하는 것을 말하며 Volunteer Computing은 자원봉사자의 컴퓨터를 유휴

시간 동안에만 잠시 빌려 쓰는 형식을 취한다. Cluster Computing은 과제 수행의 신뢰도 및 가용성이 Volunteer Computing에 비해 비교적 높다는 장점이 있으나 하드웨어의 구입 및 유지에 더 많은 비용이 들어간다는 단점이 있다. 반면 Volunteer Computing은 Cluster Computing에 비해 비교적 다양한 하드웨어와 OS를 고려해야 하고 Network Latency가 비교적 높으며 신뢰성이 떨어진다는 단점이 있다.

2.3. 애플리케이션 기반의 그리드 컴퓨팅

David는 SETI@home 프로젝트를 통해 PC기반의 Volunteer Computing 모델을 따르는 그리드 컴퓨팅 시스템을 소개하였다[6]. 이 시스템은 화면 보호기(Screen Saver)의 부 프로그램으로 구현돼 있으며 사용자의 PC가 유휴 상태일 때에 작동하여 각종 과제를 수행하도록 설계돼 있었다.

그러나 이 시스템은 PC가 완전히 유휴 상태에 들어갔을 때에 깨어나 과제를 수행하는 구조이기 때문에 사용자가 비교적 가벼운 작업을 하는 경우 즉 CPU이용률이 매우 낮은 경우에도 작동하지 않는다는 한계를 갖는다. 또한 애플리케이션 기반이기 때문에 아키텍처나 OS가 달라질 경우 그에 맞춰 에이전트를 새로 포팅 해야 하는 문제가 존재한다[2].

2.4. 스마트TV환경

스마트TV는 TV와 PC의 결합이라고 할 수 있다. 따라서 스마트TV는 PC와 많은 것이 유사하지만 그 사용 목적이 PC와는 다르기 때문에 크게 두 가지 차이점이 존재한다.

첫째 스마트TV는 PC보다 더 다양한 하드웨어 아키텍처를 사용할 것이다. 현재 PC 아키텍처는 x86으로 모두 통일된 상태이다. 그러나 스마트TV는 아직 시장이 성숙돼 있지 않기 때문에 x86을 벗어나 ARM등 각종 최신 아키텍처를 적용한 하드웨어가 범람할 가능성이 높다. 이런 하드웨어 환경에서 기존 애플리케이션 방식의 그리드 에이전트를 사용할 경우 포팅에 많은 시간을 소비하게 될 것이다.

둘째 TV는 PC와는 사용 패턴이 다르다. TV는 시청이 끝나면 전원을 끄는 것이 일반적이며 이는 스마트TV에서도 마찬가지일 것이다. 즉 2.3에서 언급한 화면 보호기의 부 프로그램으로써 동작하는 그리드 에이전트는 스마트TV환경 하에서는 작동 기회가 보장 되지 않을 것임을 유추할 수 있다.

현재 스마트TV에서 웹 브라우저는 기본적으로 설치된 상태로 공급되고 있다. 대표적으로 구글의 크롬 브라우저[7], 그리고 애플TV의 사파리 브라우저[8]를 들 수 있다. 향후 스마트TV는 인터넷 스트리밍 데이터를 처리할 수 있어야 하기

때문에 웹 브라우저는 모든 스마트TV에 기본으로 장착될 것으로 판단된다.

2.5. 병렬 자바스크립트 프로그래밍 모델과 Web Worker

Adam Welc의 3명은 Generic Worker를 소개하면서 병렬 자바스크립트 프로그래밍 모델을 제시하였다[9]. 이것은 현재 Web Worker라는 기술로써 구현되어 최신 웹 브라우저에 탑재되고 있다[10]. 현재 스마트TV에 탑재되었거나 탑재될 예정인 대부분의 브라우저가 표준을 준수하고 있으므로 Web Worker 또한 대부분의 스마트TV에서 지원할 것이다.

