이동 컴퓨팅 환경에서 파일 이주를 이용한 접근 지연 감소 기법

(A Scheme for Reducing File Access Latency with File Migration in Mobile Computing Environments)

한 문 석 † 박 상 윤 †† 엄 영 익 †††
(Moon Seog Han) (Sang Yun Park) (Young Ik Eom)

요 약 본 논문은 이동 컴퓨팅 환경에서 다중 이동 클라이언트를 위한 원격 파일 접근 문제를 다루었다. 이러한 환경에서 사용자의 이동성은 파일 접근에 영향을 미치며 이동 호스트는 제한된 용량의 비휘발성 저장 장치를 가진다는 점에서 자원에 대한 심각한 제약이 따른다. 따라서, 통신 및 연산 부하가 파일 접근 지연을 야기할 수 있다.

본 논문은 이러한 문제를 해결하기 위하여 파일 이주 기법을 이용하여 접근 지연을 줄일 수 있는 기법을 제안한다. 이 기법의 목표는 모든 이동 호스트들에게 가능하면 신속하게 요청한 파일을 전달하여 접근 지연을 최소화하는데 있다. 우리는 이동 호스트의 파일 요청이 있을 때 서버가 파일 이주를 결정하는 주문형 기법을 개발하였다.

시뮬레이션을 통하여 파일 접근 빈도, 파일 크기, 이동성 등이 파일시스템 접근 지연에 미치는 영향을 실험하였다. 시뮬레이션 결과에서 제안된 이주 기법은 파일 접근 빈도가 높고 이동성이 작은 이동 호스트가 요청한 파일에 대한 접근 지연을 줄이는데 효율적이라는 것을 보였다.

Abstract We consider some problems of remote file accesses of multiple clients in mobile computing environments. In this environment, user mobility affects file access latency. Mobile hosts have severe resource constraints in terms of limited size of non-volatile storage. Thus, the burden of computation and communication load raise file access latency.

In this paper, we propose a scheme for reducing the file access latency through the file migration. The objective is to minimize the latency of file accesses for all mobile hosts which delivering the file to clients as quickly as possible. We develope an on-demand scheme which determines when the file server should migrate files to another server, or when it should transfer files to mobile hosts.

Using simulation, we examine the effects which parameters such as file access frequency, file size, mobility rate have on file system access latency. Through simulation results, we show that our proposed migration scheme is effective in reducing the access latency on the requested file of a mobile host with high access rate and low mobility.

1. 서 론

최근 이동 컴퓨팅 환경은 무선 이동 통신 기술의 발전과 휴대용 단말기의 보편화에 따라 빠르게 성장하고 있다. 이동 컴퓨팅 환경은 기존의 분산 시스템을 구성하는 유선망 상에 위치한 고정 호스트들과 무선망 상에서 통신하는 이동 호스트들로 구성된다. 고정 호스트들 중에서 서버는 각종 정보 서비스를 제공하고 이동 지원국(mobile support station)은 이동 호스트와 통신할 수 있도록 무선 인터페이스를 지원할 뿐만 아니라 이동 컴퓨팅 환경의 지리적 영역인 한 개의 셀(cell)을 관리한다[1].

† 종신회원 : 원주대학 행정전산과 교수
mshan@sky.wonju.ac.kr
†† 비 회 원 : 성균관대학교 전기전자및컴퓨터공학부
bronson@ece.skku.ac.kr
††† 종신회원 : 성균관대학교 전기전자및컴퓨터공학부 교수
yieom@ece.skku.ac.kr
논문접수 : 2001년 1월 11일
심사완료 : 2001년 9월 5일

기존의 분산 시스템에 무선 환경을 갖춘 이동 분산 시스템 환경에서 호스트의 이동성은 새로운 형태의 고려사항이 될 수 있다. 시스템 전체적인 비용 측면에서 이동성으로 말미암아 기존에 고려했던 비용 외에도 이동 호스트를 탐색하는 비용이 추가된다. 즉, 이동 호스트의 이동 정보가 많으면 비용이 적게 들며, 그렇지 않으면 비용이 많이 들게 된다. 그러므로 이동성은 파일 배치에 실질적인 영향을 주게 된다.

분산 시스템에서는 비용 문제를 효율적으로 해결하고 시스템 자원들을 네트워크를 통하여 공유하여 많은 사용자들이 자원에 대한 접근성(accessibility)을 높일 수 있었다. 분산 시스템의 중요한 자원인 파일들은 다양한 크기로 존재하며 네트워크를 통하여 고정 호스트들에게 분산되어 있다. 과다한 파일 접근 요청은 부하를 불균등하게 하고 성능을 저하시킬 수 있다. 과거 분산 시스템 연구에서는 이러한 문제를 파일 이주(file migration)와 파일 복제(file replication) 기법을 통해 해결하여왔다.

이러한 컴퓨팅 환경은 장소와 시간에 구애받지 않고 이동 클라이언트에게 원격지에서 정보를 접근할 수 있도록 해 주어야 한다. 이동 클라이언트는 원격지 서버에 있는 파일들을 이동 지원국에 요청한다. 이동 지원국은 원격지 서버에서 파일을 전송 받아 이동 클라이언트에게 전달한다. 분산 시스템 환경이 유선 망에서 안정적인 파일 서비스를 하는 것과 달리 이동 컴퓨팅 환경은 원격지 서버로부터 파일을 전송 받는데 있어 분산시스템에서 파일 접근 지연의 원인이었던 네트워크 트래픽(traffic)과 서버 작업부하(workload)에 추가로 이동 호스트들의 이동성(mobility), 그리고 무선망에서의 적은 대역폭(bandwidth) 등으로 말미암아 파일 접근을 지연시킬 수 있다. 또한 이동 클라이언트가 이동 지원국에게 빈번히 파일 서비스를 요구함으로서 호스트들이 대역폭 경쟁(contention)을 야기해 전체적인 서비스 지연이 발생할 수 있다[2].

이동 호스트들은 서비스 요구에 대해 빠른 응답을 기대하며 짧은 기간 동안 셀에 머무르거나 다른 셀로 이동할 가능성이 있기 때문에 고정 호스트들은 시의 적절하게 서비스 요구에 응답해야 한다. 즉, 사용자가 이동하는 중에도 자신이 원하는 작업을 계속할 수 있도록 지원해야 한다. 이것은 사용자가 이동 중에도 자신이 원하는 서비스와 자원을 지연 없이 투명하게(transparently) 접근할 수 있는 위치-독립적 작업(location-independent work)을 지원해야 한다는 의미이다.

이를 위하여 시스템 환경에 심한 오버헤드를 주지 않으면서 투명하게 서비스 지연을 방지할 수 있는 기법이 클라이언트 측에 파일들을 캐싱(caching)하는 것이다. 캐싱은 클라이언트 측에 원격지 서버로부터 가져온 파일을 저장하여 네트워크 단절(disconnection)이 있을지라도 계속적인 서비스를 받을 수 있도록 해 준다. 이 기법은 캐쉬된 파일의 최신성을 보장하기 어렵고 PDA와 같은 개인 휴대용 단말기는 제한된 자원, 즉 낮은 속도의 CPU, 제한된 메모리 그리고 디스크가 없는 환경이 대부분이므로 연산 및 통신 부하가 과중하게 걸리는 단점이 있다. 이는 고정 호스트와 이동 호스트간에 부하를 분산시킬 수 없다는 것을 의미한다.

