

# 이동 컴퓨팅 환경을 위한 저널 파일 시스템의 설계 및 구현

정은영, 강용혁, 엄영익  
성균관대학교 전기전자 및 컴퓨터공학부  
e-mail:ifthen@ece.skku.ac.kr

## The Design and Implementation of Journal File System for Mobile Computing Environments

Eun-Young Chung, Yong-Hyeog Kang, Young Ik Eom  
School of Electrical and Computer Engineering,  
Sungkyunkwan University

### 요약

이동 컴퓨팅 환경에서의 통신은 유선 네트워크 상에서 이루어지는 통신에 비해 대역폭이 적고 패킷 에러율이 크다는 제약 사항들을 갖는다. 이러한 환경에서 분산 파일 시스템 서비스는 데이터 송수신 시간과 위치에 따라 일정하지 않고 데이터가 손실되었을 때 복구 가능성이 낮은 문제점이 있다. 본 논문에서는 기존의 파일 시스템에 사용되는 저널링 기법과 기존의 이동 컴퓨팅 환경을 위한 파일 시스템(mobile filesystem)에서 사용되는 연산 적재(operation shipping) 기법을 혼합한 방식으로 이동 컴퓨팅 환경을 위한 파일 시스템을 고안하였다. 이 기법은 기존의 이동 컴퓨팅 환경을 위한 파일 시스템보다는 데이터 복구가능성을 높이고 데이터 재통합(reintegration) 기능을 효율적으로 수행할 수 있다.

### 1. 서론

무선 통신 기술의 발달과 휴대용 컴퓨터의 소형화로 인해 이동 컴퓨팅 환경에서 무선 통신을 이용하는 사용자 수가 증가하고 있다. 그러나 현재는 이를 지원하기 위한 이동 컴퓨팅 환경에서의 파일 시스템들이 갑작스런 서버 다운과 휴대용 컴퓨터의 에너지 소모, 통신 장소 이동 등으로 인해 야기되는 데이터 손실 문제에 적절히 대응하지 못하고 있는 상황이다.

이동 컴퓨팅 환경에서의 통신은 유선 네트워크 상에서 이루어지는 통신에 비해 대역폭이 작고, 신뢰성이 낮으며, 호스트의 이동으로 인한 서버와의 통신 단절이 자주 발생하는 등의 제약사항을 고려해야 한다. 또한 이동 컴퓨팅 환경에서 이루어지는 통신은 제한된 에너지로 일정시간 통신을 유지해야 하므로 에너지 소비의 효율성 문제를 고려해야 한다. 일

반적으로 데이터를 송신하는 것은 데이터를 수신하는 것보다 비용(cost)이 높으며 에너지 소비가 크므로 송신 데이터의 양을 수신 데이터의 양보다 줄여야 한다. 이동 컴퓨팅 환경에서의 통신을 위한 파일 시스템은 이러한 제약사항들을 고려하여 설계되어야 한다[1].

기존의 이동 컴퓨팅 환경에서의 이동 컴퓨팅 환경을 위한 파일 시스템은 데이터의 일관성(consistency)과 복구 가능성(recoverability)의 측면에서 볼 때, 캐쉬 상에 필요한 모든 정보를 저장하였다가 데이터가 수정되었을 때 이를 서버에게 전송하는 방식을 택하였다. 이러한 경우 서버가 다운되거나 클라이언트와 서버 사이의 접속이 끊어졌을 때, 수정된 데이터의 정보는 손실된다. 이러한 문제점을 보완하기 위하여 본 논문에서는 저널링 기능을 추가한 이동 컴퓨팅 환경에서의 파일 시스템을 제안한다.

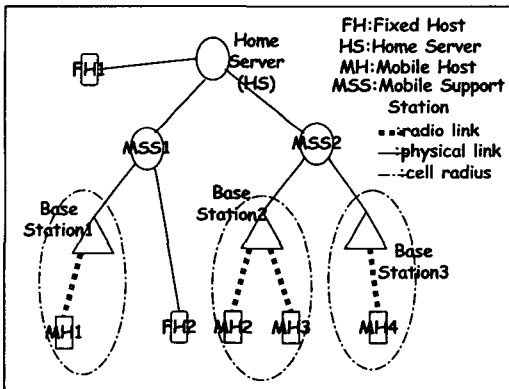
저널링 기능을 추가한 이동 컴퓨팅 환경에서의 파일 시스템은 손실된 파일 시스템 정보에 대한 복구 가능성을 높일 수 있으며, 서버와 재접속 할 때 클라이언트의 데이터에 대한 재통합(reintegration) 기능을 강화할 수 있다[6,7].