Web Worker는 HTML5에 도입된 새로운 멀티스레드 자바스크립트 기술이다. 이 기술을 사용하면 기존에는 싱글 스레드만 가능했던 웹 애플리케이션을 멀티스레드로 동작시킬 수 있으며 플랫폼 독립적인 자바스크립트의 특성상 다양한 하드웨어 및 브라우저에서 수정 없이 동작시킬 수 있다[11].

Shusuke Okamoto는 Web-based MMORPG에서 발생하는 서버의 workload를 분산시키기 위해 Web Worker를 사용했다[11]. 이 논문에서 Shusuke는 Web Worker를 사용하여 메인 콘텐츠를 방해하지 않고 백그라운드에서 각종 처리를 담당해 서버의 부담을 줄이는 방법을 제안하였다.

III. Web Worker 기반의 그리드 컴퓨팅 시스템

현재의 대다수 스마트TV는 인터넷 스트리밍 서비스에 크게 의존하기 때문에 웹 브라우저가 필수적으로 포팅돼 있으며 이들 웹 브라우저에는 표준을 준수하는 자바스크립트 엔진이 탑재돼 있으므로 웹 브라우저의 상위 계층에서 동작하는 자바스크립트 엔진은 ECMA-262 표준을 준수하며 Web Worker를 지원하는 것을 가정할 수 있다[10,13].

본 논문에서는 스마트TV에 포팅된 웹 브라우저가 제공하는 Web Worker에 기반한 플랫폼 독립적인 그리드 에이전트를 제안한다[12]. 제안하는 그리드 에이전트 시스템은 그림 2와 같은 계층 구조를 가지며 이 중 웹 브라우저의 자바스크립트 엔진은 그 상위 단계의 Grid Agent에 플랫폼 독립성을 제공한다.



그림 2. Web Worker 그리드 에이전트의 구현 계층도
Fig. 2. System Hierarchy for Web Worker Grid Agent

전체적인 그리드 시스템은 그림 3에 보이는 바와 같이 크게 서버 사이드(Server side)와 클라이언트 사이드(Client Side)로 나뉜다.

그림 3의 왼쪽에 있는 서버 사이드에서는 Map Reduce Framework가 그리드에 참여한 컴퓨터(여기서는 스마트 TV)에 적절한 Job을 분배하고 결과를 취합하는 역할을 맡는다. 그림 3의 오른쪽에 있는 클라이언트 사이드에서는 Web Worker로 구현된 그리드 에이전트가 서버로부터 수신한 Job을 백그라운드에서 처리해 결과를 서버로 전송한다. 이 과정은 클라이언트가 서버와의 연결을 종료할 때까지 즉 하나의 콘텐츠(즉 동영상)의 소비를 마칠 때까지 반복되며 소비를 마친 이후 프로그라운드 태스크(Foreground Task)와 백그라운드 태스크(Background Task)가 동시에 종료되면서 자원을 반납한다.

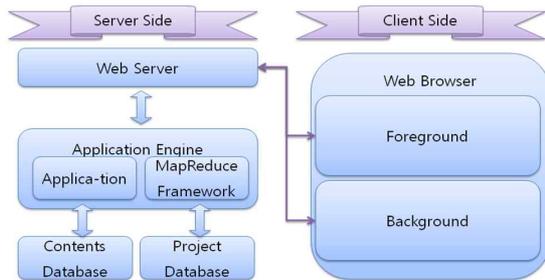


그림 3. 아키텍처 디자인
Fig. 3. Architecture Design

이상의 과정을 그림 4에 Flow Chart형식으로 표시하였다. TV사용자가 보고 싶은 VOD콘텐츠를 선택하면 서버에서는 해당 콘텐츠를 프로그라운드 태스크로 보내면서 동시에 Web Worker 그리드 에이전트를 전송한다. 클라이언트는 전달받은 그리드 에이전트를 백그라운드 태스크로 실행시킨다. 백그라운드 태스크에서 실행중인 에이전트(그림 4의 오른

쪽)는 현재 여유 CPU연산능력을 측정하고 그 결과를 서버의 MapReduce Framework에 통지한다.

