

멀티 매니징 기법을 이용한 웹기반 분산 병렬 컴퓨팅 환경

맹 헤 선[†] · 한 탁 돈^{††} · 김 신 덕^{†††}

요 약

자바 언어를 이용하여 이질적인 컴퓨팅 자원으로 구성된 컴퓨팅 환경에서 효율적으로 분산 병렬 처리를 수행할 수 있도록 하는 웹컴퓨팅 방법이 제시되었다. 본 연구에서는 인트라넷 상의 유휴 컴퓨팅 자원을 활용하는 '협력하는 웹컴퓨팅 환경 (Cooperative Web Computing Environment: CWCE)'을 구축하였다. 이 환경은 특히 협력하는 병렬 프로그램을 효율적으로 수행할 수 있도록 한다. CWCE 컴퓨터들 사이의 통신 집중 감소를 위해 복수 개의 매니저 컴퓨터를 운용할 수 있는 방법을 제공한다. 또한 최적화된 수행을 지원하기 위해 필요한 매니저 컴퓨터의 개수를 결정할 수 있는 '매니징 레벨 결정 함수'를 제공한다. 본 연구에서는 동기화된 통신을 요구하는 응용 프로그램이 CWCE 환경에서 복수 개의 매니저 컴퓨터를 이용함으로써 보다 효율적으로 수행될 수 있음을 분석과 실험을 통해서 보이도록 하였다.

Web-based Distributed Parallel Computing Environment with Multi-Managing Method

Hye-Seon Maeng[†] · Tack-Don Han^{††} · Shin-Dug Kim^{†††}

ABSTRACT

The portability of Java language makes it possible to use heterogeneous computers without re-compiling of application programs. Java applet can also be transported to other computers via Web browser. In this research, a *Cooperative Web Computing Environment(CWCE)* that uses idle computers on the Intranet for cooperative parallel computing work is suggested. The CWCE allows to use more than a manager computer that sends applets and manages communication between other computers. The number of manager computers can be determined according to the characteristics of computing environment and any chosen application program. it can reduce the amount of communication overhead for the application programs especially with synchronized communication. For the CWCE, a decision function to determine the managing level is provided. The CWCE turns out to be useful computing environment for the applications with less computation request ratio and multi-managing can help to reduce the communication overhead especially for the applications with a high ratio of synchronization purpose communications.

1. 서 론

네트워크로 연결된 컴퓨팅 자원을 병렬처리에 이용

* 이 논문은 한국과학재단 특정기초연구지원을 받아 수행된 결과의 일부임(KOSEF 97-0102-0201-3).
† 준 회 원 : 연세대학교 대학원 컴퓨터학과
†† 중 심 회 원 : 연세대학교 기전공학부 정보산업전공 교수
††† 정 회 원 : 연세대학교 기전공학부 정보산업전공 교수
논문접수 : 1999년 3월 30일, 심사완료 : 1999년 5월 31일

하려는 시도가 90년대 초반부터 활발하게 이루어져 왔다. 워크스테이션 클러스터링으로 대표되는 분산 병렬 컴퓨팅 환경은 비용 대 성능비가 뛰어난 병렬 컴퓨팅 플랫폼의 하나로써 자리잡게 되었다[1,2]. 그러나 분산 병렬 컴퓨팅 환경은 모든 컴퓨팅 자원들에 대한 사용자 계정(user account)과 부가적인 소프트웨어[3,4]의 설치를 요구한다. 또한 이질적인 컴퓨팅 자원으로 구

성된 환경에서는 컴퓨팅 자원들 각각의 플랫폼에 맞는 컴파일링 작업이 추가로 요구된다. 따라서 분산 병렬 컴퓨팅 환경을 보다 다양한 분야에서 활발히 이용하기 위해서는, 보다 용이하게 컴퓨팅 자원을 확보하고 컴퓨팅 환경을 구축할 수 있는 방법이 지원되어야 한다.

최근 들어, 자바 언어[8]의 이식성을 이용하여, 이질적인 컴퓨팅 자원으로 구성된 컴퓨팅 환경에서 추가적인 컴파일 작업 없이 분산 병렬 처리를 수행할 수 있도록 하는 방법이 제시되었다. 이들 중에서 몇몇 연구[5,6,7]는 웹 인터페이스를 이용하여 자바 애플릿을 직접 전송하는 방법을 통해 응용 프로그램의 전달 및 수행을 용이하게 한다. 특히 웹 인터페이스 상에서 자바 애플릿을 수행 하는 방법은 컴퓨팅 자원에 대한 사용자 계정(user account)을 요구하지 않으면서도 프로그램 수행을 가능하게 하기 때문에 컴퓨팅 자원의 확보를 용이하게 한다. 이렇게 자바 언어로 프로그램을 작성하고 애플릿 형태의 프로그램을 웹 인터페이스를 통해서 전달 한 후에 수행시키므로써 컴퓨팅을 수행하는 방법을 '웹컴퓨팅(Web computing)'이라고 한다. 본 연구에서는 인터넷 상에 존재하는 유휴 컴퓨팅 자원을 이용하여 협력하는 병렬 프로그램을 효율적으로 수행할 수 있는 '협력하는 웹컴퓨팅 환경(Cooperative Web Computing Environment: CWCE)'을 구축하였다.

웹컴퓨팅 환경에서는 분산된 컴퓨팅 자원들 사이의 통신 방법으로써 메시지 전달 방식 또는 공유 메모리 방식을 이용한다. 그러나 프로그래밍 환경의 차이는 존재할지라도, 웹컴퓨팅에서 모든 통신은 자바 애플릿을 전송시킨 컴퓨터를 통해서 이루어진다. 따라서 메시지 전달 방식과 공유 메모리 방식 모두 웹컴퓨팅에 참여하는 컴퓨터 수가 많아지고 이들 사이에 동기화된 통신이 요구되는 경우에는 상호 통신을 제공하는 컴퓨터로의 통신 집중 문제를 직면하게 된다.

CWCE에서는 웹컴퓨팅에 참여하는 컴퓨터들에 대한 관리 및 상호 통신을 제공하는 컴퓨터를 응용 프로그램과 컴퓨팅 환경의 특성에 따라서 복수 개를 운용할 수 있는 방법을 제공한다. 이렇게 관리 및 상호 통신을 제공하는 컴퓨터를 두 개 이상 사용하는 방법을 '멀티 매니징(multi managing)'이라고 한다. 멀티 매니징은 CWCE만의 독자적인 방법으로서 많은 수의 컴퓨터들이 동기화된 통신을 수행하는 응용 프로그램에서 통신 집중을 감소시킬 수 있게 한다. CWCE에서는 분산된 컴퓨팅 자원들 사이의 통신 방법으로써 공유 메

모리 기반 프로그래밍 환경을 제공한다.

공유 데이터를 위한 데이터 타입은 크게 두 종류로 구분되는데 각각의 타입들은 자바 클래스로 정의되며 서로 다른 운용 방법을 가진다. 멀티 매니징을 이용하는 경우에는 데이터의 특성에 따라서 공유 데이터 타입을 달리 정의하므로써 공유 메모리 제공으로 인한 오버헤드를 최소화시킬 수 있다.