이 문제의 해결을 위하여 이동 클라이언트가 홈 서버로부터 멀리 떨어져 있는 정도에 따라 자주 사용하는 파일과 서비스를 이동 셀의 서버에 동적 복제(dynamic replication)를 함으로써 정보를 공간적으로 분산시키는 것이 한 방법이 될 수 있다. 그러나 이 기법은 파일 사본들 간에 일치성을 유지하기 어려운 단점이 있다.

본 논문에서는 자원이 제한적인 이동 호스트들이 요구하는 파일을 홈 서버에서 이동 호스트가 체류하는 셀인 원격지 서버로 파일을 이주한 후 이동 호스트에게 서비스하는 파일 이주 기법을 이용하여 안정된 서비스를 제공할 수 있는 고정 호스트들에게 부하를 분산시킴으로서 이동 호스트의 부하를 줄일 수 있었다. 즉, 파일 접근을 요구했을 때 파일 이주의 여부를 결정하는 동적인 파일 이주 기법을 제시하였다. 파일 이주는 파일을 소유하고 있는 서버에서 다른 서버로 복사하고 자신의 것은 삭제하는 일련의 프로세스이다. 이러한 파일 이주 기법은 언제, 어느 파일을, 그리고 어디에 이주시킬 것인지가 문제이다. 본 논문은 이동 컴퓨팅 환경에서 파일 이주 기법을 위한 고려 사항 중에서 파일의 크기, 특정 시간에 이동 호스트가 특정 셀에 머무르는 시간 구역성(temporal locality), 이동 호스트의 이동율(mobility rate), 그리고 파일 접근 빈도(file access frequency)를 고려하여 파일 이주 기준을 수립하였다. 마지막으로 우리는 이동 호스트가 요청하는 파일을 이동 지원국, 이동 호스트까지 전달하는 비용과 이동 셀의 서버로 파일을 이주하여 서비스하는 비용을 비교하여 이주 기준에 따라 성능을 분석한다.

본 논문의 2장에서는 본 논문과 관련된 연구를 고찰하고, 3장에서는 파일 이주 시스템 모델을 제시한다. 4장은 이주 기준에 의거한 파일 이주 기법을 제안하고 5장은 시뮬레이션을 통한 평가 결과를 기술한다. 6장은 제안된 기법을 평가하고 앞으로의 연구 방향을 제시하기로 한다.

2. 관련연구

이동 호스트들은 대규모로 변화하는 환경아래서 사용

된다. 자주 변화하는 조건에서 이동 호스트가 효율적으로 운영되기 위해서 이동 호스트는 환경에 잘 적응해 나가야 한다. 이동 컴퓨팅 환경에서의 자원 관리는 고정 네트워크 상에서는 분산 시스템에서와 같이 동일하게 관리되지만 무선 네트워크 상에서는 많은 제약 조건이 따른다. 전통적인 분산 시스템에서는 파일 배치(file assignment), 파일 할당(file allocation), 파일 이주, 파일 복제, 그리고 프로세스 이주(process migration)를 통하여 모든 가용한 서버로 부하를 분산시켜 자원의 가용성을 높이는 연구가 있었으나 이동 컴퓨팅 환경에서의 연구는 그리 많지 않은 형편이다.

분산 시스템에서 부하 공유 및 균등에 대한 연구는 부하가 많은 서버에서 적은 서버로 프로세스를 이주하는 기법을 통하여 수행되었다[3]. 반면, 이동 컴퓨팅 환경에서 Pope는 이동 호스트의 연산 집중적 응용 프로그램을 고정 네트워크의 서버 상에서 수행될 수 있도록 이주시킴으로써 이동 호스트의 부하를 감소시키고 서비스 지연을 막을 수 있는 방안을 제시하였다[4]. 이러한 응용 프로그램 이주 설계에 대한 연구는 시스템 성능, 호스트의 이질성, 그리고 호스트의 이동성에 대한 문제점을 완전히 해결하지는 못하였다.

분산 시스템에서의 파일 배치와 파일 할당은 서버들 사이에서 파일을 효율적으로 배치, 할당하는 문제이다. 파일 할당 기법은 주어진 파일 접근에 대한 통계 값, 평가 빈도 등을 축적하여 파일 배정 방법을 찾아내고 비용 함수가 최적화 되도록 서버에 대하여 파일 배정을 효율적으로 하는 기법이다[5][6][7]. [8]의 연구에서는 이러한 파일 할당 기법 중에서 중앙 집중적 제어 환경에서 이미 알려진 도착 비율을 이용하여 시간 t에서 파일 할당을 결정하는 동적 파일 할당 기법을 연구하였다. 파일 이주는 파일 할당 문제에 대한 확장으로 볼 수 있으며, 파일 이주와 복제를 위한 알고리즘은 이주 및 복제의 결정을 위하여 높은 연산 복잡도가 있는 장기 정책(long-term policy)이 주로 연구되었다. 그러나 장기 정책은 현재 시점의 시스템 상태 정보를 이용하는 단기 정책(short-term policy)보다 적응성이 떨어지는 단점을 보였다[9][10]. Hac은 분산 알고리즘을 이용하여 파일 접근 빈도와 파일 크기에 따라 파일 복제, 파일 이주 그리고 프로세스 이주를 연구하였다[11]. Hurley는 파일 이주와 복제가 함께 이루어지며 파일 복제를 파일 이주의 확장으로 보고 파일 이주 기법과 더불어 정적/동적 파일 복제 기법을 개발하였다[12]. 결국 파일 배치, 할당, 이주 그리고 복제는 일련의 파일들의 가용성과 부하 균등을 위한 작업이다. 파일 이주 문제에서 발생할 수 있는 저장공간에 대한 연구[13]에서는 저장 공간이 유한한 서버에서 제한된 수의 파일만을 저장할 수밖에 없는 단점을 해결하기 위하여 파일 요청에 대한 총 응답 시간이 감소되도록 이주할 서버를 선택하는 기법을 사용하였다. 파일 이주 기법의 성능에 대한 연구는 시간의 변화에 따라 파일 접근 확률을 모델링하고 파일 요청에 대한 총 응답 시간이 낮아질 때 파일 이주를 수행하는 기법에 대한 연구가 있었다[14][15].

이동 컴퓨팅과 같은 위치 독립적 환경은 기존의 분산 환경과는 달리 무선 환경, 이동성, 그리고 자원의 취약성 등이 고려사항이다. 이러한 이동 컴퓨팅 시스템의 성능 향상을 위한 연구는 지금까지 캐싱 기법에 대한 것이 대부분이었다. 이동 컴퓨팅 환경의 캐싱 기법은 통신이 단절되거나 취약한 자원을 갖는 이동 호스트의 작업 부하(파일 접근 빈도 등)로 인하여 통신 연결이 안정적이고 자원이 풍부한 분산 시스템의 기법과는 상이할 것이다. 따라서 사용자의 파일 요구 형태에 대한 정보가 캐쉬 관리 기법에 지대한 영향을 줄 것이다[16][2].