통지를 받은 MapReduce Framework는 현재 전송중인 VOD콘텐츠의 남은 재생시간 및 클라이언트의 연산 능력을 토대로 가장 적절한 Job을 클라이언트로 전달한다. 여기서 가장 적절한 Job이란 클라이언트가 VOD재생을 끝내기 이전에 완료할 수 있는 Job중 가장 연산 시간이 오래 걸리는 Job이다.

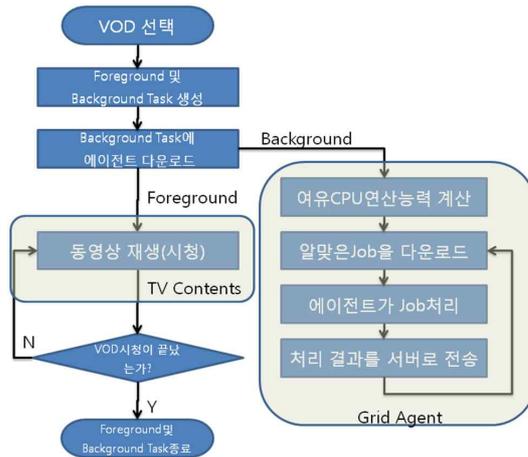


그림 4. 작업 흐름
Fig. 4. Workflow

Job은 그림 5와 같은 구조로 이루어져 있다. 그림 5-1의 Subject ID는 과제의 고유번호이며 결과를 서버로 통보할 때 이 ID를 응답 패킷의 Subject ID에 적는다. 데드라인(Deadline)은 이 과제를 완료할 때까지 허용된 시간으로 이 시간을 초과하여 계산을 수행하는 과제는 에이전트가 조용히 파기한다. 프로그램 코드(Program Code)는 함께 전송되는 데이터를 계산하기 위한 자바스크립트 코드이다. 이 패킷을 클라이언트에서 동작중인 Web Worker 그리드 에이전트가 받아서 수행한다.

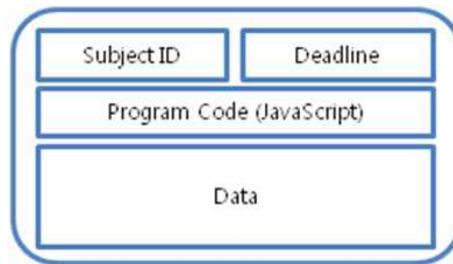


그림 5. Job Packet의 구조
Fig. 5. Job Packet

처리가 완료되면 그림 6과 같은 구조의 응답 패킷을 서버의 MapReduce Framework로 전송한다. Subject ID는 그림 5의 Subject ID와 같은 값이며 경과 시간(Elapsed time)은 과제를 수행하는데 실제로 걸린 시간이다. 데이터(Data)는 연산 결과값을 담고 있다.



그림 6. 응답 Packet의 구조
Fig. 6. Job Response Packet

만약 과제가 완료되었음에도 VOD콘텐츠가 재생 중이면 에이전트는 MapReduce Framework에 현재 여유 CPU연산능력을 통지함으로써 위 과정을 반복한다.

IV. 실험

스마트TV환경을 시뮬레이션 하기 위해 일반 PC에서 YouTube동영상을 시청 중 Web Worker 그리드 에이전트를 작동시켜 CPU사용량을 측정하였다.

먼저 에이전트를 작동시키지 않은 상태에서 순수 동영상 시청 중의 CPU사용량을 측정하였으며 실험 결과는 아래와 같다.