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 이동 컴퓨팅 환경에서의 파일 시스템 특성과 저널 파일 시스템에 대해 설명한다. 3장에서 본 논문에서 제안하는 이동 컴퓨팅 환경을 위한 저널 파일 시스템에 대한 설계와 구현 방법을 설명하며, 4장에서 결론과 향후 과제를 알아본다.

## 2. 관련연구

### 2.1 이동 컴퓨팅 환경

이동 컴퓨팅 환경은 그림 1과 같이 홈서버(HS), 이동 지원국(MSS), 기지국(Base Station), 이동 호스트(MH)들로 구성되어 있다.



[그림 1] 이동 컴퓨팅 환경

홈 서버는 이동 호스트가 등록되어 있는 서버이며, 이동 호스트의 파일들에 대해 저장장치(storage)를 지원한다. 기지국은 무선 통신 장비를 갖춘 고정 호스트로서 자신이 관리하는 셀안에 있는 이동 호스트들과 통신을 한다. 이동 호스트는 데스크탑 컴퓨터의 기능과 무선 통신의 기능을 동시에 갖추고 있는 휴대용 컴퓨터이다.

이동 컴퓨팅 환경에서는 대역폭이 작고 패킷 에러율이 크며 네트워크를 통해서 정보를 전송하는 비용이 비싸다. 이동 호스트는 패킷을 수신하는 것보다 송신하는 비용이 더 비싸며 메시지 송신은 수신보다 에너지를 더 많이 소비한다. 즉, 이동 지원국과 이동

호스트 사이의 통신 비용은 비대칭적이다. 또한 이동 호스트의 이동성으로 인해 연결 상태가 자주 변경되어 재접속 오버헤드가 크다[2].

### 2.2 이동 컴퓨팅 환경에서의 파일 시스템

이동 컴퓨팅 환경을 위해 제안된 파일 시스템들은 무선 통신과 이동 호스트의 이동성을 극복하기 위해 다음과 같은 특성을 가지고 설계된다[2,3].

- 1) 이동 호스트들을 위한 비접속 연산 (disconnected operation)
  - 비 접속 이동 호스트들의 데이터 재통합 기능
  - 무선 대역폭에 대한 적응성
- 2) 오류 복원력(resilience)
  - 읽기/쓰기 복제(replication) 서버
  - 서버간 충돌 해결(conflict resolution)
  - 이동 호스트의 비접속과 네트워크 분할 관리
- 3) 성능과 확장성(scalability)
  - 파일, 디렉토리 및 속성(attribute) 캐싱 기법
  - write back 메커니즘

### 2.3 저널 파일 시스템(Journal Filesystem)

파일 시스템에서 캐싱 기법은 일반적으로 성능을 증가시키기 위해 사용되어진다. 그러나, 버퍼로부터 디스크에 데이터를 쓰는 작업을 완료하기 전에, 시스템에 장애(crash)가 발생되었을 때 데이터의 일관성(consistency)을 유지할 수 없다는 문제가 생긴다. 그러나 데이터가 캐쉬에서 지워졌다 하더라도 하드 디스크 상에서는 남아 있으며, 이것은 저널 파일 시스템에 그대로 이용되어진다.

저널 파일 시스템은 연산명(operation name) 뿐만 아니라, 연산을 수행하기 전의 인수 내용(argument content)에 대한 정보도 기록한다. 모든 단일 트랜잭션(single transaction)이 수행된 후에 commit 연산이 수행되고 캐쉬의 내용이 디스크에 쓰여지므로, 시스템에 장애가 발생되더라도 첫 번째 commit 상태로 로그 파일을 추적할 수 있다. 따라서 저널 파일 시스템은 기존의 파일 시스템보다 일관성과 복구 가능성을 높일 수 있으며, 시스템 오류시 파일 시스템 검사(fsck) 같은 재시작 시간(restart-time) 유틸리티에 의존하지 않고 단지 최근에 참조된 로그 기록만 조사함으로써 시스템 재시작 시간을 단축시킬 수 있다. 이러한 저널 파일 시스템으로는 SGI의 XFS, IBM의 JFS(Journaled Filesystem)등이 있으며 본 기법에서는 IBM의 JFS를 기반으로 한다[7]. JFS

의 디스크 구조와 로그 메커니즘은 다음과 같다[6].