클라이언트가 접근할 가능성 있는 파일들을 예측하여 사전에 캐싱 하는 것은 성능 향상에 긍정적 효과를 거둘 것이다. 특히 이 기법은 이동 컴퓨팅 환경에서 이동 호스트의 접속 상태가 양호하지 못하거나 단절되었을 경우 매우 효율적 기법이다. 캐싱 기법 중에서도 접속이 양호한 상태에서 사용 가능성이 있는 파일을 사전에 캐싱하는 방법에는 선인출(prefetching)과 선저장(hoarding)이 있다[17]. 선인출 기법은 클라이언트가 특정 파일에 접근할 때 파일 전송 지연을 느낄 수 없도록 접근 가능성이 있는 파일들을 사전에 캐싱 하는 것이다[18]. 이 기법은 예측이 잘못되었을 경우 네트워크 대역폭을 낭비할 가능성이 높은 단점이 있다[1]. 선저장 기법은 접속이 정상적 일 때 접속 단절을 대비하여 클라이언트에게 필요한 파일을 미리 저장하는 작업으로서 선인출 기법과는 달리 파일의 가용성(availability)을 높이는 방법이다.

이동 컴퓨팅에서 이동 호스트는 네트워크에 접속 중 위치를 변경하는 일이 자주 발생한다[19]. 이동 호스트는 이동 중에 정보를 접근하는데 있어 자신이 있는 위치에서 가장 가까운 서버에 접속하여 정보를 취득하려 할 것이다[1]. 따라서, 본 논문은 이동 호스트의 파일 접근 지연을 최소화하고 캐싱에서의 최신성 문제로 발생하는 오버헤드(overhead)를 줄이기 위하여 기존 분산 시스템에서 파일 이주 기준으로 많이 연구되었던 파일 접근 빈도와 이동 컴퓨팅 환경의 이동성을 고려하여 이동 호스트에 대한 서비스 지연을 최소화함으로써 이

동 컴퓨팅 시스템의 성능을 향상시킬 수 있는 파일 이주 기법을 제안하고자 한다.

3. 시스템모델

<그림 1>에서와 같은 제안된 시스템 모델은 크게 유선 망인 고정 네트워크와 무선 망인 이동 네트워크로 구성되어 있다. 고정 네트워크에는 다수개의 서버와 이동 지원국이 존재한다. 하나의 이동 지원국은 한 셀을 관리한다. 무선 네트워크에는 셀들이 있고 이동 호스트들이 셀 내에 존재한다. 원격지 서버 집합 내에는 다수의 파일이 있는 것으로 가정한다. 파일 이주는 서버들 사이에서만 발생하고 서버와 이동 지원국, 이동 호스트 간에는 이주가 이루어지지 않는다. 이동 호스트는 무선 연결을 통하여 이동 지원국까지, 유선 망을 통하여 서버로 파일을 요청한다. 특정 시간 t에 서버는 파일 이주 기준에 따라 파일을 이동 호스트로 전송하든지 이동 호스트가 있는 셀의 서버로 파일을 이주하게 된다.

시스템 모델에 대한 가정은 다음과 같다.

- 파일 이주와 관련된 모든 시스템 상태는 특정 시간 t시점의 정보이며 서버간에 파일의 공유는 없는 것으로 간주한다.
- 고정 네트워크는 접속 단절(disconnection)이 없는 것으로 가정한다.
- 고정 네트워크의 서버들은 파일들의 사본(replica)을 갖지 않는다.
- 이동 호스트는 개인용 휴대 컴퓨터로 디스크가 없이 제한된 자원을 갖는 PDA와 같은 호스트이다.
- 특정 셀에 체류하는 이동호스트들은 하나의 서버로부터 서비스를 받는다.
- 이동호스트가 체류하는 셀의 이동지원국과 서버간 통신비용보다 서버들 간의 통신비용은 크다고 가정한다.
- 체류 셀의 이동지원국과 원격지 서버간의 통신비용은 이동호스트가 체류하는 셀의 이동지원국과 서버 간 통신비용보다 크다고 가정한다.

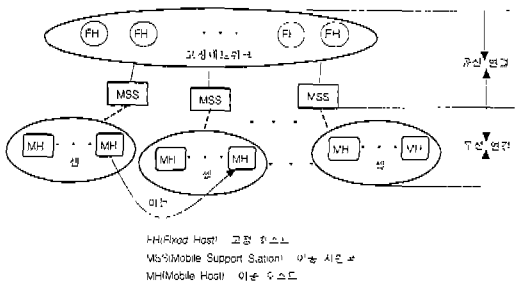

그림 1 시스템 모델

본 논문에서 정의된 표기법은 다음과 같다.

S : 이동 컴퓨팅 시스템의 고정 호스트인 원격지 서버 집합

S_n : n번째 서버

SM : 시스템 상에 위치하는 전체 이동 호스트 집합

mh_l : l번째 이동 호스트

C : 전체 셀(cell)들의 집합

C_k : k번째 셀

MSS : 전체 이동 지원국(mobile support station)들의 집합

MSS_k : k번째 이동 지원국

SF : 원격지 서버들에게 저장된 전체 파일 집합

f_m : 전체 파일 집합 중에서 m번째 파일

$SM(C_k, t)$: 시간 t상태에서 셀 C_k에 체류하는 이동 호스트 집합

$SF(S_n, t)$: 시간 t상태에서 서버 S_n에 존재하는 파일 집합

이와 같은 시스템 모델의 전체 상태를 정형화하면,

$S=\{S_1, S_2, ..., S_n\}$: n개의 *server*

$SM=\{mh_1, ..., mh_l\}$, l개의 이동 호스트

$C=\{C_1, ..., C_k\}$: k개의 *cell*

$MSS=\{MSS_1, MSS_2, \cdots, MSS_k\}$:
k개의 *mobile support station*

$SF=\{f_1, f_2, ..., f_m\}$, m개의 파일

어떤 특정 시간 t에서 특정 셀 내의 이동 호스트의 상태 집합은 다음과 같다.

$$SM(C_1, t) = \{ mh_{11}, \ldots, mh_{1l_1} \}$$
$$\vdots$$
$$SM(C_k, t) = \{ mh_{k1}, \ldots, mh_{kl_k} \}$$

$$l = \sum_{i=1}^{k} l_i : \quad l \text{ 개의 이동 호스트}$$

어떤 시간 t에 각 원격지 서버에 존재하는 파일 집합은 아래와 같다.

$$SF(S_1, t) = \{ f_{11}, \ldots, f_{1m_1} \}$$
$$\vdots$$
$$SF(S_n, t) = \{ f_{n1}, \ldots, f_{nm_n} \}$$

$$m = \sum_{i=1}^{n} m_i : \quad m \text{ 개의 파일}$$

전체 시스템에서 데이타를 전송하는 단위는 대역폭이 안정적인 유선 망과 낮은 대역폭을 갖는 무선 망의 특징을 고려하여 구성 요소간의 전송 단위를 달리한다. 고정 네트워크에서 서로 데이타를 주고받는 서버들은 상

호 간에 전체 파일 단위로 전송한다. 왜냐하면 고정 네트워크는 고속의 유선 망으로 연결되어 있고 네트워크 단절이 없다고 가정했기 때문이다. 반면, 접속 단절이 빈번한 무선 망의 데이타 전송 단위는 블록(block) 단위로 전송한다고 가정한다.