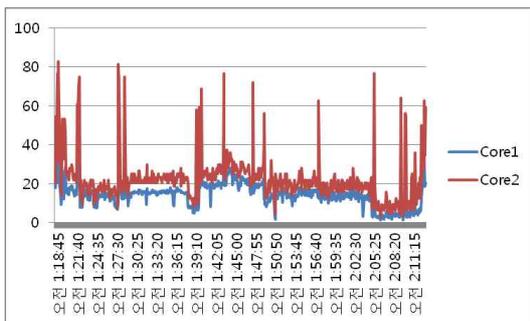


그림 7. 동영상 재생시의 CPU사용량
Fig. 7. CPU usage while video playing only

그림 7의 세로축은 CPU사용량을 백분율로 나타낸 것이다. 동영상 재생 중 CPU사용량은 약 20%에서 일정하였다. 스마트TV에서도 동영상 재생 중 CPU사용량은 위와 유사한

그래프를 그릴 것으로 예상된다.

다음 에이전트를 작동시킨 상태에서의 CPU사용량을 측정하였다. 이때 에이전트가 동영상의 재생을 방해하여서는 안 된다. 따라서 실험 중 동영상이 끊기는 경우가 있는지 확인하였다.

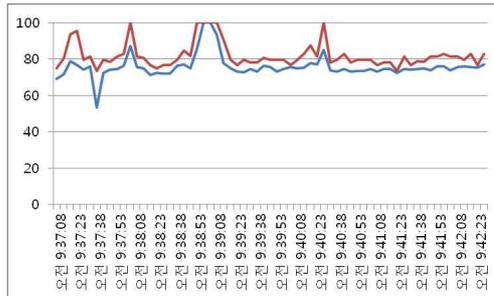


그림 8. 동영상 변환 응용과제의 CPU사용량
Fig. 8. CPU usage of video translation job while another video playing

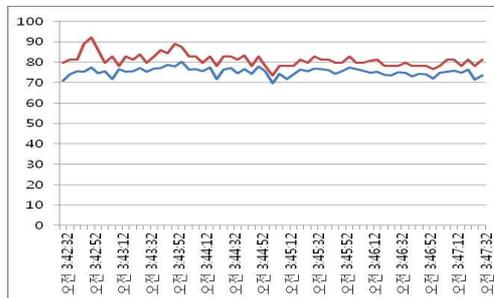


그림 9. WordCount 응용과제의 CPU사용량
Fig. 9. CPU usage of WordCount job while video playing

그림 8과 9는 각기 특성이 다른 두 가지 그리드 응용과제를 수행한 결과 그래프이다. 각각의 선은 듀얼 코어 CPU에서 각 코어의 사용량을 나타낸다.

그림 8은 HD급 동영상을 SD급 동영상으로 변환하는 애플리케이션의 수행 결과다. 이 애플리케이션은 입력되는 데이터에 대해 일정한 CPU자원을 소모하며 Job의 종료 시점을 정확히 예측 가능한 특성이 있다. 그림 9는 한 문서에 나타나는 모든 단어의 빈도수를 계산하는 애플리케이션의 수행 결과다. 이 애플리케이션은 입력되는 데이터에 따라 CPU자원 소모율이 달라지며 Job이 언제 끝날지 예측하기 어렵다.

두 애플리케이션 모두 약 80%정도에서 일정한 CPU이용률을 보였다. 또한 그림 8에서 CPU이용률이 100%가 되는 구간이 다소 존재하나 동영상은 끊김 없이 재생되었다.

이상의 실험 결과를 바탕으로 동영상 재생 중 Web

Worker 그리드 에이전트는 해당 PC에 한해 약 60%의 CPU자원을 일정하게 사용함을 알 수 있다. 실험의 공정성을 위해 사양이 서로 다른 4개의 PC에서 동일한 실험을 수행하였으나 모두 위 그래프와 유사한 결과를 보였다.