### 2.3.1 애그리게이트와 파일셋, 익스텐트

애그리게이트(aggregate)는 JFS에서 슈퍼블록과 할당 맵(allocation map)을 포함하는 특정한 형태의 디스크 블록 개념이며, 파일셋(Fileset)은 마운트 가능한 서브 트리(sub-tree)를 구성하는 파일과 디렉토리의 집합이다. 익스텐트(extent)는 JFS 객체에 할당되어지는 연속적인 애그리게이트 블록 시퀀스이다. 하나의 익스텐트는 하나의 애그리게이트에 할당되어야 하지만, 익스텐트가 큰 경우, 여러 개의 애그리게이트 그룹을 가질 수 있다. 하나의 익스텐트의 크기는 1에서  $2^{24}-1$ 까지 가능하다. 하나의 익스텐트는 <logical offset, length, physical offset>으로 구성되며, 이러한 각각의 익스텐트들에 대한 검색 및 읽기와 쓰기 성능을 향상시키기 위해 전체 익스텐트의 주소 할당 구조를 B+ 트리 형태로 구성하였다.

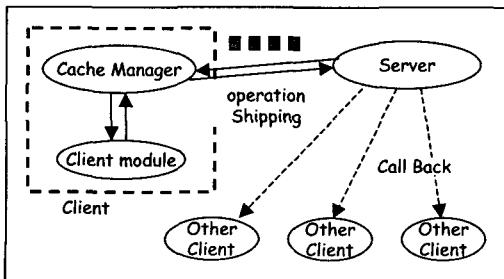
### 2.3.2 로그(Logs)

JFS 로그는 각각의 애그리게이트에 유지되어지며, 메타 데이터 연산에 대한 정보를 기록하는데 사용되어진다. 디렉토리와 파일에 대한 수정이 완벽하게 수행되어졌을 때, 로그 연산 commit을 기록한다.

## 3. 설계 및 구현

### 3.1 개요

본 논문에서 고안할 이동 컴퓨팅 환경을 위한 파일 시스템은 IBM의 JFS를 기반으로 하며, 기본적인 동작 방식은 그림2와 같다. 클라이언트와 서버는 각각 저널 파일 시스템의 기능을 갖춘 JFS로 구성되어 있으며, 익스텐트 단위로 통신하게 된다. 즉, 클라이언트는 디스크에 로그 데이터를 쓴 후 한번에 익스텐트 단위로 서버와 통신한다.



[그림 2] 이동 컴퓨팅 환경을 위한 저널 파일 시스템 모델

서버에 장애가 발생되었거나, 서버와 통신하고 있는 클라이언트 중의 하나가 서버상의 로그 데이터를 수정하였을 때, 같은 로그 데이터를 참조하는 모든 클라이언트에게 콜백(callback)을 수행한다. 로그 데이터가 서버나 다른 클라이언트에 의해 수정되었을 경우 익스텐트 전체의 구조적 변경 없이 변경된 익스텐트만 재전송한다.

## 3.2 제안기법 설계와 구현

### 3.2.1 캐쉬 매니저

본 제안 기법에서도 캐싱을 통해서 네트워크의 트래픽을 감소시킨다. 클라이언트와 서버 모두 캐쉬 매니저를 가지고 있으며, 사용자들은 캐쉬를 통해 많은 양의 데이터들에 빠르게 접근할 수 있다. [1]

본 논문에서 캐쉬 매니저는 Andrew 파일 시스템의 커널 레벨 캐쉬 매니저와는 다르게 어플리케이션 레벨로 캐쉬 매니저를 구현한다. 각각의 캐쉬 엔트리는 파일이나 디렉토리를 나타내는 Fid와 ExtentId로 구성되며, ExtentId는 디스크상에 존재하는 익스텐트에 대한 고유 식별자이다. 캐쉬상에 존재하지 않지만, 로그 데이터로 남아있는 정보를 ExtentId로 디스크 상에서 검색할 수 있다.

### 3.2.2 중앙 데이터 관리자

서버는 요구된 데이터와 서비스한 데이터에 대한 정보를 리스트로 유지하고 있다. 만약 클라이언트에 의해 요구된 데이터가 유효하지 않거나 서비스된 데이터가 갱신되었을 경우 서버는 이를 클라이언트에게 통보한다[1,2,5].

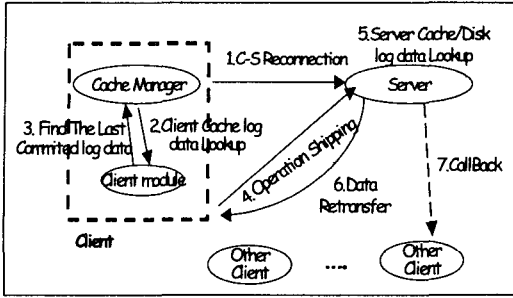
서버는 현재 접속하고 있는 클라이언트들 중, 서버상의 같은 익스텐트를 참조하고 있는 클라이언트들에 대한 리스트를 유지한다. 클라이언트들 중의 하나가 익스텐트를 변경시켰을 경우, 변경된 익스텐트에 대해서 ExModified 플래그가 설정되며 이를 참조하고 있는 다른 모든 클라이언트에게 알리게 된다.