CWCE에서는 컴퓨팅 환경과 응용 프로그램의 특성에 따른 통신 시간에 대한 분석적인 모델을 제시하였다. 또한 이 모델을 이용하여 최적화된 수행을 지원하기 위해 필요한 매니저 컴퓨터의 개수를 결정할 수 있는 '매니징 레벨 결정 함수'를 제시하였다. 웹컴퓨팅 환경의 이용자는 매니징 레벨 결정 함수값과 멀티 매니징 비용에 대한 고려를 통하여 특정 응용 프로그램 수행을 위한 매니저 컴퓨터의 개수를 결정할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 현재까지 수행된 관련 연구를 소개하고 CWCE의 기여점을 제시하였다. 3장에서는 CWCE 구성 요소와 수행 메카니즘을 설명하였다. 4장에서는 CWCE에서 제공하는 공유 메모리의 특성과 운용 방식을 제시하였다. 5장에서는 CWCE 환경에서 응용 프로그램의 수행 시간 및 통신 시간에 대한 분석을 통하여 매니징 레벨 결정 함수를 제시하였다. 6장에서는 서로 다른 특성을 가지는 응용 프로그램들 분석하고 매니징 레벨 결정 함수를 이용하여 최적화된 수행을 위한 매니저 컴퓨터의 개수를 결정하는 예를 보였다. 또한 응용 프로그램에 대한 실험을 통해서 멀티 매니징으로 인한 이득과 함께 매니징 레벨 결정 함수의 정확성을 보였다.

2. 관련 연구

현재까지 웹컴퓨팅에 관한 대부분의 연구들은 자바 애플릿과 웹 인터페이스를 이용하여 병렬 컴퓨팅을 수행할 수 있는 기본 메커니즘을 제안하는 것에 초점을 맞추어 수행되었다. Charlotte[5]은 웹컴퓨팅의 프로그래밍 환경에 공유 메모리를 제공하였다. 공유 메모리 개념은 프로그래밍을 용이하게 하고 기존에 작성된 공유메모리 기반 멀티프로세서 환경의 응용 프로그램을 수행시킬 수 있는 장점을 가진다. Bayanihan[7]은 HORB[13]을 이용하여 원거리 객체를 호출하는 방법으로 웹 컴퓨팅 환경을 구축하였다. 이 연구에서는 특징적으로 웹 컴퓨팅의 응용 분야로서 Web Crawler를 제시하므로

써 과학계산에 관련된 컴퓨팅 작업 뿐만 아니라 웹 기반 환경에서 자체적으로 발생된 응용 문제에도 웹 컴퓨팅 환경이 이용될 수 있는 가능성을 보였다.

웹컴퓨팅에 관한 연구들은 거의 대부분 공통적으로 결합 허용과 스케줄링 방법을 거론하고 있다. 이것은 이 연구들이 웹컴퓨팅의 수행 환경을 신뢰할 수 없는 익명의 컴퓨팅 자원을 이용하는 인터넷 환경으로 가정하고 있기 때문이다. 반면 인터넷 상에 분산되어 있는 익명의 컴퓨팅 자원들을 웹 컴퓨팅에 참여 시키는 것에 대해서는 실질적인 방법을 제시하지 못하고 있다.

CWCE는 웹 컴퓨팅의 수행 환경을 인터넷에서 인트라넷으로 축소한다. 인트라넷 상의 컴퓨터를 이용하는 것은, 컴퓨팅 자원을 웹컴퓨팅에 참여시키는 것을 용이하게 하고 컴퓨팅 자원의 성능 정보와 함께 컴퓨팅 자원에 대한 신뢰성을 확보할 수 있게 한다. 컴퓨팅 자원에 대한 신뢰성의 확보는 수행 시간 도중에 중간 결과를 공유하면서 수행되어야 하는 협력하는 병렬 프로그램(co-operative parallel program)의 수행을 가능하게 한다. CWCE에서는 기존의 웹컴퓨팅 연구들이 간과해왔던, 분산된 컴퓨팅 자원의 상호 통신 중에 발생할 수 있는 통신 집중 문제에 대한 해결 방법을 제시한다. CWCE에서는 애플릿을 전송시키고 컴퓨팅 자원을 관리하는 컴퓨터를 복수 개 이용하는 방법을 제공함으로써 통신 집중을 감소시키므로써 궁극적으로 통신 오버헤드를 감소시키도록 한다. 또한 공유 데이터의 특성에 따라서 서로 다른 운용 방법으로 관리 되도록 하므로써 공유 메모리 관리로 인한 오버헤드[10]를 최소화하도록 하였다.

3. CWCE의 구성과 수행 메커니즘

CWCE를 구성하는 컴퓨터들은 워커 컴퓨터(worker computer), 매니저 컴퓨터(manager computer), 중재자 컴퓨터(arbitrator computer)의 세 종류로 구분된다. 워커 컴퓨터는 웹컴퓨팅에서 자신의 컴퓨터를 컴퓨팅 자원으로 제공한다. 매니저 컴퓨터는 웹컴퓨팅에 참여한 워커 컴퓨터들에게 애플릿을 전송하고 이들을 관리한다. 워커 컴퓨터들 사이의 통신을 위한 공유 메모리는 매니저 컴퓨터에 위치하게 된다. 중재자 컴퓨터는 워커 컴퓨터를 매니저 컴퓨터에 연결시키는 일을 담당한다. CWCE에서는 두 개 이상의 매니저 컴퓨터가 관리하는 멀티 매니징과 구분하여 매니저 컴퓨터를 한 개

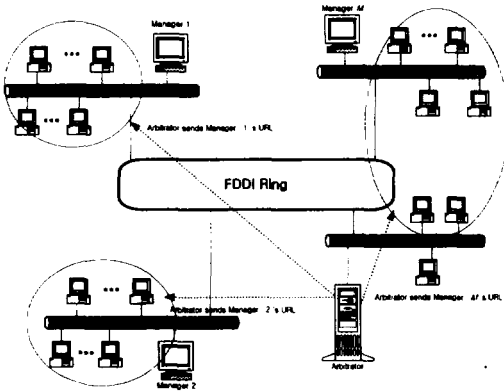
이용하는 경우는 '싱글 매니징(single-managing)'으로 언급하도록 한다.

웹컴퓨팅은 애플릿을 웹브라우저를 통하여 전송시키는 방법을 통하여 분산 병렬 컴퓨팅을 수행한다. 그러나 자바 애플릿은 보안상의 문제로 인하여 자신을 전송 시킨 서버 컴퓨터에게로만 네트워크 연결을 수행할 수 있으며 이러한 특성은 워커 컴퓨터들이 동시 발생된 통신을 수행할 때 매니저 컴퓨터로의 통신 집중을 야기시킨다. 자바 1.1 이상에서는 애플릿에 대한 제약을 선택적으로 완화할 수 있는 방법을 제시하고 있으나 컴퓨팅 자원 확보의 유용성 면에서는 자바 애플릿의 안전 보장을 지켜주는 것이 필요하다. 애플릿을 이용하지 않는 경우에도 공유 메모리 환경을 제공하는 경우에는 동일한 통신 집중이 발생할 수 있다. CWCE에서는 멀티 매니징 방법을 이용하여 매니저 컴퓨터로 발생할 수 있는 통신집중을 분산시키므로써 웹컴퓨팅의 통신 집중을 감소시키도록 한다.

중재자 컴퓨터는 접속된 워커 컴퓨터를 네트워크 위치에 따라서 그룹으로 나누어 각각의 그룹을 담당하는 매니저 컴퓨터로 연결시킨다. 각각의 그룹을 담당하는 매니저 컴퓨터를 해당 그룹에 속하는 워커 컴퓨터들의 '부모 매니저 컴퓨터(parent manager computer)'라고 부른다.

