

## UbiFS: 유비쿼터스 컴퓨팅 환경에 적합한 파일 시스템

김경윤<sup>o</sup> 송인준 김영필 유혁  
고려대학교 컴퓨터학과  
{kykim<sup>o</sup>, ijsong, ypkim, hxy}@os.korea.ac.kr

### UbiFS: A File system for Ubiquitous Computing Environments

Kyeong-Yun Kim<sup>o</sup> In-Jun Song Young-Pill Kim Hyuck Yoo  
Dept. of Computer Science and Engineering, Korea University

#### 요 약

유무선 통신 기술의 발전으로 언제 어디서나 컴퓨터를 이용하는 유비쿼터스 컴퓨팅에 관심이 집중되고 있다. 하지만 기존의 데이터 관리 방법은 유비쿼터스 컴퓨팅 환경에서 파일 서비스를 제공하는 데 어려움이 있다. 유비쿼터스 컴퓨팅 환경의 파일 시스템은 데이터의 유용성, 응용 프로그램에 대한 투명성, 자동 데이터 선택 및 저장, 이질적인 장치의 상호 운용을 지원할 수 있어야 한다. 본 논문에서는 이러한 요구 사항을 충족시킬 수 있는 유비쿼터스 파일 시스템(UbiFS)을 제안한다. UbiFS는 클라이언트의 현재 상태를 XML을 이용하여 상황 정보(Situation Information)로 구조화하고 SI 메시지를 통해서 서버에게 서비스를 요청하게 된다. 서버는 클라이언트의 SI 메시지를 관리하는 매니저를 통해서 데이터 동기화와 같은 문제를 해결한다. 그리고 사용자가 직접 데이터를 선택하는 번거로움을 줄이기 위해서 UbiFS는 사용자의 데이터 접근 패턴을 분석하여 시스템 수준에서 데이터를 자동적으로 선택하고 저장할 수 있다.

간과 컴퓨팅 환경 사이에 이동이 가능하다.  
[2][3]

#### 1. 서 론

정보통신 기술이 발전하면서 많은 사람들은 미래 사회는 유비쿼터스 시대가 될 것이라고 말하고 있다. 그리고 이러한 시대를 이루기 위해서 유비쿼터스 컴퓨팅에 관한 수많은 연구가 이루어지고 있다. 유비쿼터스 컴퓨팅은 문자적 의미로 직역하면 '편재하는 컴퓨팅'로 해석할 수 있다. 즉, 언제, 어디서나 컴퓨터를 이용할 수 있다는 것이다. 이러한 기본적인 의미를 바탕으로 유비쿼터스 컴퓨팅은 컴퓨터가 위치할 수 있는 곳에 따라서 사이버 공간의 가상 컴퓨팅과 실세계의 리얼 컴퓨팅으로 나누어 생각해 볼 수 있다. 하지만 최근 유비쿼터스 컴퓨팅의 연구 동향은 소프트웨어, 하드웨어, 물리적인 개체를 차세대 컴퓨팅 환경으로 통합하여 가상세계와 실세계를 병합하려는 모습을 보이고 있다.[1] 그리고 이러한 유비쿼터스 컴퓨팅 환경은 다음과 같은 특징을 가지고 있다.

- **이질성 (Heterogeneity)**: 서로 다른 요구 사항과 형태를 가지고 있는 장치들이 존재한다. 다양한 종류의 공간에 소프트웨어가 맵핑된다.
- **분산 (Distribution)**: 소프트웨어는 공간 안에 분산된 자원을 이용할 수 있다.
- **상호작용 (Interaction)**: 소프트웨어가 원격으로 제어될 수 있으며 물리적인 공간 안에서 발생한 변화가 센서에 의해 감지된다.
- **네트워크 (Network)**: 소프트웨어 컴포넌트가 공간 내에서 유선으로 연결되어 있고 모바일 장치는 무선 네트워킹으로 통신이 가능하다.
- **이동성 (Mobility)**: 사용자와 장치가 물리적인 공