4. 이주 기법 설계

이동 컴퓨팅 환경은 여러 가지 요인으로 인해 파일의 접근 지연을 야기한다. 본 논문에서는 이동 클라이언트가 파일을 요청할 때 이동 컴퓨팅 시스템의 순간 시스템 상태를 파악하고 이동 클라이언트 측의 캐싱 전략보다는 신뢰성 있는 고정 네트워크에서의 파일 이주를 통해 접근 지연을 최소화하여 전체적인 시스템 성능 향상을 유도하였다. 기존 분산 시스템의 파일 이주 요인과 더불어 우리는 이동 시스템에서 나타나는 특징적인 요소들을 이주 기준으로 하여 파일 이주를 결정할 수 있도록 이주 기법을 설계하였다.

4.1 파일 이주 기준

분산 시스템은 파일 이주를 결정하는 기준으로 특정 파일에 대한 크기 및 접근 빈도를 중요하게 다루었다. 본 논문은 파일 접근 빈도와 더불어 이동 시스템에서의 가장 큰 특징인 이동성(mobility) 및 이동 호스트의 지리적 영역인 셀에서 보이는 특정 이동 호스트의 체류 시간(staying time) 등을 고려하여 이주를 결정하고자 한다. 특히 응답 시간을 고려하여 각 기준들이 산출하는 비용에 대한 최적의 임계값(threshold)으로 파일 이주 여부를 동적으로 결정할 수 있어 전체적인 서비스 지연을 배제하고 시스템 성능을 향상시킬 수 있었다.

이동 호스트 mh_i가 서버 S_k에 있는 파일 f_{kj}를 요청하여 전송 받는 응답 시간 RT^i_{kj}와 서버 S_l에 있는 파일 f_{ij}를 서버 S_l으로 이주하는 시간 MT^i_{ij}를 정의한다. 파일 이주를 결정하기 위하여 RT^i_{kj}과 MT^i_{ij}에 관심을 두고 기준이 되는 요인들간에 상관 관계를 고찰하였다. 다음은 본 논문에서 고려한 이동 컴퓨팅 시스템의 파일 이주 기법에 대한 기준을 설명하기로 한다.

ㅇ 파일 접근 빈도(file access frequency)

$\triangle$: 현재시간 t에서 이전 시간 구간 $[t_i, t]$ 즉 $\triangle = t - t_i$

$a_{ij}(t, \triangle)$: 이동 호스트 i가 파일 j를 요청한 시간 t에서 이전 $\triangle$시간 동안 접근한 빈도

$A_{ij}(t_0, t)$: 이동 호스트 i가 파일 j를 최초 요청 시간 t_0에서 현재시간 t까지 접근한 총 누적 빈도

$A^i_{total}(t, \triangle) = \sum_{j=1}^{m} a_{ij}(t, \triangle)$: 이동 호스트 i가 파일 요청을 한 서버의 파일들에 대한 총 접근 빈도

ㅇ 이동율(mobility rate)

$N^i(t, \triangle)$: 시간 t시점에서 이전 $\triangle$시간동안 이동 호스트 i의 핸드오프(handoff) 수

$N^i_{a,b}(t, \triangle)$: 이동 호스트 i가 시간 t시점에서 이전 $\triangle$ 시간동안 셀 a에서 셀 b로 이동한 회수

$N^i(t, \triangle) = \sum_{a=1}^{k}(\sum_{b=1}^{k} N^i_{a,b}(t, \triangle))$:이동 호스트 i의 총 이동 회수(단, $N^i_{a,a}(t, \triangle) = 0$ for *all* C_a를 가정함)

$N^i_{avg}(t, \triangle) = N^i(t, \triangle)/\triangle$: 일정 시간당 이동 호스트 i의 평균 이동율(mobility rate)

$N^i_b(t, \triangle) = \sum_{a=1}^{k} N^i_{a,b}(t, \triangle)$: 시간 t에서 이전 $\triangle$시간 동안 이동 호스트 i가 특정 셀 b로 이동한 회수

ㅇ 이동 호스트가 이동한 셀에 체류한 시간(cell staying time)

$T^i_b(t, \triangle)$: t시점에서 이전 $\triangle$시간동안 이동 호스트 i가 셀 b에 체류한 총 시간

$\sum_{b=1}^{k} T^i_b(t, \triangle) = \triangle$: 이동 호스트 i가 $\triangle$시간동안 각 셀에 머문 시간의 합은 결국 $\triangle$이다.

$T^i_b(t, \triangle)/N^i_b(t, \triangle)$: t시점에서 이전 $\triangle$시간동안 이동 호스트 i가 한번 셀 b에 진입하여 체류한 평균 시간

특정파일 i를 이동 호스트가 요청하였다면 서버는 파일 이주 여부를 결정하기 위하여 이주 기준을 적용한다. 원격지에서 목적지 서버까지 이주하였을 때 소요되는 파일 이주 시간과 이동 호스트가 파일을 요청하여 서비스 받는 응답 시간은 본 논문의 비용 산출의 주요 대상이다. 따라서 파일 이주와 파일 전송에 대한 비용 산정을 위한 요소들을 아래에 기술한다.

M : 파일을 요청할 때 메시지 크기

L_j : 요구받은 파일 j의 크기

α : 파일 전송을 위한 블록 크기(block size)

$k = \frac{L_j}{\alpha}$: 파일 j의 총 블록 수

B_{fixed} : 고정 네트워크의 대역폭(fixed network bandwidth)

$B_{wireless}$: 무선 네트워크의 대역폭(wireless network bandwidth)

RT^i_{kj} : 이동 호스트 mh_i가 서버 S_k에 있는 파일 f_{kj}를 요청하여 응답 받는 시간

MT^i_{ij} : 서버 S_i에 있는 파일 f_{ij}를 서버 S_l로 이주하는 시간

$C = [c_{ij}]$, $1 \le i \le k$, $1 \le j \le n$: 이동 지원국 MSS_i에

서 서버 S_j까지의 통신비용

$D=[d_{ij}]$, $1 \le i \le n$, $1 \le j \le n$ (단, $i=j$ 이면 $d_{ij}=0$):

서버 S_i에서 S_j까지의 통신비용

응답 시간은 이동 호스트가 파일을 요청할 때 서버까지의 메시지 전송 시간과 서버가 이동 호스트에게 파일을 전송하는데 걸리는 파일 전송 시간을 포함한 개념이다. 이주 시간은 서버간의 파일만을 전송하는데 걸리는 시간이다. 이때 고려해야할 인자가 각 네트워크가 가지고 있는 대역폭과 메시지 및 파일의 크기이며 특히 우리가 가정한 전송 단위인 전체 파일 전송 단위 및 블록 전송 단위를 전송 단계 별로 적용해야 한다.

다음은 위의 산출 인자를 기준으로 응답시간과 이주 시간을 산출하는 식을 보여 준다.