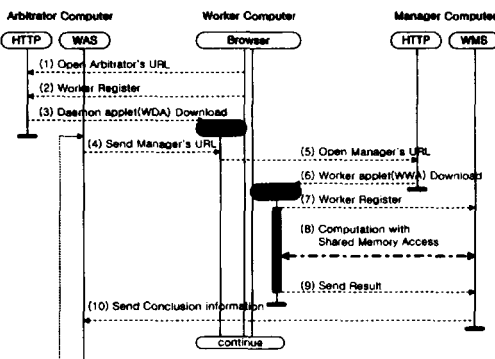
중재자 컴퓨터가 워커 컴퓨터를 그룹으로 나눌 때에는 같은 서브네트워크 상에 위치하는 워커 컴퓨터를 함께 묶어서 역시 같은 서브네트워크 상에 존재하는 매니저 컴퓨터로 연결시키는 방법을 이용한다. 워커 컴퓨터가 위치한 서브네트워크 상에 매니저 컴퓨터가 존재하지 않을 때에는 인접한 지역의 서브네트워크에 위치한 매니저 컴퓨터로 연결시킬 수 있다. 그러나 워커 컴퓨터와 매니저 컴퓨터가 서로 다른 서브네트워크 상에 위치할 때에는 같은 서브네트워크 상에 위치할 때에 비하여 통신 시간이 증대될 수 있다. 따라서 중재자 컴퓨터는 되도록 매니저 컴퓨터가 존재하는 서브네트워크 상에 존재하는 워커 컴퓨터들로 그룹을 구성하는 방법을 우선적으로 사용한다. 동일한 서브네트워크에 위치하는 워커 컴퓨터들을 하나의 그룹으로 연결할 수 있는 방법으로는 그룹핑[14]을 이용할 수 있다.

(그림 1)은 전체 네트워크가 FDDI로 연결되어 있으면서 인터넷이 서브 네트워크를 구성하고 있는 네트워크 환경에서 중재자 컴퓨터에 의해서 워커 컴퓨터들이 그룹 지어진 후 매니저 컴퓨터로 연결되는 상황을 보여준다.



(그림 1) 중재자 컴퓨터의 중재

워커 컴퓨터가 중재자 컴퓨터에 접속하면 중재자 컴퓨터는 워커 컴퓨터에게 매니저 컴퓨터에 대한 정보를 전달한다. 워커 컴퓨터는 매니저 컴퓨터에 재 접속하여 작업을 할당받고 이에 대한 처리를 수행한다. 모든 수행이 마쳐지고 결과가 매니저 컴퓨터에 전달되면 워커 컴퓨터는 새로운 작업을 할당 받기 위해서 중재자 컴퓨터에 메시지를 보내게 된다. 중재자 컴퓨터는 또 다시 워커 컴퓨터가 접속할 매니저 컴퓨터에 대한 정보를 전달하므로써 워커 컴퓨터는 새로운 작업을 시작하게 된다. (그림 2)는 CWCE에서의 웹컴퓨팅이 수행 주체에 따라서 순서대로 진행되는 것을 보여준다.



(그림 2) CWCE 수행 단계

4. CWCE 공유 메모리 환경

분산 공유 메모리를 운용하는 알고리즘[11]은 크게 복사본을 가지지 않는 방법과 복사본을 가지는 방법으

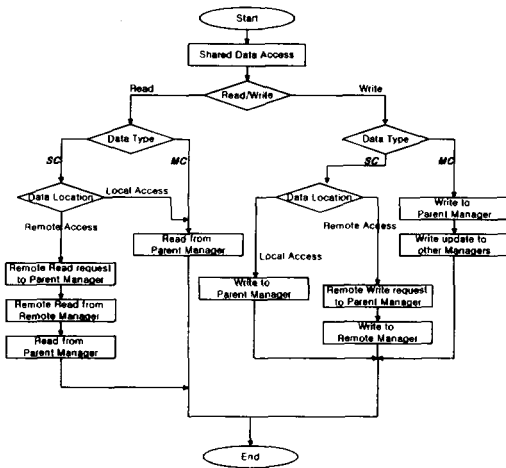
로 구분된다. 복사본을 가지지 않는 방법은 메모리 일관성을 유지시키는 방법이 간단하며 읽기와 쓰기 비용이 동일하다. 반면 복사본을 가지는 방법은 읽기 요청들이 여러 복사본에 대해서 동시에 수행될 수 있으므로 평균적인 읽기 비용을 감소시킨다.

CWCE에서는 데이터의 참조 특성에 따라서 각각의 장점을 취할 수 있도록 하기 위해서 복사본을 가지는 방법과 복사본을 가지지 않는, 두 가지 종류의 공유 데이터 타입을 제공한다. 전자의 방법을 이용하는 데이터 타입을 MC(Multiple Copied) 타입이라고 하고 후자의 타입을 SC(Single Copied) 타입이라고 한다. 각각의 타입은 자바 클래스로 정의되어지며 데이터 타입에 따라서 객체를 생성하여 이용될 수 있다.

MC 타입은 분산 공유 메모리 운용 알고리즘 중에서 완전-복사(full-replication) 방법에 따라서 운용된다. 따라서 모든 매니저 컴퓨터는 MC 타입의 데이터에 대한 복사본을 가지게 된다. 워커 컴퓨터가 부모 매니저 컴퓨터에게 MC 타입의 데이터에 대해서 읽기를 요청하면, 부모 매니저 컴퓨터는 자신의 공유 메모리에 존재하는 데이터를 가지고 읽기 요청을 처리한다. 그러나 워커 컴퓨터가 쓰기를 요청하면 이 요청을 받은 부모 매니저 컴퓨터 뿐만 아니라 다른 모든 매니저 컴퓨터에 존재하는 복사본에 대해서 수정이 이루어진다. 이러한 방법은 캐쉬 운용 방법 중에서 쓰기-수정(Write-update) 방식을 따르는 것이다.

SC 타입은 분산된 집중(distributed-central) 방법에 따라서 운용된다. 특정한 공유 데이터가 SC 타입으로 정의되면 모든 매니저 컴퓨터에는 SC 타입의 객체가 생성된다. 그러나 데이터 값이 변경되었을 때에 이들 객체들 모두에 대해서 데이터값 복사가 이루어지지 않는다. 다만 특정한 매니저 컴퓨터의 공유 메모리에 존재하는 객체만이 항상 최신의 데이터 값을 유지하게 된다. 이렇게 SC 타입 데이터의 최신값을 유지하고 있는 매니저 컴퓨터를 그 데이터의 '소유자 매니저 컴퓨터(owner manager computer)'라고 한다. SC 타입의 데이터 참조를 원하는 워커 컴퓨터의 부모 매니저 컴퓨터와 소유자 매니저 컴퓨터가 같은 컴퓨터인 경우에는 SC 타입 데이터에 대한 참조는 부모 매니저 컴퓨터의 공유 메모리에 위치하는 SC 타입의 객체 참조를 통해서 제공된다. 그러나 두 컴퓨터가 일치하지 않을 때에는 부모 매니저 컴퓨터는 소유자 매니저 컴퓨터에게 참조 요청을 발생시키고 원거리 참조를 수행한 후

에 워커 컴퓨터의 참조 요청을 처리하게 된다. SC 타입의 데이터에 대해서는 데이터의 위치에 따라서 소유자 매니저 컴퓨터와 부모 매니저 컴퓨터와 일치하는 경우 참조되는 방식을 자역적인 참조(local access)라고 하며 그렇지 못한 경우 참조되는 방식을 원격지 참조(remote access)로 구분하여 운용된다. (그림 3)에서는 공유 데이터 참조가 발생했을 때에 공유 데이터 타입과 데이터의 위치에 따른 운용 방식을 보여준다.



(그림 3) 공유 데이터의 운용 방식

5. 수행 시간과 멀티 매니징 레벨에 대한 분석적인 모델

CWCE 환경에서 응용 프로그램을 수행하는데 걸리는 시간은 일반적인 병렬 프로그램의 수행시간과 같이 계산시간과 통신시간으로 구분된다. 계산시간은 각 워커 컴퓨터의 컴퓨팅 파워에 따라서 일정한 시간이 소요되지만 통신시간은 여러 상황에 의해서 다양한 오버헤드가 발생될 수 있다. 따라서 다음과 같이 수행 시간을 분석하고 그 중에서 통신시간에 대한 분석적인 모델을 구성하는 것이 통신 집중으로 인한 오버헤드를 줄이기 위한 멀티 매니징 레벨을 결정할 수 있는 근거가 된다.