### 3.2.3 연산 적재(Operation Shipping)

연산 적재는 열려진 파일의 close 연산이 수행되어질 때 한번에 클라이언트 혹은 서버에 쓰는 방식으로 행해지며, 익스텐트 단위로 전송되어진다[4].

이동 컴퓨팅 환경의 특성으로 인해 서버의 다운, 클라이언트 이동, 또는 에너지 소비 등으로 인해 클라이언트와 서버 사이의 접속이 단절되었을 경우 재

통합을 위한 연산 적재 메커니즘은 그림 3과 같다.



[그림 3] 재통합을 위한 연산 적재 메커니즘

먼저, 클라이언트 캐쉬에서 클라이언트와 서버 연결에 대한 로그 데이터를 검색(lookup)한다. 클라이언트와 서버가 단절될 당시 수행되어지던 연산은 캐쉬상에 남아있지 않으므로, 이를 클라이언트 디스크에서 찾는다. 클라이언트 디스크에서 마지막으로 commit 연산이 수행된 로그 데이터를 검색한 후 서버로 연산 적재를 수행한다.

클라이언트에서 보내온 로그 데이터를 서버 캐쉬와 디스크에서 검색하고 클라이언트 요구에 부합하는 데이터를 서버로부터 재전송한다.

### 3.3.3 원격 파일 접근 기법

응용 프로그램이 접근할 수 있는 파일의 종류는 자신의 시스템에 장착된 지역 파일들과 서버에서 제공하는 원격 파일들이 있다. 이 중 지역 파일은 시스템에 존재하는 파일 시스템에게 요청하여 접근할 수 있으나 원격 파일의 접근은 다른 기법을 요구한다. 본 논문에서 제안하는 기법에서도 NFS(Network File System)나 AFS(Andrew File System)에서처럼 마운트 프로토콜과 RPC(Remote Procedure Call) 프로토콜을 사용하여 투명한(transparent) 원격 파일 접근을 지원한다. 즉 원격 파일 시스템을 지역 파일 시스템에 마운트하여 원격 파일들의 이름을 명명하고, 명명된 이름으로 원격 파일을 접근할 때에는 RPC 프로토콜을 사용하여 원격에 있는 파일 시스템에게 요구하고 그 결과를 응용 프로그램에게 보여주는 방식을 사용한다.

## 4. 결론 및 향후 과제

오늘날 이동 컴퓨팅 환경에서의 통신은 수요가 증가하는 만큼 빠른 속도로 발전하고 있다. 이동 컴퓨팅 환경을 위한 파일 시스템에 대한 연구는 이동 컴

퓨팅 환경에 기인한 제약사항들을 개선하는데 중점을 두고 진행되고 있다. 본 논문에서도 이동 컴퓨팅 환경에서의 무선 통신이 갖은 데이터 손실을 야기한다는 사실에 중점을 두었으며, 이에 데이터의 강력한 복구 가능성이라는 특성을 지닌 저널 파일 시스템과의 접목을 제안하였다.

이동 컴퓨팅 환경을 위한 저널 파일 시스템은 트랜잭션이 수행된 연산에 대한 로그 데이터를 지니고 있으므로 갑작스런 시스템 장애에도 데이터 손실을 줄일 수 있다. 또한, 로그 데이터를 통해 데이터를 복구함으로써 이전에 수행된 연산들을 재개할 수 있다. 향후 본 논문에서 제안한 기법에 대해 시뮬레이션에 의한 성능 평가를 수행하여 기존 파일 시스템과 비교 분석하고 제안 기법의 장단점을 검증하는 일이 필요할 것이다.

## 참고문헌

- [1] Nikki Mirghafori and Anne Fontaine, "A Design for File Access in a Mobile Environment," IEEE Workshop on Mobile Computing Systems and Applications, Dec. 1994.
- [2] M. Satyanarayanan, "Fundamental challenges of mobile computing," ACM Symposium on Principles of Distributed Computing, 1995.
- [3] P. Braam, "The Coda Distributed File System," <http://www.coda.cs.cmu.edu/ljpaper/lj.html>.
- [4] Yui-Wah Lee and Kwong-Sak Leung, "Operation-based update Propagation in Mobile File System," USENIX Annual Technical Conference, June 6-11, 1999.
- [5] P. Braam, M. Callahan, and P. Schwan, "The InterMezzo File System," The Perl Conference 3, O'Reilly Open Source Convention, Aug. 1999.
- [6] Steve Best and Dave Kleikamp, "JFS layout: How the Journaled File System handles the on-disk layout," <http://www-106.ibm.com/developerworks/library/jfslayout/index.html>.
- [7] Juan I. Santos Florido, "Journal File Systems," <http://www.linuxgazette.com/issue55/florido.html>.