이러한 유비쿼터스 컴퓨팅 환경에서 데이터 관리는 시스템 수준에서 지원되어야 한다. 모바일 장치의 경우 갑작스러운 전력 중단, 불안정한 연결로 인해서 데이터의 유용성이 떨어질 수 있다. 그리고 자주 이용되는 데이터를 사용자가 직접 선택하여 전송하는 것은 사용자에게 부담이 될 뿐만 아니라 오류가 발생하기 쉽다. 또한, 사용자가 환경 안에 존재하는 다양한 종류의 데이터를 통합적인 방법으로 접근하는 것이 가능해야 한다. 따라서 유비쿼터스 컴퓨팅 환경은 파일시스템을 통해서 사용자와 장치의 데이터 접근을 지원해야 한다. 이러한 유비쿼터스 파일 시스템이 갖추어야 할 요건은 다음과 같다.

- **유용한 데이터 서비스**: 전원 차단이나 네트워크 연결 끊김과 같은 예기치 않은 상황의 발생으로 데이터가 손실될지라도 그것을 사용가능한 상태로 유지할 수 있도록 해야 한다.
- **자동 데이터 선택 및 저장**: 사용자의 데이터 접근 사례를 분석하여 시스템이 자동적으로 데이터를 제공하는 것이 필요하다.
- **응용 프로그램 투명성**: 데이터의 일관성 유지나 동기화, 복사를 사용자가 인식하지 못하도록 파일 서비스를 해야 한다.
- **이질적인 장치 사이의 상호 운용**: 컴퓨팅 환경 안에 존재하는 다른 장치의 데이터를 이용할 수 있도록 하는 것이 필요하다.[2][6]

본 논문의 구성은 다음과 같다. 2장에서는 유비쿼터스 컴퓨팅 환경에서 파일 서비스를 제공하기 위해 제안된 방법을 살펴본다. 그리고 3장에서는 유비쿼터스 컴퓨팅 환경에 적합한 파일 시스템을 제안한다. 마지막으로 4장에서는 결론 및 향후 과제를 제시한다.

본 논문은 대학 IT 연구 센터 육성 지원 사업의 연구 결과로 수행되었음.

2. 관련 연구

유비쿼터스 컴퓨팅 환경의 파일 서비스에 대한 연구는 유비쿼터스 환경과 전통적인 분산 환경의 차별성을 증명으로 한 것과 유비쿼터스 환경을 분산 환경의 연장선으로 파악하는 방향으로 이루어지고 있다.

2.1 컨텍스트 파일 시스템 (Context File System)

CFS는 유비쿼터스 환경의 작업 공간을 활동 공간 (Active Space)으로 정의하고 활동 공간 안의 환경, 작업 정보 등을 컨텍스트로 나타낸다. 컨텍스트는 환경 정보인 외부 컨텍스트와 응용 프로그램이나 현재 장치 정보를 나타내는 내부 컨텍스트로 구분된다. 이러한 컨텍스트의 내용은 상호 운용을 위해서 XML로 구조화되고 서버에 의해서 관리된다. 하지만 CFS는 유비쿼터스 환경의 불안정한 네트워크 연결, 장치의 전력 제약 문제를 고려하고 있지 않기 때문에 데이터 유용성을 보장하기 어렵다.[3][4][5]

2.2 유비쿼터스 모바일 파일 서비스 (UbiData)

모바일 환경의 파일 서비스를 위해서 UbiData는 더블미들웨어 기반의 구조를 제안하고 있다. 즉, 자동 데이터 선택, 데이터 저장, 일관성 유지, 기기종 간의 통신, 연결이 끊어진 상태의 동작을 지원하기 위해서 M-MEM (Mobile-Mobile Environment Manager), F-MEM (Fixed-Mobile Environment Manager) 두 가지 미들웨어를 시스템에 추가하였다. 그러나 UbiData는 유비쿼터스 환경 정보에 대한 추상화의 부재로 클라이언트가 상황에 적절한 파일 서비스를 받기 어렵다는 문제점을 갖는다.