이동 호스트의 메시지는 파일을 요청하는 서버까지 무선 네트워크와 고정 네트워크를 통해 전달되므로 각 네트워크의 대역폭으로 메시지 크기를 나누면 전송 시간이 산출된다.

$$\text{메시지 전송 시간 : } \frac{M}{B_{fixed}} + \frac{M}{B_{wireless}}$$

이동 호스트로부터 파일을 요청 받은 서버가 이주가 아닌 전송을 결정했다면 서버는 해당 이동 지원국까지 전체 파일을, 이동 지원국은 이동 호스트까지 블록 단위를 전송한다. 따라서 서버는 이동 지원국까지 전체 파일을 한 번 전송하고 이동 지원국은 크기가 L_j인 파일을 이동 호스트까지 k번 전송되게 된다. 따라서 파일 전송 시간은 다음과 정의한다.

$$\text{파일 전송 시간 : } \frac{L_j}{B_{fixed}} + \frac{k \cdot \alpha}{B_{wireless}}$$

응답 시간은 메시지 전송 시간과 파일 전송 시간을 합하여 산출할 수 있다.

$$\text{응답 시간 : } RT_{kl}^{j} = \frac{L_j + M}{B_{fixed}} + \frac{k \cdot \alpha + M}{B_{wireless}}$$

이주 시간은 특정 파일을 이주하기로 결정했을 때 고정 네트워크를 통하여 서버와 서버간에 전체 파일 단위로 전송하므로 대역폭으로 파일 크기를 나누어 산출할 수 있다.

$$\text{이주 시간 : } MT_{ij}^{l} = \frac{L_j}{B_{fixed}}$$

4.2 이주기법

본 논문에서 제시하고자 하는 기법은 3가지 경우를 고려하여 파일 이주 여부를 결정한다. 이러한 기법들의 평가는 이주하였을 때와 이주하지 않았을 때의 비용을 산출하여 비교할 수 있다. 비용 모델은 파일 접근 빈도와 이동성에 따라 이주 및 비 이주 기법의 상대적 비용을 고찰하였다. 아래 <그림 2>는 총비용을 산출하는데 필요한 모델을 보여준다.

셀 C_a에 체류하는 이동 호스트 mh_i가 이동 지원국 MSS_a에게 파일 f_j를 요구할 때 이 파일을 요청 받은 서버 S_k는 이주 결정 기준에 의거 이주 또는 비 이주를 결정하게 된다. 이때 이주할 때와 이주하지 않았을 때 비용을 산출하는 식은 다음과 같다(단, $d_{kl} > c_{al}$, $c_{ak} > c_{al}$).

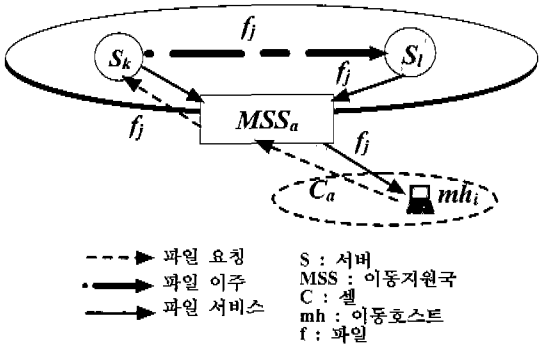

그림 2 비용 모델

총 파일 이주 기법 비용($COST_{FM}$) :

$$COST_{FM} = \frac{L_j + M}{B_{fixed}} \cdot d_{kl} \cdot N'(t, \wedge) + \left(\frac{L_j + M}{B_{fixed}} \cdot c_{al} + \frac{k \cdot \alpha + M}{B_{wireless}} \right) \cdot a_{ij}(t, \triangle)$$

총 파일 비 이주 기법 비용($COST_{NM}$) :

$$COST_{NM} = \left(\frac{L_j + M}{B_{fixed}} \cdot c_{ak} + \frac{k \cdot \alpha + M}{B_{wireless}} \right) \cdot a_{ij}(t, \triangle)$$

첫 번째 기법은 파일 접근 빈도에 따르는 응답 시간을 비교하여 이주를 결정한다.

시간 t시점에 이동 호스트로부터 파일을 요청 받은 서버는 이전 △시간동안 파일 접근 빈도를 계산한다. 만일 서버가 이동 호스트로 직접 파일을 전송하는데 소요되는 총 응답 시간 보다 파일을 이동 셀의 서버로 이주하여 요청에 응답하는 총 시간이 적고 파일 접근 빈도가 임계값 이상이면 파일을 이주한다. 임계값은 이주할 때보다 이주하지 않았을 때 응답 시간이 커지는 시점의 파일 접근 빈도이다.

파일 접근 빈도는 이동 호스트가 파일을 요청할 때마다 빈도를 하나씩 증가시켜 서버에서 유지하고 서버는 현재 시점 t에서 이전 △시간동안 누적된 빈도를 가지고 응답 시간을 계산한다. 총 빈도에서 △시간 구간의 시점

t_i까지의 빈도를 빼면 시간 t에서 이전 △시간 동안의 접근 빈도를 구할 수 있다.

$$a_{ij}(t, \triangle) = A_{ij}(t_0, t) - A_{ij}(t_0, t_i)$$

아래의 식은 파일 접근 빈도를 이용한 파일 이주 결정 식이다.

$a_{ij}(t, \triangle)RT^i_{kj} \geq MT^i_{kj} + a_{ij}(t, \triangle)RT^i_{ij}$ & $a_{ij}(t, \triangle) > \Phi$

. Φ는 임계값

이 기법의 비용은 파일 접근 빈도, 통신 비용, 파일 전송 단위, 파일 전송 비용, 파일 이주 비용에 의해 산출할 수 있다.

두 번째 기법은 파일 접근 빈도와 이동성의 상대 비율에 따라 이주를 결정한다.

이동 호스트가 파일을 요청하고 다른 셀으로 자주 이동할지라도 파일 접근 빈도에 비해 이동성이 상대적으로 높으면 파일 요청을 받은 서버가 이동 셀의 서버로 파일을 빈번히 이주시키는 것은 파일 전송 비용보다 이주 비용이 월등히 많이 소요된다. 따라서 이동성이 많은 호스트가 파일을 요청할 때 서버는 파일을 이주하는 것보다 파일을 전송하는 것이 더 효율적이다. 이 기법은 파일 접근 빈도와 이동성의 상대 비율이 임계값 이상으로 크면 파일을 이동 셀의 서버로 이주한다.

$$\Lambda = a_{ij}(t, \triangle)/N^i_{avg}(t, \triangle) > \xi; \xi\text{는 임계값}$$

파일 이주가 없을 때 비용은 이동 호스트가 이동하면서 서버 S_k 에 존재하는 파일 f_j를 접근하여 전송 받는 비용이다. 파일 이주가 있을 때 비용은 이동셀의 서버로 파일을 이주하는 비용과 이동 지원국에서 파일을 요청한 호스트로 전송하는 비용이다.

세 번째 기법은 이동 호스트의 이동율 대비 셀 체류 시간에 의한 파일 이주 결정이다.