5.1 CWCE에서의 수행 시간

응용 프로그램을 병렬처리를 통해서 수행할 때에는, 프로그램의 순차수행 중간에 병렬수행이 포함되는 형태를 가진다. 순차수행과 병렬수행의 반복 회수는 프

로그래밍의 특성에 따라서 달라진다. 워커 컴퓨터들과 매니저 컴퓨터들로 구성된 CWCE환경 WC에서 응용 프로그램 A를 수행하는 데 소요되는 시간은 다음과 같이 정의할 수 있다.

$$T(A, WC) = TS_0 + \sum_{k=1}^m (Max_k(TP_k^i) + TS_i) \quad (1)$$

TS_i 는 i 번째 순차수행 부분의 수행시간을 의미하며 TP_k^i 는 워커 k 에 의해서 수행된 i 번째 병렬수행 부분의 수행 시간을 의미한다. 병렬로 처리되는 부분은 컴퓨터에 따라서 수행을 마치는 시간이 달라질 수 있기 때문에 가장 긴 수행시간에 의해서 병렬수행 부분의 수행시간이 결정된다. 병렬로 수행되어야 하는 작업이 참여하는 컴퓨터들 사이에 균형 있게 분배되었다면 병렬수행 시간을 간단히 TP_i 로 표시할 수 있다. 식 (1)은 프로그램 수행 도중 병렬수행 부분이 m 번 발생하는 응용 프로그램의 수행시간을 나타낸다. 보다 자세한 분석을 위해서 식을 간략화 시키면, 병렬수행의 회수 m 을 식 (2)에서와 같이 1로 정할 수 있다. 간략화된 식은 병렬수행 부분과 순차수행 부분의 쌍을 원하는 만큼 반복시키므로써 어떠한 프로그램 수행에도 적합하게 확장하는 것이 가능하다.

$$T(A, WC) = TS_p + TP + TS_s \quad (2)$$

TS_p 는 병렬수행 전의 순차수행 시간을 의미하고 TS_s 는 병렬수행 후의 순차수행 시간을 의미한다. 웹 컴퓨팅을 수행하기 위해서는 병렬수행 전에 웹컴퓨팅을 위한 준비 시간을 가져야 한다. 예를 들면 워커 컴퓨터들이 중재자 컴퓨터에 접속하는 시간 등이 이에 해당하며 식 (2)에서는 TS_p 에 그 값이 포함된다.

수행 시간 중에서 병렬수행 부분의 수행 시간은 다음과 같이 나누어질 수 있다.

$$TP = T_{CP} + T_{CM} \quad (3)$$

T_{CP} 는 병렬수행 부분에서 워커 컴퓨터 내부적으로 계산을 수행하는데 걸리는 시간이며 T_{CM} 은 다른 컴퓨터와의 통신을 수행하는데 걸리는 시간이다. 특정한 병렬수행 부분에서 몇 차례의 통신이 요구되는가는 응용 프로그램의 특성에 따라 다르다. 병렬수행 부분 중에서 계산을 통한 수행 중간 중간에 통신을 위한 시간

이 삽입되기 때문에 T_{CP} 와 T_{CM} 는 다음과 같이 정의할 수 있다.

$$T_{CP} = \sum_{j=1}^i T_{CP}^j, T_{CM} = \sum_{j=1}^i T_{CM}^j. \quad (4)$$

T_{CP}^j 는 병렬수행 부분에서의 j 번째 계산시간을 의미하며 T_{CM}^j 는 병렬수행 부분에서의 j 번째 통신시간을 의미한다. 병렬수행 부분에서 통신에 대한 요구가 t 번 일어난다면 계산과 통신은 번갈아 가면서 t 회 수행된다. 병렬수행에서 통신에 대한 요구 회수는 기본적으로 응용 프로그램의 특성을 따른다. 그러나 통신에 소요되는 시간은 응용 프로그램에서 기본적으로 요구되는 통신량에 더하여 웹컴퓨팅이 수행되는 수행 환경 특성의 영향을 받는다. 수행환경에 따라 기본적인 통신 시간에 더하여 부가적인 오버헤드가 추가될 수 있다.

5.2 공유 메모리 참조 비용

CWCE에서 통신 시간은 워커 컴퓨터가 공유 메모리를 참조하는 시간에 해당한다. 싱글 매니징을 이용하는 경우에는 MC 타입과 SC 타입이 운용상의 차이를 가지지 않는다.

멀티 매니징을 이용하는 경우에, 공유 데이터를 참조하는 비용은 공유 데이터의 타입과 위치에 따라 달라진다. 워커 컴퓨터가 공유 데이터 참조를 원하면 (그림 3)에서 제시하는 바에 따라서 참조가 수행된다. 공유 데이터 참조 비용 계산을 위한 기본적인 매개 변수들을 <표 1>과 같이 정의할 수 있다.

<표 1> 공유 데이터 참조 시간 매개 변수

	정 의
U_i	기본 단위 데이터를 전송 시간
$ D $	전송되는 데이터의 크기/기본 단위
P	웹컴퓨팅에 참여하는 워커 컴퓨터의 개수
M	웹컴퓨팅에 이용되는 매니저 컴퓨터의 개수
c	통신 요구 비율, T_{CM}/T_{CP}

멀티 매니징을 이용하는 경우, 웹컴퓨팅에 참여하는 워커 컴퓨터의 개수를 P 라 하고 이들을 관리하는 매니저 컴퓨터의 개수를 M 이라고 하면 하나의 매니저 컴퓨터가 담당하게 되는 워커 컴퓨터의 개수는 평균적으로 P/M 이 된다. 통신 요구 비율 c 는 응용 프로그

램이 나타내는 계산 시간에 대한 통신 시간 비율을 나타낸다. CWCE에서 c 의 값은 최대 1의 값을 가지는 응용 프로그램만을 수행한다. 예나하면 c 의 값이 1을 넘는 경우에는 병렬 수행으로 인한 이득을 얻을 수 없기 때문이다. 공유 메모리 참조 비용을 싱글 매니징을 이용하는 경우와 멀티 매니징을 이용하는 경우로 나누어서 계산하면 다음과 같다.

5.2.1 싱글 매니징에서의 공유 메모리 참조

싱글 매니징을 이용하는 경우 워커 컴퓨터는 매니저 컴퓨터와 같은 서브 네트워크에 속한다고 가정하면, 읽기나 쓰기에 대한 기본적인 공유 메모리 참조 비용은 식 (5)와 같이 계산될 수 있다.

$$T^{W-M} = |D| \cdot U_i. \quad (5)$$

읽기의 경우에는 읽기 요청을 위해서, 쓰기의 경우에는 쓰기 확인을 위해서 4 바이트의 신호가 전달되어야 하지만 이 크기는 참조되는 데이터 크기에 비하여 매우 작은 시간을 의미하기 때문에 이에 대한 비용은 생략하기로 한다.

T^{W-M} 라는 것은 워커 컴퓨터와 매니저 컴퓨터 사이에서 경쟁하는 컴퓨터가 없는 경우, 즉 통신 집중이 발생하지 않았을 때의 공유 메모리 참조 비용을 의미한다. 따라서 식 (5)를 수정하여 n 개의 워커 컴퓨터가 동시에 공유 메모리 참조를 요청했을 때에 소요되는 시간을 나타내면 식 (6)과 같다.

$$T^{W-M}(n) = n \cdot |D| \cdot U_i. \quad (6)$$

모든 워커 컴퓨터가 동시에 공유 메모리 참조를 요구하면 이에 대한 수행이 마쳐지는 시간은 $T^{W-M}(P)$ 가 된다. 식 (6)을 이용하여 싱글 매니징 이용 시 공유 메모리 참조 비용을 계산하면 <표 2>와 같다

<표 2> 싱글 매니징에서 공유 메모리 참조 비용

동기화 목적	소요 시간	비 율
있 음	$T^{W-M}(P)$	ρ
없 음	$T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor)$	μ

동기화 목적이 있는 데이터 참조에 대해서는, 모든 워커 컴퓨터들이 거의 같은 시기에 데이터 참조 요청을 발생하게 된다. 따라서 참조 비용은 통신 집중이 발생된

상황을 고려하여 계산된다. 동기화 목적이 없는 데이터 참조들은 워커 컴퓨터들이 동시에 참조 요청을 발생하므로 생기는 통신 집중은 가지지 않는다. 그러나 계산 시간 대 통신 시간의 비율이 높다면 여러 워커 컴퓨터들의 공유 메모리 참조 요구가 겹쳐져서 발생할 수 있기 때문에 최악의 경우 모든 워커 컴퓨터가 동시에 공유 메모리 참조를 요구하는 상황이 발생할 수도 있다. 따라서 이러한 데이터의 참조 시간은 $T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor)$ 로 계산한다. $\lfloor (P-1) \cdot c \rfloor$ 은 비슷한 시기에 통신 요구를 발생시킬 수 있는 워커 컴퓨터의 수를 의미하며, 이 값은 통신 요구 비율에 따라 달라진다.