[2]

3. UbiFS

유비쿼터스 컴퓨팅 환경은 클라이언트가 특정한 상황을 갖는다는 점에서 전통적인 분산 컴퓨팅 환경과 구별될 수 있다. 따라서 본 논문에서는 클라이언트의 상황 정보를 XML을 이용하여 상황 정보(Situation Information)로 구조화할 것을 제안한다. 서버는 클라이언트의 SI를 인식하여 적절한 서비스를 제공할 수 있다. 그리고 기기종의 파일 시스템을 마운트할 수 있는 UbiFS는 데이터 유용성, 자동 데이터 선택 및 저장, 응용 프로그램 투명성을 보장할 수 있도록 했다.

3.1 상황 정보 (Situation Information)

상황 정보는 유비쿼터스 컴퓨팅 환경에서 현재 클라이언트의 상태를 나타낸다. 공간 내의 현재 위치, 공간 안에서의 활동, 현재 날씨, 장치 특징, 데이터 존속 시간, 데이터 포맷 같은 사항은 상황 정보의 구체적인 예가 된다. 상황 정보는 시스템 정보와 환경 정보로 나누어 생각해 볼 수 있다. 시스템 정보는 클라이언트의 로컬 파일 시스템 정보, 접근 데이터 정보, 사용자가 요구하는 데이터 포맷과 같은 것들이 있다. 환경 정보는 위치, 날씨 등을 포함한다. 클라이언트와 서버 간의 통신은 XML 기반의 프로토콜을 통하여 이루지고 상황 정보는 XML

메시지 형태로 구조화되어 서버에게 전달된다. 즉, 위에서 언급된 구체적인 예들은 XML 문서의 DTD로 작성된다. 플랫폼 독립적인 XML을 통해서 기기종 간의 상호 운용을 보장할 수 있다. 예를 들어, 프리젠테이션 파일을 가지고 있는 클라이언트 A와 이미지 뷰어만을 가지고 있는 클라이언트 B가 있다고 가정해 보자. B가 A의 데이터를 받아서 보기 위해서는 프리젠테이션 파일이 이미지 오브젝트로 전환되는 작업이 필요하다. 이러한 경우 프리젠테이션 파일에 관한 B의 요청은 SI 메시지의 형태로 서버에 전달되고 서버의 변환 모듈을 통해서 B에 적합한 데이터가 생성된다. (A의 데이터 사본은 이미 서버에 존재한다) 그리고 생성된 데이터가 B로 보내져서 프리젠테이션 파일을 이미지 뷰어로 볼 수 있게 된다.