다음, 스마트TV의 시청 패턴을 분석하여 기존 PC방식의 그리드 시스템과 제안하는 Web Worker Grid 시스템간의 자원 제공 효율성을 비교하였다.

TNS미디어코리아[14]의 시청률 조사 자료에 따르면 전체 가구 중 0.4%가 약 5시간 30분간 TV를 방송신호 없음 상태(즉 Idle 상태)로 놔두는 것으로 나타났다. 기존 그리드 시스템이 이 시간 동안 CPU를 100% 이용할 수 있다고 가정하고 전체 스마트TV의 CPU이용률을 아래의 수식을 통해 산출했다

$$R = \frac{Mw}{Ma} \times D \times U \dots\dots\dots (1)$$

- R: 전체 스마트TV에 대한 CPU이용률(%)
- Mw: 그리드 에이전트가 가동중인 TV의 수
- Ma: 전체 스마트 TV의 수
- D: 하루 중 가동시간 비율(duty ratio)
- U: 그리드 에이전트의 CPU이용률(%)

계산 결과 기존 그리드 에이전트를 스케줄 정책 변화 없이 그대로 사용했을 때 0.09%의 CPU이용률만을 가질 수 있었다.

반면 제안하는 Web Worker 그리드 에이전트는 Idle시간을 이용하지 않고 콘텐츠를 소비하는 시간 동안 CPU의 여유연산 자원을 활용하는 구조로 돼 있다.

TNS미디어코리아[14]의 주간 시청률 Top20을 평균한 결과 약 15.1%의 시청자가 동시에 TV를 시청하는 것으로 조사되었다. 이는 지상파TV의 20개 프로그램에 대한 평균 시청률이며 케이블TV 및 VOD시청자는 반영돼 있지 않다. 따라서 실제 동시 시청자수는 이보다 많을 것으로 예상할 수 있다.

또한 가구별 시청 패턴 조사 리포트에 따르면 한 가정에서 평균 약 6시간 동안 TV를 시청하는 것으로 조사되었다[15]. 이 시간 중 1시간가량을 HD콘텐츠(영화/드라마)를 소비하는데 사용하는데 스마트TV의 CPU의 성능이 PC보다 낮은 것을 감안하여 HD콘텐츠를 시청하는 중에는 그리드 에이전트가 작동할 만한 여유가 없다고 가정하였다.

상기한 데이터를 통해 예상되는 자원 이용률을 실험을 통해 파악하여 아래의 결과를 얻었다.

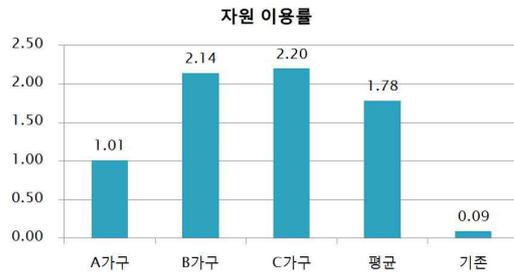


그림 10. 가구별 CPU이용률
Fig. 10. CPU utilization for each family

그림 10의 세로축은 식 1의 R값을 의미한다. 기존 그리드 시스템이 0.09인 반면 가장 시청시간이 짧은 A가구는 1.01로 약 10배, 평균적으로 1.78의 효율성을 보여 약 20배의 성능 향상이 있음을 알 수 있다. 따라서 실험 결과에 의하면 제안하는 그리드 스케줄링 정책이 기존 그리드 스케줄링 정책 보다 적합하다는 것과 자원 이용률이 효율적임을 보였다.

V. 결론

사용자들의 스마트 TV 사용 패턴이 기존 PC사용 패턴과 달라서 PC기반의 그리드 시스템을 그대로 스마트TV에 적용하기에는 적합하지 않다. 따라서 본 논문에서는 스마트 TV에 적합하도록 개선된 시스템을 제안하고 PC기반의 그리드 컴퓨팅 시스템과의 비교 실험을 통해 성능을 평가하였다.