ρ , μ 는 응용 프로그램 A에서 공유데이터 참조들 중에서 동기화 목적이 있는 데이터 참조에 대한 비율과 동기화 목적이 없는 데이터 참조에 대한 비율을 의미하며 그 전체 합은 다음과 같다.

$$\rho + \mu = 1. \quad (7)$$

응용 프로그램에서 공유 메모리 참조에 대한 시도는 여러 회 이루어질 수 있으므로 여기서는 평균적으로 1회 걸리는 비용에 대해서 정의하도록 한다. 위의 식을 바탕으로 싱글 매니징(IM)을 이용하는 경우 공유 메모리 1회 참조에 대한 통신 비용은 다음과 같다. 단 계산의 편의상 참조되는 데이터의 크기가 동일하다고 가정한다.

$$T_{CM}^M = T^{W-M}(P) \cdot \rho + T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor) \cdot \mu. \quad (8)$$

식 (8)에 식 (6)을 대입하면 다음과 같다.

$$T_{CM}^M = U_i \cdot |D| \cdot \{P \cdot \rho + (1 + \lfloor (P-1) \cdot c \rfloor) \cdot \mu\}. \quad (9)$$

식 (9)에 의하면 워커 컴퓨터의 개수, P가 증가할수록 공유 메모리 참조 비용이 증가한다. 증가율은 특히 동기화 목적을 가진 데이터에 대한 참조 비율이 커지거나 통신 요구 비율이 커질수록 더욱 증가한다.

5.2.2 멀티 매니징에서의 공유 메모리 참조

응용 프로그램 A를 멀티 매니징을 이용하여 수행할 때에는 공유 데이터를 적절한 공유 데이터 타입으로 정의해야 한다. 그러나 같은 데이터 타입으로 정의되었다 하더라도 해당 데이터의 특성과 위치에 따라서

참조 비용이 달라질 수 있다.

멀티 매니징에서의 공유 데이터 참조 비용을 계산하기 전에 멀티 매니징 수행시 공유 메모리 참조 비용에서 새로 생겨난 요소 비용들을 정의하도록 한다. 식 (5)와 식 (6)에서는 워커 컴퓨터에서 부모 매니저 컴퓨터에 존재하는 공유 데이터를 참조할 때 소요되는 비용을 계산하였다. 멀티 매니징을 이용하는 경우에는 부모 매니저 컴퓨터가 아닌 다른 매니저 컴퓨터의 공유 데이터를 참조하는 작업이 발생할 수 있다. 매니저 컴퓨터와 매니저 컴퓨터 사이에서 공유 메모리 참조를 수행할 때에 동시에 이를 수행하려는 컴퓨터의 개수가 n개인 경우에 대한 참조 비용은 식 (10)과 같다.

$$T^{M-M}(n) = 2 \cdot n \cdot |D| \cdot U_i. \quad (10)$$

식 (6)과 식 (10)을 비교해보면, $T^{M-M}(n)$ 은 $T^{W-M}(n)$ 의 두 배 값을 가진다. 이것은 매니저 컴퓨터는 서로 다른 서브 네트워크에 위치하기 때문이다. 매니저 컴퓨터 사이에서 데이터를 주고 받기 위해서는 서브네트워크 → 메인 네트워크 → 서브네트워크로 이어지는 데이터 전송이 이루어져야 한다. 메인 네트워크들을 연결하는 네트워크의 속도는 기본적으로 서브네트워크의 10배 이상의 속도를 내기 때문에 이 값을 생략하고 서브네트워크상에서 소요되는 시간만을 계산할 수 있기 때문이다.

$T^{M-M}(1)$ 은 매니저 컴퓨터가 다른 매니저 컴퓨터의 공유 데이터 참조를 수행할 때 이를 시도하는 매니저 컴퓨터가 한 개인 경우에 대한 비용이다. 하나의 매니저 컴퓨터가 동시에 다른 모든 매니저 컴퓨터에게로 같은 데이터에 대한 공유 메모리 참조를 수행해야 하는 경우는 $T^{M-M}(M-1)$ 로 계산될 수 있다. 멀티 매니징에서는 워커 컴퓨터와 부모 매니저 컴퓨터 사이의 통신 집중을 최대 P/M대의 워커 컴퓨터가 집중된 통신을 발생시키는 것으로 계산한다. 식 (6)과 식 (10)을 이용하여 멀티 매니징 이용 시 공유 메모리 참조 시간을 계산하면 <표 3>과 같다.

멀티 매니징을 이용하면서 공유 데이터는 데이터 타입을 가지게 된다. 기본적으로 공유 데이터에 대한 참조는 동기화 목적이 있는 데이터 참조와 동기화 목적이 없는 데이터 참조로 구분되는데 동기화 목적이 있는 데이터 참조가 주로 요구되는 데이터는 쓰기 참조에 대한 읽기 참조 비율이 높지 않기 때문에 효율적인

참조를 위해서 SC 타입으로 정의되어야 한다. 동기화 목적이 없는 데이터 참조가 주로 요구되는 데이터는 MC 타입과 SC 타입으로 정의될 수 있다. 프로그래머는 데이터 참조의 읽기와 쓰기 비율 등을 고려하여서 타입을 정해줄 수 있다.

멀티 매니징에서, SC 타입 데이터에 대한 참조에서는 싱글 매니징에서와 달리 원거리 참조(remote access)가 발생될 수 있으므로 이 비율만큼 매니저 컴퓨터 사이의 공유 데이터 참조 비용이 포함된다. q_s 와 q_N 은 SC 타입의 데이터 참조 중에서 각각 동기화 목적을 가진 데이터 참조에서의 원거리 참조 비율과 동기화 목적을 가지지 않은 데이터 참조에서의 원거리 참조 비율을 의미한다.

<표 3> 멀티 매니징에서 공유 메모리 참조 시간

동기화 목적	타입	참조	소요 시간	비율
있음	SC	읽기/쓰기	$T^{W-M}(P/M) + q_s \cdot T^{M-M}(P/M)$	ρ
없음	MC	읽기	$T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_{MR}
		쓰기	$T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor) + T^{M-M}(M-1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_{MW}
	SC	읽기/쓰기	$T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor) + q_N \cdot T^{M-M}(1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_s

$\rho, \mu_{MR}, \mu_{MW}, \mu_s$, 는 응용 프로그램 A에 대한 공유 데이터 참조들 중에서 동기화 목적이 있는 데이터 참조에 대한 비율과 동기화 목적이 없는 데이터 참조 중에서 MC 타입으로 정의된 데이터에 대한 읽기 참조 비율, 쓰기 참조 비율, SC 타입으로 정의된 데이터에 대한 비율을 의미하며 그 전체 합은 다음과 같다.

$$\rho + \mu_{MR} + \mu_{MW} + \mu_s = 1 \tag{11}$$

<표 3>의 각 데이터 타입 및 참조별 소요시간을 그 데이터 타입 및 참조에 대한 비율과 곱한 합은 M개의 매니저 컴퓨터를 이용하여 멀티 매니징을 수행했을 때의 평균적인 1회 통신 비용 $T_{CM}^{MM}(M)$ 이라고 정의한다.