일반적으로 이동 호스트는 셀을 이동하면서 특정 셀에 대해 체류하는 시간이 많을 수도 있다. 이동 호스트가 특정 시간 구간 동안 여러 셀을 빠르게 이동하면서 그 중에서 특정한 한 셀에 오래 머무른다면 파일을 이주하는 것이 비용이 적게 들것이다. 이동 호스트의 이동율이 임계값을 넘을지라도 현재 t시점에서 이전 △시간 동안 이동 호스트 i가 이동 셀 b에 체류한 평균 시간이 길어지면 서버는 파일을 이주하는 것이 효율적이다.

$$\Pi = T^i_b(t, \triangle)/N^i_{avg}(t, \triangle) > \theta \ (\theta\text{는 임계값})$$

5. 평가 실험결과

5.1 평가모델

파일 이주 기법의 성능 평가를 위한 시뮬레이션은 [20]에서 소개하고 있는 임의의 수(random number) 생성 알고리즘과 시드(seed)를 사용하고 FORTRAN 기반의 시뮬레이션 언어인 SIMLIB을 C++로 구현하였다. SIMLIB에서 제시하는 시뮬레이션 과정에 따라 초기화, 이벤트 스케줄링 및 평가 데이타 계산 등을 반복하여 생성되는 평균값에 대한 조건을 변화해 가면서 시뮬레이션을 수회 실시하여 결과를 산출하였다.

본 논문은 이동 컴퓨팅 환경에서 파일 이주를 이용하여 파일 접근 지연을 최소화시킬 수 있는지를 시뮬레이션하기 위하여 5개의 셀을 가정한다. 이동 호스트는 임의의 이동성을 갖고 인접한 셀로 핸드오프를 할 수 있으며, 셀 체류 시간 동안에 임의의 간격으로 파일 서비스를 요청할 수 있다. 각 셀 내에는 이동 호스트를 지원하는 이동 지원국과 파일 서비스를 제공하는 서버가 하나씩 존재하고, 서버는 FIFO 큐를 포함한다. 파일 이주 또는 접근을 위해 셀간에 구성된 고정네트워크는 대상 셀을 중심으로 인접한 셀들의 서버가 완전히 연결된 모델로 가정한다. 실험의 목표는 이주와 비 이주 기법을 적용하였을 때 비용을 비교하여 가장 최적의 비용을 나타내는 상태가 어떤 값들을 갖는지 조사한다.

시뮬레이션은 이동 호스트의 도착 시간(arrival time), 출발 시간(departure time), 파일을 서비스하고 종료하는 사건들, 이동 호스트의 이동성 빈도, 파일 접근 빈도, 파일 크기 등을 고려하였다. 분포는 어떤 특정 기간(△) 동안 발생하는 지수 분포를 가지고 시뮬레이션 하였다. 우리의 실험에서 이동 빈도는 단위 시간당 셀 체류 시간의 평균 비율로 산출하였고 파일 크기는 1 블록(512바이트)을 단위로 하여 최소 크기에서 최대 파일 크기가 될 때까지 임의의 수를 사용하였다. 대역폭은 유선 대역폭과 무선 대역폭을 일정하게 각각 10Mbps, 2Mbps로 놓고 실험하였다.

이벤트는 셀에 이동 호스트가 도착하는 시간과 셀을 떠나는 시간, 그리고 이동 호스트의 파일 요청에 따른 파일 서비스 시작 시간과 종료 시간을 기준으로 이벤트를 처리하였다. 우리의 시뮬레이션에서 시뮬레이션 시간은 1440분, 즉 하루를 가정했다. 통신에러는 무시하고 메시지 크기는 512 Kbyte로 정하여 실험하였다. 시뮬레이션 방법은 동일한 시드에 의해 생성된 임의 수를 통한 시뮬레이션을 5회 수행한 평균값을 임계값에 대한 1회 실험 결과로 하고, 임계값과 시드의 변화에 대한 실험을 5회 실시하였다.

우리는 이동 호스트의 이동성과 접근 빈도를 이용하여 파일 이주와 파일 비 이주 기법을 동적으로 적용함으로써 파일 서비스 지연을 최소화하고자 한다.

<표 1>은 실험에서 고려한 평가 요소를 나타낸다.

표 1 실험 평가 요소

실험인자 (Parameter)	설명	값
유선 대역폭	고정 네트워크 상의 대역폭	10 Mbps
무선 대역폭	무선 네트워크 상의 대역폭	2 Mbps
실험 시간	시뮬레이션 시간 구간	24 시간
실험 회수	시뮬레이션 회수	5 회
이동 호스트 도착 시간	이동 호스트의 셀 도착 시간 간격	10 분마다
메시지 크기	이동 호스트의 파일 요청 메시지 크기	512 Kbyte
파일 크기	이동 호스트가 요청하는 파일 크기의 가변요소	확률변수
셀 체류 시간	이동 호스트가 셀에 체류하는 평균 시간	확률변수
파일 접근 간격	서버내의 특정 파일을 요청하는 시간 간격	확률변수

5.2 실험 및 결과

본 논문의 실험 결과는 이동성, 파일 크기, 그리고 파일 접근 빈도에 따라 이주 기법의 비용과 비 이주 기법의 비용을 비교 분석함으로써 이동성이 파일 접근 지연에 미치는 영향과 파일 크기, 파일 접근 빈도의 변화에 따른 성능을 관찰할 수 있었다.

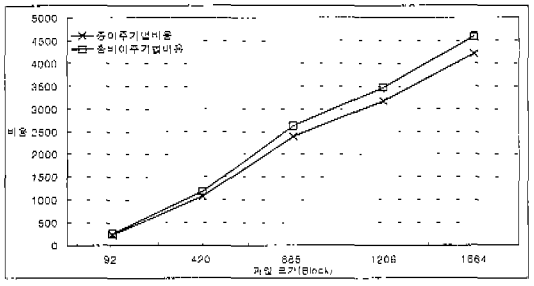

그림 3 파일 크기 증가에 따른 비용 비교

<그림 3>에서는 파일 크기의 증가에 따른 기법들의 비용 변화를 예시한다. 지수 분포(exponential distribution)에 따라 증가하는 파일 크기에 대한 비용 변화만을 획득하기 위하여 비용 모델에 포함된 이동 빈도 및 파일 접근 빈도 등의 매개변수는 균일 분포(uniform distribution)를 따른다. 이동 호스트가 요청하는 파일 크기 상승하면 파일 이주 비용과 비 이주 비용 모두 상승하면서 비용이 같은 비율로 증가하거나 감소하지만 이주 기법 비용이 항상 적게 들며 비용 격차도 함께 증가함을 알 수 있다. 이것은 요청한 파일 크기가 증가할 경우 파일 서비스를 제공할 때마다 유선 네트워크에서는 전체 파일, 그리고 무선 네트워크에서는 블록 단위로 여러 번 전송하는데 따르는 통신 비용이 모두 증가하기 때문이다. 즉 파일 크기가 증가하면 할수록 고정 망에서 한번의 이주를 통하여 이동 호스트의 파일 접근 지연을 최소화시킬 수 있는 이주 기법이 시스템 성능에 장점이 있음을 보여준다.