첫째로 PC와 달리 멀티플랫폼, 멀티OS인 스마트 TV환경에서 호환성 및 이식성을 확보하기 위해 자바스크립트 기술을 이용하였다. 둘째로 기존 그리드 시스템의 스케줄 방식이 스마트TV에 맞지 않으므로 TV시청 중에 에이전트가 남는 CPU자원을 이용하도록 정책을 수정하였다. 그 결과 PC기반의 그리드 컴퓨팅 시스템의 스케줄 방식에 비해 제안하는 스마트TV에 적합하도록 구현한 그리드 컴퓨팅 시스템 방식이 평균 약 1.78의 자원을 제공해 주어 기존 그리드 시스템 대비 약 20배 더 효율적이었다.

Acknowledgments

본 연구는 정보통신산업진흥원의 IT/SW 창의연구과정의 연구결과로 지식경제부와 에스케이씨앤씨 주식회사에 의해 지원된 과제로 수행되었음”

(NIPA-2010-C1810-1006-0002)

참고문헌

- [1] DisplaySearch Quarterly TV Design and Features Report, "DisplaySearch Connected TV Forecast," 2011.
- [2] C. H. Noh, K. C. Cho, Y. B. Ma, and J. S. Lee, "Grid Resource Selection System Using Decision Tree Method," The Korea Society of Computer and Information, Vol. 13, No. 1 pp. 1-10, 2008.
- [3] J. Y. Choe, W. J. Lee, and C. H. Jeon, "Real-Time Job Scheduling Strategy for Grid Computing," The Korea Society of Computer and Information, Vol. 15, No. 2, pp. 1-8, 2010.
- [4] M. Baker, "Cluster Computing White Paper," University of Portsmouth, UK, pp. 3-12, 2000.
- [5] O. Nov, D. Anderson, and O. Arazy, "Volunteer Computing: A Model of the Factors Determining Contribution to Community-based Scientific Research," Proceedings of the 19th international conference on World wide web. pp. 741-750. 2010.
- [6] D. P. Anderson, "SETI@home: an experiment in public-resource computing," Communications of the ACM, Vol. 45, Issue 11, pp. 56-61, 2002.
- [7] Google TV Features, <http://www.google.com/tv/features.html>
- [8] Web Browse on Apple TV With Safari HD, <http://gizmodo.com/357626/web-browse-on-apple-tv-with-safari-hd-out-now>
- [9] A. Welc, R. L. Hudson, T. Shpeisman, and A. Adl-Tatabai, "Generic Workers - Towards Unified Distributed and Parellel JavaScript Programming Model," Programming Support Innovations for Emerging Distributed Applications, ACM New York, pp. 1-25. 2010.
- [10] Web Worker, <http://caniuse.com>
- [11] S. Okamoto, "Load Distribution by Using Web Workers for a Real-Time Web Application," iiWAS2010, pp. 592-597, 2010.
- [12] Samsung Electronics, "Cable business development in the Smart TV environment," Samsung Electronics, <http://goo.gl/pSPWY>, p. 4, 2011.
- [13] ECMAScript Language Specification, <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>
- [14] Weekly ratings report, <http://home.tnms.tv/rating>
- [15] Watch pattern survey for each household, <http://www.reportshop.co.kr/dview/50675>

저자 소개



김 현 식

2010~ 현재 : 인하대학교 컴퓨터정보공학과 석사과정
 관심분야 : 고성능 컴퓨팅, 인공지능
 Email : wbstory@eslab.inha.ac.kr



조 근 식

1982 : 인하대학교 전자계산학과 학사
 1991 : City University of New York, Computer Science, Ph.D.
 1991 ~ 현재 : 인하대학교 컴퓨터정보공학과 교수
 관심분야 : 인공지능, 시맨틱웹, 지능형 에이전트 시스템
 Email : gsjo@inha.ac.kr