<표 2>와 <표 3>을 통하여 각 데이터 특성과 그에 대한 비율에 따른 멀티 매니징 방법을 수행하므로써 얻을 수 있는 감소된 통신 오버헤드를 <표 4>와 같이 계산할 수 있다.

<표 4> 멀티 매니징으로 인해 감소된 통신 오버헤드

동기화 목적	타입	참조	소요 시간	비율
있음	SC	읽기/쓰기	$T^{W-M}(P) - T^{W-M}(P/M) - q_s \cdot T^{M-M}(P/M)$	ρ
없음	MC	읽기	$T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor) - T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_{MR}
		쓰기	$T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor) - T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor) - T^{M-M}(M-1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_{MW}
	SC	읽기/쓰기	$T^{W-M}(1 + \lfloor (P-1) \cdot c \rfloor) - T^{W-M}(1 + \lfloor (P/M-1) \cdot c \rfloor) + q_N \cdot T^{M-M}(1 + \lfloor (P/M-1) \cdot c \rfloor)$	μ_s

<표 4>를 바탕으로 공유 메모리 참조 1회에 대하여 멀티 매니징으로 얻을 수 있는 감소된 통신 오버헤드를 다음과 같이 계산할 수 있다.

$$R(M) = T_{CM}^{1M} - T_{CM}^{MM}(M) \\ = U_i \cdot |D| \left\{ \rho \cdot P \left(1 - \frac{1}{M} (1 + 2q_s) \right) + \mu \cdot \lfloor (P-1) \cdot c \rfloor - (\mu_{MR} + 3\mu_{MW} + (1 + 2q_N) \cdot \mu_s) \cdot \lfloor \left(\frac{P}{M} - 1 \right) \cdot c \rfloor - 2((M-1)\mu_{MW} + q_N \cdot \mu_s) \right\}$$

매니징 레벨의 결정은 기본적으로는 $R(M)$ 이 최대가 되는 M을 선택하는 것이다. $R(M)$ 을 구성하는 요소 중에서 $U_i \cdot |D|$ 값은 $R(M)$ 의 증감이 변하는 데에 영향을 주지 않는다. 따라서 멀티 매니징 결정식에서는 이 요소를 제외한 식을 이용하여 멀티 매니징 레벨을 결정하도록 한다. 멀티 매니징 레벨 결정값 $L(M)$ 은 다음과 같이 정의된다.

$$L(M) = \rho \cdot P \left(1 - \frac{1}{M} (1 + 2q_s) \right) + \mu \cdot \lfloor (P-1) \cdot c \rfloor - (\mu_{MR} + 3\mu_{MW} + (1 + 2q_N) \cdot \mu_s) \cdot \lfloor \left(\frac{P}{M} - 1 \right) \cdot c \rfloor - 2((M-1)\mu_{MW} + q_N \cdot \mu_s) \tag{12}$$

매니징 레벨은 매니징 레벨 결정 함수 $L(M)$ 이 최대값을 갖을 때의 M으로 결정된다. $L(M)$ 의 증감은 매니저 컴퓨터 수의 변화에 따라서, 통신 집중으로 인한 오버헤드가 감소하는 정도를 나타낸다. 매니저 컴퓨터의 수, M이 증가함에 따라서 $L(M)$ 의 변화하는 정도가 미미할 때에는 최대값을 가지는 M을 선택하기 보다는 $L(M)$ 의 증가율이 일정 한계값을 넘지 못하는 범

위 내에서 멀티 매니징 레벨 M 을 결정할 수 있다.

6. 응용 프로그램별 분석 및 실험

응용 프로그램의 특성과 웹컴퓨팅 환경에 따라서 최적 수행을 위한 매니저 컴퓨터의 수가 달라짐을 보이기 위하여 두 가지 종류의 응용 프로그램에 대하여 매니징 레벨 결정 함수를 적용하였다. 또한 각각의 응용 프로그램을 네트워크 시뮬레이션 환경에서 수행함으로써 매니징 레벨 결정 함수의 정확성과 함께 유용성을 입증하도록 하였다.

모든 실험은 SMPL[9] 시뮬레이션 라이브러리의 Ether 모델을 이용하여 수행되었다. Ether 모델에서 제공하는 이더넷 모델에 부모 매니저 컴퓨터와 그에 속한 워커 컴퓨터들이 위치하도록 설정하였다. 다른 이더넷에 속한 매니저 컴퓨터로부터 데이터를 참조할 때에는 그에 대한 참조 비용이 계산되도록 하였다.

6.1 야코비안 알고리즘을 이용한 온도 분포 추정

야코비안 알고리즘은 알려지지 않은 벡터 x 를 알기 위해서 선형 시스템인 $Ax = B$ 를 푸는 반복하는 방법 [12]으로서 $N \times N$ 격자상에서 점진적인 온도 분포 (temperature distribution) 추정 문제를 푸는데 이용될 수 있다. 이 문제에서 격자의 셀은 처음 온도로 초기화된다. 매 회마다 각 셀에 대한 새 온도값이 다음 식에 의해서 계산된다. 우측 식의 값들은 t 회에 (i, j) 셀의 이웃하는 셀들의 온도값이다. 이들의 평균값을 이용하여 $t+1$ 회에 (i, j) 셀의 온도값을 계산할 수 있다.

$$x'_{i,j} = \frac{x_{i-1,j} + x_{i,j-1} + x_{i+1,j} + x_{i,j+1}}{4}$$

$N \times N$ 격자는 P 개의 워커 컴퓨터들에게 N/P 크기의 열과 N 크기의 행을 가지는 격자로 나누어질 수 있다. 첫번째 워커 컴퓨터와 P 번째 워커 컴퓨터를 제외하고는 모두 인접한 워커 컴퓨터와 $1 \times N$ 격자 데이터를 두 부분씩 공유해야 한다. 첫번째 워커 컴퓨터와 P 번째 워커 컴퓨터는 한쪽에 존재하는 이웃한 워커 컴퓨터와 $1 \times N$ 격자 정보를 공유한다. 이 프로그램을 수행할 때, 최적화된 매니저 컴퓨터의 개수를 결정할 수 있는 매니징 레벨 결정 함수를 이용하기 위해서는 정의될 수 있는 데이터 타입과 참조 특성을 찾아내야 한다.

P 개의 워커 컴퓨터를 사용한다면 양쪽 끝에 위치한 워커 컴퓨터를 제외하고는 모든 워커 컴퓨터가 양쪽에 위치하는 $1 \times N$ 짜리 데이터 영역을 공유 메모리에 저장한다. 첫번째 워커 컴퓨터와 P 번째 워커 컴퓨터는 각각 한쪽에 위치하는 $1 \times N$ 짜리 데이터 영역을 공유 메모리에 저장한다. 따라서 공유 메모리에 저장되는 데이터의 크기는 $2(P-1) \times N$ 이 된다.

이 데이터들은 동기화의 특성을 가진다. 매 회에 대한 계산이 마쳐지면 모든 워커 컴퓨터들 사이에서 데이터 교환이 이루어지고 다시 다음 회의 계산이 시작되기 때문이다. 따라서 SC 타입으로 정의된다.

인접한 영역을 맡은 워커 컴퓨터들을 M 개의 매니저가 관리하는 그룹으로 나눈다면 $2(M-1) \times N$ 크기의 데이터에 대한 참조는 원거리 참조의 대상이 된다. 이 데이터를 공유하는 매니저 컴퓨터의 수는 두 개가 되며 각각이 번갈아 가면서 참조한다고 했을 때 원거리 참조 비율은 0.5가 된다. 따라서 모든 데이터 참조에 대한 원거리 참조 비율은 다음과 같이 계산된다.

$$a_s = 0.5 \times \frac{2(M-1) \times N}{2(P-1) \times N} = \frac{M-1}{2(P-1)}$$

공유 데이터 참조 특성별 비율을 계산하면 <표 5>와 같다.