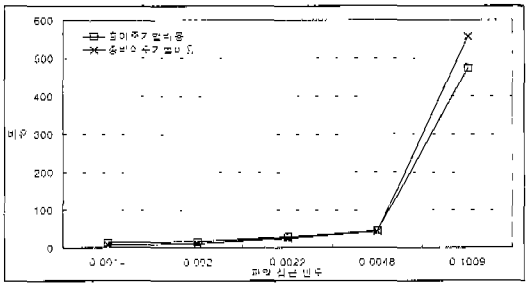

그림 4 파일 접근 빈도에 따른 비용 비교

파일 접근 빈도는 이동 호스트가 단위 시간당(△시간) 특정 파일을 접근하는 평균 빈도를 의미한다. <그림 4>와 같은 실험 결과는 이동 호스트의 파일 접근 빈도가 증가하면 이주 기법보다 비 이주 기법에 소요되는 비용이 근소하게 적은 비용이 들지만 어떤 임계값(ϕ)이 되었을 때 이주 기법의 비용이 급격하게 증가함을 관찰할 수 있다. 실험결과에서는 ϕ=0.0048일 경우 비 이주 기법과 이주 기법의 비용 교차가 생기므로 이를 임계값으로 설정할 수 있다. 파일 접근 빈도가 커질 때 유무선 망의 통신 비용이 모두 증가하므로 우리의 기법에선 $\frac{a_{ij}(t, \triangle)}{\triangle} = \phi$가 되는 순간 이동 셀의 서버로 요청한 파일을 이주함으로써 이동 호스트의 파일 접근 시간 지연을 줄일 수 있다.

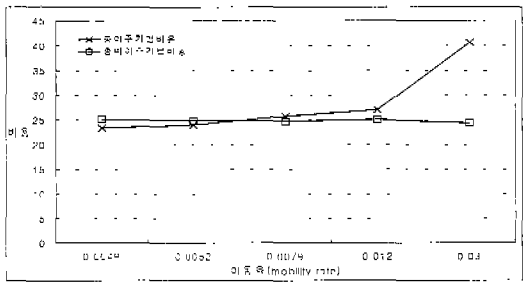

그림 5 이동성 증가에 따른 비용 비교

이동율은 이동 호스트가 단위 시간당(Δ) 다른 셀로 이동하는 회수, 즉 핸드오프(handoff)를 평균한 것을 말

한다. 이동성이 많아지면 이주 기법은 이동 셀의 서버로 요청한 파일을 빈번하게 이주하므로 통신 트래픽의 증가와 함께 비용이 증가하게 된다. 반면, 비 이주 기법은 파일을 요청했을 때 항상 자신의 서버로부터 파일 서비스를 받기 때문에 이동성에 별로 영향을 받지 않게 된다. <그림 5>에서는 이동성과 관련된 비용을 나타낸다. 이동성이 작을 수록 이주 기법은 비용이 적게 들며 이동성이 많을수록 비 이주 기법이 유리함을 알 수 있다. 이주 기법 비용은 이동성 평균 비율이 0.006이 되었을 때 이주기법 보다 급격히 비용이 증가하므로 임계값은 η =0.006으로 평가된다. 이러한 결과로 미루어 $N^{i}_{avg}(t, \triangle) > \eta$ (임계값)이 되었을 때 파일을 이주하지 않는 것이 이동 호스트의 파일 접근 비용을 줄일 수 있다.

또 하나의 고려 사항은 이동성과 파일 접근 빈도의 상관 관계라 할 수 있다. 앞에서 이주 기법에서 언급하였듯이 파일 접근 빈도가 많아지고 이동성이 적다면 파일을 이동 셀의 서버로 이주하는 것이 바람직하고 반대의 경우라면 이주하지 않는 것이 서비스 지연을 방지할 수 있을 것이다. <그림 6>은 이것을 뒷받침하는 실험 결과이다. 즉 앞에서 정의한 $\Lambda = a_{ij}(t, \triangle)/N^{i}_{avg}(t, \triangle) > \xi$ 가 되었을 때 파일 이주를 결정하는 것이 비용이 적게 들것이다. 본 실험 결과에서는 파일 접근 비율 대비 이동율이 1.61이 되었을 때 비 이주 기법 비용이 서서히 증가하다가 1.90 시점부터 급격히 비용이 증가함을 알 수 있다. 따라서 상대 비율을 앞에서 ξ로 정의하였으므로 ξ=1.61이 파일 이주를 결정할 수 있는 임계값으로 관찰할 수 있었다.

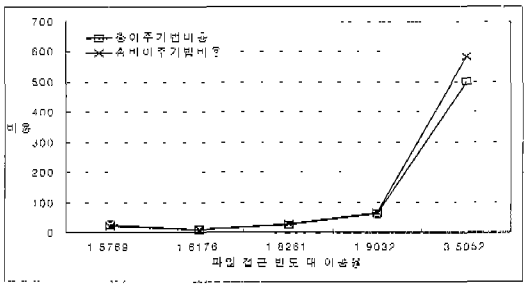

그림 6 파일 접근 빈도 대 이동율(Λ)

세 번째 기법에서 고려되었던 이동율 대비 셀 체류시간의 상대비율에 따른 기법간의 비용 비교를 <그림 7>의 실험 결과에서 관찰할 수 있었다. 시간 구간 △시간 동안 이동 호스트가 특정 셀에 진입하여 체류한 평균 시간이 이동율 보다 커지게 되면 체류 셀의 서버로 요청한 파일들을 이주하는 것이 파일 접근 지연을 막을 수 있는 한 방안이 될 것이다. <그림 7>에서 보여주듯이 $\Pi = T^{i}_{h}(t, \triangle)/N^{i}_{avg}(t, \triangle) > \theta$가 되었을 때 파일을 이주하는 것이 비용이 적게 소요됨을 알 수 있다. 즉, 이동 호스트의 이동율 보다 특정 셀에 진입하여 체류한 시간이 상대적으로 길어지면 원격지 서버에 접속하여 파일을 접근하는 비용이 많이 들게 된다는 결론을 관찰할 수 있었다.

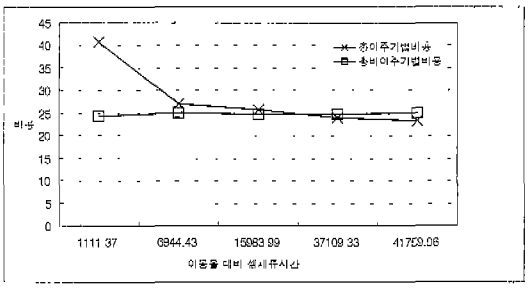

그림 7 이동율 대비 셀 체류시간의 상대 비율(Π)

실험 결과에서 이주 및 비 이주 기법 모두가 파일 크기 및 파일 접근 빈도가 증가함에 따라 비용이 다 같이 증가하며 비 이주 기법 비용이 더 크게 비용이 증가됨을 관찰할 수 있었다. 따라서, 이동 호스트의 요청 파일이 크면 클수록, 파일 접근 빈도가 임계값 ϕ이상일 때 이주 기법을 적용하는 것이 이동 호스트의 파일 접근 지연을 최소화에 기여할 수 있었다. 이동성은 비 이주 기법에는 큰 영향을 못 미치지만 이동할 때마다 파일을 이주하는 것은 이동성이 증가할수록 비용이 급격히 증가하기 때문에 이동성이 클 경우는 비 이주 기법이 좋음을 알 수 있다. 따라서 이동성이 임계값(η)에 도달했을 때 이주를 결정하는 것이 효율적이다.

본 실험 결과는 파일 접근 비율 대비 이동율의 상대비율(Λ)에 따른 이주기준도 도출할 수 있었다. 상대 비율이 증가하면서 비 이주 기법 비용이 서서히 증가하다가 1.90 시점부터 급격히 비용이 증가함을 알 수 있었다. 따라서 상대비율(Λ)이 임계값 ξ에 도달하면 파일 이주를 결정하여야 한다는 것을 알 수 있다. 또한, 이동율 대비 셀 체류시간의 상대비율(Π)에 대한 실험 결과에서 Π가 증가되면, 즉 특정 셀에 체류하는 시간이 길어지면 이주 기법의 비용이 더 적어짐을 관찰할 수 있었다. 따라서, $\Pi > \theta$가 되는 시점에 이주를 결정하는

것이 효율적이라는 것을 확인할 수 있었다.