3.2 시스템 구조

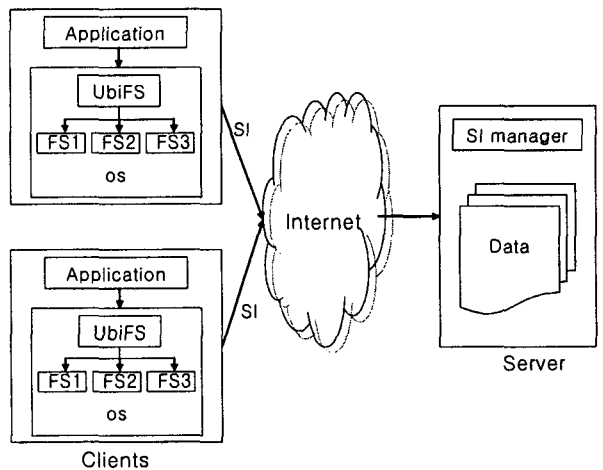


그림 1. 시스템 구조

그림 1에서 클라이언트의 UbiFS는 유비쿼터스 컴퓨팅 환경을 지원하기 위해서 기존의 파일 시스템의 능력을 확장시켰다. 첫째, 데이터 유용성을 위해서 기존의 분산 시스템에서 사용된 파일 복제 및 연결이 끊어진 상태의 동작을 지원한다. 파일 복제를 통해서 손실된 파일의 내용을 복구하고 연결이 끊어지더라도 사용자가 그것을 인식하지 못하도록 파일 서비스를 할 수 있다. 둘째, 커널을 훅킹하여 사용자의 파일 접근 이벤트를 수집하고 이것을 분석하여 쉽게 다음 워킹 셋을 예측하는 자동 파일 선택 및 저장 메커니즘을 지원한다. 셋째, 쓰기 연산을 트래킹하여 클라이언트의 파일 업데이트를 감지함으로써 서버와 클라이언트 사이에 데이터를 동기화한다. 물론, 클라이언트 내의 로컬 업데이트는 SI메시지를 통해서 서버에 전달되며 서버 내의 데이터가 갱신된다. 넷째, 다양한 종류의 파일 시스템을 마운트 할 수 있도록 VFS(Virtual File System)의 공통 파일 모델을 따른다. 다른 클라이언트의 파일 시스템을 마운트하여 가상 파일 계층을 만들 수 있다.

한편, 서버의 상황 정보 매니저(SI manager)는 클라이

언트로부터 전달된 XML 메시지를 관리하는 역할을 한다. 즉, SI 메시지 파일의 일관성을 유지하고 SI 파일 동기화를 위한 업데이트, 손실된 파일 복구를 위한 정보를 제공한다.

#### 4. 결론 및 향후 과제

유비쿼터스 컴퓨팅 환경에 적합한 파일 시스템은 세 가지 요건을 갖추어야 한다. 우선, 언제 어디서나 컴퓨팅이 가능해야 한다는 목적을 충족시킬 수 있도록 클라이언트의 매 상황을 데이터로 추상화시킬 수 있어야 한다. 그리고 유비쿼터스 파일 시스템은 전력 공급 문제, 연결의 불안정성으로 인한 데이터 손실을 최소화 시켜서 데이터 유용성을 확보할 수 있어야 한다. 마지막으로 유비쿼터스 환경의 파일 서비스는 시스템 수준에서 파일 서비스를 제공함으로써 사용자의 관여를 최소화시킬 수 있어야 한다. 이러한 유비쿼터스 컴퓨팅 환경의 파일 시스템이 갖는 요구 사항은 본 논문에서 제안된 UbiFS 구조로 충족될 수 있다. 향후 과제로는 UbiFS 구조를 유비쿼터스 컴퓨팅 환경에 적용시키고 사용자의 다음 워킹 셋을 보다 정확하게 예측하는 메커니즘을 보완할 계획을 가지고 있다.

#### 참고문헌

- [1] 김완석, "각국의 유비쿼터스 컴퓨팅 개념 비교", TTA Weekly On-line IT Standard Magazine 2003.
- [2] Jinsuo Zhang, Abdelsalam (Sumi) Helal and Joachim Hammer, "Ubidata: Ubiquitous Mobile File Service", *Eighteenth ACM Symposium on Applied Computing*, Melbourne, FL, 2003.
- [3] A Context File System for Ubiquitous Computing Environments, Christopher K. Hess and Roy H. Campbell. Technical Report UIUCDCS-R-2002-2285 UIIU-ENG-2002-1729, University of Illinois at Urbana-Champaign, July 2002.
- [4] Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, In *IEEE Pervasive Computing*, pp. 74-83, Oct-Dec 2002.
- [5] Bray, T., Paoli., and Sperberg-McQueen, C. M., Extensible Markup Language (XML), *W3C Proposed Recommendation*. Dec 1997.
- [6] Satyanarayanan, M., Coda: A Highly Available File System for a Distributed Workstation Environment, *Proceedings of the Second IEEE Workshop on Workstation Operating System*, Sep. 1989, Pacific Grove, CA.