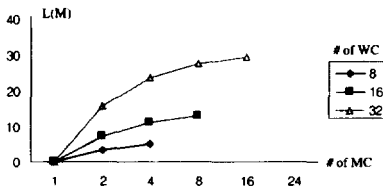
<표 5> 2차원 온도 분포 추정 프로그램의 데이터 특성별 비율

비율	ρ	μ_{MR}	μ_{MW}	μ_s
값	1	0	0	0

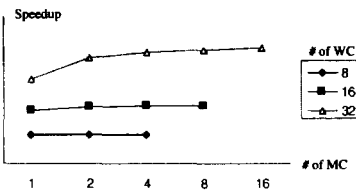
<표 5>의 요소들을 매니징 레벨 결정 함수인 식 (12)에 대입하면 (그림 4)와 같은 결과를 얻을 수 있다. 이 결과에서 매니징 레벨 결정 함수값은 같은 매니저 컴퓨터 수에 대해서, 워커 컴퓨터가 증가함에 따라 더 높은 값을 갖음을 보여준다. 이것은 워커 컴퓨터의 수가 증가함에 따라 통신 오버헤드가 증가하고 따라서 멀티 매니징으로 인한 이득도 증가하기 때문이다. 또한 매니징 레벨 결정 함수값은 워커 컴퓨터의 개수가 일정 숫자 이상이 되면 증가하는 폭이 둔해져서 수렴하는 것을 알 수 있다.

실험은 $10^5 \times 10^5$ 의 입자들이 500회가 지났을 때의 온도 변화 상태를 추정하도록 수행되었다. (그림 5)는

매니저 컴퓨터의 수가 증가함에 따라서 변화하는 스피드업을 보여준다. 워커 컴퓨터의 수가 많은 경우에는 매니저 컴퓨터의 수를 복수개로 운용하므로써 스피드업의 크게 증가함을 알 수 있다. 또한 (그림 5)의 스피드업 변화는 (그림 4)의 멀티 매니징 레벨 결정식과 같은 형태로 증감을 나타내고 있다. 이는 (그림 5)의 멀티 매니징 레벨 결정식의 값의 변화가 감소시킬 수 있는 통신 오버헤드의 양을 정확하게 예측해내고 있기 때문이다.



(그림 4) TD의 L(M) 함수값



(그림 5) TD의 실험 결과

6.2 행렬 곱셈(Matrix Multiplication)

행렬 곱셈은 $N \times N$ 행렬인 A 와 B 에 대해서 $C = A \times B$ 를 만족하는 $N \times N$ 행렬 C 를 계산하는 것이다. 워커 컴퓨터가 P 개 주어졌을 때 행렬 A 를 N/P 개의 연속적인 행으로 나누어 각각의 워커 컴퓨터에게 분배한다. 행렬 B 는 모두 N 개의 연속적인 열로 나눈다. i 번째 워커 컴퓨터는 행렬 A 의 $N/P(i-1)+1$ 번째 행으로부터 $N/P \cdot i$ 번째까지의 행을 가지고 행렬 c 의 $N/P(i-1)+1$ 번째 행으로부터 $N/P \cdot i$ 번째까지의 행에 대한 값을 계산한다. 행렬 B 의 값은 N 개의 연속적인 열로 나누어져서 각 워커 컴퓨터에 의해서 열단위로 참조된다.

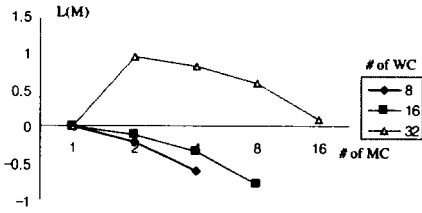
행렬 B 와 C 는 모두 공유 메모리에 저장되며 동기화의 목적 없이 참조된다. 행렬 B 의 데이터는 모든 워커 컴퓨터들에 의해서 참조되어야 한다. 맨 처음 초기 값이 '쓰기' 수행된 후에는 오직 '읽기' 참조만이 수행되기 때문에 '읽기' 비용이 저렴한 MC 타입으로 정의된다. 행렬 C 의 데이터는 각 부분이 오직 한 워커 컴

퓨터에 의해서만 '쓰기' 참조 되기 때문에 '쓰기' 비용이 저렴한 SC 타입으로 정의된다. 이때 모든 참조는 부모 매니저 컴퓨터에서 이루어질 수 있으므로 원거리 참조 비율 q_N 은 0의 값을 가진다. 공유 메모리에 저장되는 데이터의 크기는 행렬 B 의 데이터와 행렬 C 의 데이터의 크기가 같지만 참조 비율은 다르게 계산된다. 행렬 B 의 데이터에 대해서는 모든 데이터에 대한 한번의 쓰기 참조 수행 후, 각 열이 모든 워커 컴퓨터에 의해서 참조된다. 행렬 C 의 데이터에 대해서는 i 번째 워커 컴퓨터가 $N/P(i-1)+1$ 번째 행으로부터 $N/P \cdot i$ 번째까지의 행에 대해 참조한다. 따라서 참조 비율을 계산하면 $N \times N$ 행렬 B 와 C 에 대하여, 행렬 B 의 경우는 $N \times 1$ 의 데이터가 $N \times 1$ 회 쓰기 참조된 후 $N \times 1$ 의 데이터가 $N \times 1$ 회 읽기 참조되며 행렬 C 의 경우는 $1 \times N$ 의 데이터가 $N \times 1$ 회 쓰기 참조된다. 통신 요구 비율 c 는 행렬의 크기와 네트워크 속도에 따라서 달라진다. 기본적으로 이 응용 프로그램은 이동되는 데이터량에 대한 연산 비율이 $(1+P)/N$ 의 값을 가진다. 연산 속도와 통신 속도는 많은 차이를 가지므로 통신 요구 비율 c 의 값은 $10^3(1+P)/N$ 의 값으로 계산하도록 한다. 실제로 이 값은 네트워크 속도와 컴퓨터의 연산 속도에 따라서 달라질 수 있다. 공유 데이터 참조 특성별 비율을 계산하면 <표 6>과 같다.

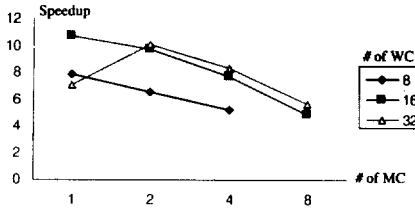
<표 6> 행렬 곱셈 프로그램의 데이터 특성별 비율

비율	ρ	μ_{MR}	μ_{MW}	μ_s
값	0	$\frac{P}{P+2}$	$\frac{1}{P+2}$	$\frac{1}{P+2}$

<표 6>의 요소들을 매니징 레벨 결정 함수인 식 (12)에 대입하면 (그림 6)과 같은 결과를 얻을 수 있다. 이 결과에서 매니저 컴퓨터의 수가 증가함에 따라, 매니징 레벨 결정 함수값은 증가 후 감소하거나 처음부터 감소하는 것을 알 수 있다. 이것은 행렬 곱셈 문제가 동기화의 목적을 가지지 않은 통신을 수행하는 특성을 가지고 있기 때문이다. 따라서 멀티 매니징으로 인해 얻을 수 있는 통신 집중의 분산보다 멀티 매니징 유지를 위해 수행해야 하는 데이터 일관성 유지 비용이 더 증가할 수 있기 때문이다. 실험은 $10^4 \times 10^4$ 의 행렬 A 와 B 에 대해서 곱셈을 수행하도록 하였다. (그림 7)은 매니저 컴퓨터의 수가 증가함에 따라서 변화하는 스피드업을 보여준다.