6. 결 론

본 논문에서는 파일 이주 기법을 이용하여 안정적 서비스를 할 수 있는 고정 호스트들에게 부하를 분산시켜 이동 호스트의 부하를 줄이고 자원이 제한적인 호스트들이 요구하는 파일을 서버에서 이동 호스트가 체류하는 셀의 서버로 이주하는 기법을 사용하였다. 이동 컴퓨팅 환경에서 우리는 파일 이주 기법을 위한 고려 사항으로 파일의 크기, 셀 체류 시간, 특정 시간에 이동 호스트가 특정 셀에 머무르는 시간 구역성(locality), 이동 호스트의 이동율, 그리고 파일 접근 빈도를 이주 기준으로 하였다. 이러한 기준에 의거 파일을 요구하는 시점에 파일 이주 여부를 결정하는 동적인 파일 이주 기법을 제시하였다.

우리는 이동 호스트가 요청하는 파일을 이동 지원국, 이동 호스트까지 전달하는 비용과 이동 셀의 서버로 파일을 이주하여 서비스하는 비용을 비교하여 이주 기준에 따라 성능을 분석하였다. 실험 결과에서 고찰했듯이 이동 컴퓨팅 환경에서는 기존의 분산 시스템 환경과는 달리 파일 접근 빈도 외에도 이동성에 따른 파일 접근 지연이 발생할 수 있다는 것을 알 수 있었다. 따라서 본 논문은 이동성 문제를 고려한 파일 이주 기법을 통하여 캐싱을 하기 어려운 자원이 제한적인 이동 호스트들의 파일 접근 지연을 줄일 수 있는 이주 기법에 대한 연구를 하였다.

앞으로의 연구 방향은 고정 네트워크 상의 분산 시스템과 무선 네트워크에 있는 이동 호스트들 사이에 파일 접근 전략에 있어 파일 복제와 캐싱 전략에 대한 파일 이주 기법과의 연동성을 찾아내는데 주안점을 두고 연구하여야 할 것이다.

참 고 문 헌

[1] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, vol. 27 , No. 4, Apr. 1994.

[2] Laura Bright and Louiqa Raschid, "Efficient Remote Data Access in a Mobile Computing Environment," International Workshops on Parallel Processing, pp. 57-64, 2000.

[3] James Griffioen, Todd A. Anderson and Yuri Breitbart, "A Dynamic Migration Algorithm for a Distributed Memory-Based File Management System," Proceedings of the IEEE Seventh International Workshop on Research Issues in Data Engineering, pp. 151-160, 1997.

[4] S. Pope, "Application Migration for Mobile Computers," Proceedings of 3'rd International Workshop on Services in Distributed and Networked Environments, pp. 20-26, 1996.

[5] K.D. Levin, " Adaptive Structuring of Distributed Databases," Proceedings of the National Computer Conference, pp. 691-696, Jun. 1982.

[6] W. W. Chu, " Optimal File Allocation in a Multiple Computer System," IEEE Transactions on Computer, C-18, 10, pp. 885-889, Oct. 1969.

[7] R. G. Casey, " Allocation of Copies of a File in a Information Network," AFIPS SJCC, 40, pp. 617-625, 1972.

[8] A. Segall, "Dynamic File Assignment in a Computer Network," IEEE Transactions on Automatic Control, AC-21,2, pp. 161-173, Apr. 1976.

[9] Q. Kure, "Optimization of File Migration in Distributed Systems," PhD Thesis, Univ. of California, Berkeley, 1988.

[10] O.R. Lui Sheng, "Models for Dynamic File Migration in Distributed Systems," PhD Thesis, Univ. of Rochester, 1986.

[11] A. Hac, "A Distributed Algorithm for Performance Improvement Through File Replication, File Migration, and Process Migration," IEEE Trans. Software Engineering, vol. 15, no. 11, pp. 1,459-1,470, 1990.

[12] R.T. Hurley, "File Migration and File Replication : A Symbiotic Relationship," IEEE Trans. on Parallel and Distributed Systems, vol. 7, no. 6, pp. 578-587, June 1996.

[13] R.T. Hurley, J.P. Black, and J.W. Wong, "Limited Effects of Finite Storage on a Beneficial File Migration Policy," Proc. 19th IEEE Conf. Local Computer Networks, pp. 432-439, Oct. 1994.

[14] R.T. Hurley, S.A. Yeap, J.W. Wong and J.P. Black, "Potential Benefits of File Migration in a Heterogeneous Distributed File System," Proc. ICCI '93 : Fifth Int'l Conf. Computing and Information, pp. 123-127, May 1993.

[15] R.T. Hurley, J.P. Black, and J.W. Wong, "Performance of File Migration in a Distributed File Systems," Proc. 10th International Conference on Computer Communications, pp. 685-692, Nov. 1990.

[16] S. Khurana, A. Kahol, S. K. S. Gupta, and P. K. Srimani, "An Efficient Cache Maintenance Scheme for Mobile Environment," Proceedings of the 20th International Conference on Distributed Computing Systems, pp. 530-537, 2000.

[17] K. W. Froese, "File Cache Management for Mobile Computing," MS Thesis, Dept. of Computer Science, Univ. of Saskatchewan, Canada, 1996.
[18] T. M. Kroeger and D. D. E. Long," Predicting File System Actions from Prior Events," USENIX Conference Proceedings, pp. 319-328, Jan. 1996.
[19] Subhashini Rajagopalan and B. R. Badrinath, "An Adaptive Location Management Strategy for Mobile IP," Proceedings of the 1'st annual International Conference on Mobile Computing and Networking, pp. 170-180, 1995.
[20] Averill M. Law and W. David Kelton, Simulation Modeling and Analysis, Second edition, McGraw-Hill, 1991

한 문 석

1986년 중앙대학교 전자계산학과 학사. 1988년 중앙대학교 전자계산학과 석사. 1998년 성균관대학교 정보공학과 박사과정 수료. 1996년 ~ 현재 국립 원주대학 부교수. 관심분야는 이동 컴퓨팅 시스템, 이동 컴퓨팅 보안, 분산 시스템

박 상 윤

1997년 동국대학교 전자계산학과 학사. 1999년 성균관대학교 전기전자 및 컴퓨터공학부 석사. 2001년 성균관대학교 전기전자 및 컴퓨터공학부 박사과정 수료. 관심분야는 이동 컴퓨팅 시스템, 시스템/이동 컴퓨팅 보안, 분산 시스템

엄 영 익

1983년 서울대학교 계산통계학과 졸업(학사). 1985년 서울대학교 대학원 전산과학전공(석사). 1991년 서울대학교 대학원 전산과학전공(박사). 1983년 ~ 1986년 서울대학교 도서관 전산화준비실. 1986년 ~ 1993년 단국대학교 전자계산학과 부교수. 1993년 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 교수. 관심분야는 분산 시스템, 이동 컴퓨팅, 분산 객체 시스템, 운영체제