(그림 6) MM의 L(M) 함수값



(그림 7) MM의 실험 결과

행렬 곱셈 문제는 통신 요구 비율이 워커 컴퓨터의 수에 따라서 변화한다. 워커 컴퓨터의 수가 증가하게 되면 계산시간은 줄어들지만 상대적으로 통신시간은 변화하지 않기 때문이다. 따라서 워커 컴퓨터의 수가 증가하더라도 스피드업이 증가되지 못하고 오히려 감소되는 결과를 보여줄 수 있다. (그림 7)에서는 매니저 컴퓨터의 수가 한 대인 경우 워커 컴퓨터의 수가 32대인 경우가 8대나 16대인 경우보다 오히려 낮은 스피드업을 가진다. 그러나 매니저 컴퓨터의 수가 2대인 경우에는 멀티 매니징으로 인하여 통신 오버헤드를 줄이므로 인해 워커 컴퓨터의 수가 32대인 경우가 가장 높은 스피드업을 나타내게 된다. (그림 7)에서 보여주는 그래프는 (그림 5)에서 MMD 함수의 결과를 이용하여 추정된 통신 오버헤드 감소율이 정확하게 반영되고 있음을 보여준다.

7. 결 론

웹컴퓨팅 환경은 기존의 분산 병렬 컴퓨팅 환경의 제약으로 여겨졌던 자원 확보의 문제를 웹인터페이스와 자바 언어를 이용하여 용이하게 하였다. 그러나 자바 언어의 안전 규정으로 인해 동기화된 통신을 필요로 하는 응용 프로그램을 수행시킬 때 통신 집중의 문제를 겪을 수 있다. 본 연구에서는 인트라넷 상에 존재하는 유휴 컴퓨팅 자원을 이용하여 분산 병렬 컴퓨팅을 수행할 수 있는 협력하는 웹컴퓨팅 환경(CWCE)을

제안하였다. CWCE는 일반적인 웹컴퓨팅 환경이 가지고 있는 통신 집중 문제를 해결하기 위해서 두 개 이상의 매니저 컴퓨터를 이용할 수 있는 멀티 매니징 방법을 제공하였다. 이 방법을 통해서 웹컴퓨팅 환경에서 발생하는 동기화의 특성을 가진 통신으로 인한 오버헤드는 상당량이 감소될 수 있으며 결과적으로 수행 시간을 단축시킬 수 있다. CWCE에서는 또한 통신 시간에 대한 분석적인 모델을 제시하면서 최적화된 수행을 지원하기 위해 필요한 매니저 컴퓨터의 개수를 결정할 수 있는 매니징 레벨 결정 함수를 제시하였다. 이 함수를 통해 응용 프로그램의 특성을 고려하여 최적화된 수행을 위한 매니저 컴퓨터의 개수를 결정할 수 있으며 사용자는 멀티 매니징 비용과 수행 시간 감소에 대한 고려를 통하여 최적화된 수행 환경을 구축할 수 있다.

본 연구에서는 워커 컴퓨터가 매니저 컴퓨터와 같은 서버넷에 위치하는 경우에 대해서만 분석적인 모델을 제시하고 실험을 실시하였다. 워커 컴퓨터와 매니저 컴퓨터가 서로 다른 서버넷에 위치하는 것은 통신 시간의 지연을 가져올 수 있지만 한 서버넷에 충분한 수의 워커 컴퓨터가 존재하지 못하는 경우에는 서로 다른 서버넷에 위치하는 워커 컴퓨터와 매니저 컴퓨터가 연결되어야 할 필요가 있다. 이에 대한 분석은 향후 연구에서 보다 심층적으로 연구될 것이다. 또한 본 연구의 향후 연구에서는 동적으로 변화하는 컴퓨팅 환경을 고려하여 작업을 수행할 수 있는 CWCE 환경의 확장과 함께 이를 고려한 매니징 레벨 결정 함수를 제시할 계획이다.

참 고 문 헌

- [1] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam, "PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing," The MIT Press, 1994.
- [2] Donald J. Becker, Thomas Sterling, Daniel Savarese, John E. Dorband, Udaya A. Ranawak, Charles V. Packer, "Beowulf: a parallel workstation for scientific computation," ICPP, 1995.
- [3] William Gropp, Ewing Lusk, and Anthony Skjellum, "Using MPI: Portable Parallel Pro-

gramming with the Message-Passing Interface," The MIT Press, 1994.

- [4] Thomas E. Anderson, David E. Culler, David A. Patterson, and the Now team, "A Case for NOW(Network of Workstations)," IEEE Micro, Vol.15, No.1, pp.54-64, Feb., 1995.
- [5] Arash Baratloo, Mehmet Karaul, Zvi Kedem, and Peter Wyckoff, "Charlotte: Metacomputing on the Web," 9th International Conference on Parallel and Distributed Computing Systems, pp. 151-159, Sep., 1996.
- [6] Bernd O. Christiansen, Peter Cappello, Mihai F. Ionescu, "Javelin: Internet-Based Parallel Computing Using Java," 1997 ACM Workshop on Java for Science and Engineering Computation, pp.30-40, June, 1997.
- [7] Luis F. G. Sarmenta, "Bayanihan: Web-Based Volunteer Computing Using Java," 2nd International Conference on World-Wide Computing and its Applications (WWCA'98), Mar., 1998.
- [8] Ken Arnold and James Gosling, The Java Programming Language, Addison-Wesley, 996.
- [9] M. H. MacDougall, Simulating Computer Systems: Techniques and Tools, The MIT Press, 1987.
- [10] Holger Karl, "Bridging the Gap between Distributed Shared Memory and Message Passing," Concurrency: Practice and Experience 10:1-14 1998.
- [11] Michael Stumm and Songnian Zhou, "Algorithms Implementing Distributed Shared Memory," IEEE Computer, pp.54-64, May, 1990.
- [12] M.J. Quinn, Parallel Computing: Theory and Practice, McGraw-Hill Book Company, New York, New York, 1994.
- [13] S.Hirano, "HORB: Distributed Execution of Java Programs," WWCA' 97, Lecture Notes in Computer Science, Vol.1274(Springer, Berlin, 1998), pp.444-461.
- [14] Chun-Mok Chung, Pil-Sup Shin, and Shin-Dug Kim, "A Java Internet Computing Environment with Effective Configuration Method," 1999 In-

ternational Symposium on Parallel Architectures, Algorithms, and Networks, To be appeared, 1999.



맹혜선

e-mail : carchi@kurene.yonsei.ac.kr
 1994년 연세대학교 전산학과과 졸업(학사)
 1996년 연세대학교 대학원 컴퓨터 과학과 졸업(이학석사)
 1996년~현재 연세대학교 대학원 컴퓨터과학과 박사과정

관심분야 : 분산병렬컴퓨팅, 인터넷 컴퓨팅, 자바



한탁돈

e-mail : hantack@kurene.yonsei.ac.kr
 1978년 연세대학교 공과대학 전자공학과 졸업(학사)
 1983년 Wayne State University 컴퓨터공학(석사)
 1987년 University of Massachusetts 컴퓨터공학(박사)

1987년~1989년 Cleveland 주립대학 조교수
 1989년~현재 연세대학교 공과대학 기전공학부 정보산업전공 교수
 관심분야 : 병렬처리, 컴퓨터구조 및 알고리즘, VLSI 설계, HCI



김신덕

e-mail : sdkim@kurene.yonsei.ac.kr
 1982년 연세대학교 공과대학 전자공학과 졸업(학사)
 1987년 University of Oklahoma 전기공학(석사)
 1991년 Purdue University 전기공학(박사)

1993년 2월~1995년 2월 광운대학교 컴퓨터공학과 조교수
 1995년 3월~현재 연세대학교 공과대학 기전공학부 정보산업전공 부교수
 관심분야 : 병렬처리 시스템, 컴퓨터 구조, 인터넷 컴퓨팅