로버스트 추정을 이용한 다중 프로세서에서의 데이터 통신 예측 모델

전 장 환[†] · 이 강 우[††]

요 약

본 논문에서는 최소제곱 추정기법과 로버스트 추정기법을 사용하여 다중 프로세서 시스템에서의 데이터 통신의 빈도를 모델링하는 방법을 제안한다. 몇 가지의 서로 다른 크기의 작은 입력 데이터들을 작업부하 프로그램에 부과하여 그때마다의 통신 빈도를 측정하고, 이 측정된 값들에 두 가지 통계적 추정기법을 순차적으로 적용함으로써 통신 빈도를 정확히 예측할 수 있는 모델을 구축하는 방법이다. 이 모델링 기법은 작업부하나 목표시스템의 구조적인 사양에 무관하게 입력 데이터의 크기에만 의존하므로 다양한 작업부하와 목표시스템에 대하여 그대로 적용할 수 있는 장점이 있다. 또한 목표시스템에서 작업부하의 알고리즘적 동적특성이 수학적인 공식으로 반영되므로 데이터 통신이외의 성능 데이터를 모델링하는 데에도 적용할 수 있다. 본 논문에서는 대표적인 다중 프로세서인 공유메모리 시스템에서 데이터 통신을 유발하는 핵심 요소인 캐시접근실패의 빈도에 대한 모델을 구하였으며, 12번의 실험 중 5번의 경우에는 1% 미만, 나머지 경우에는 3% 내외의 대단히 정확한 예측 오차율을 보였다.

키워드 : 모델링, 예측 모델, 데이터 통신, 통계적 추정기법

Data Communication Prediction Model in Multiprocessors based on Robust Estimation

Janghwan Jun[†] · Kangwoo Lee[††]

ABSTRACT

This paper introduces a noble modeling technique to build data communication prediction models in multiprocessors, using Least-Squares and Robust Estimation methods. A set of sample communication rates are collected by using a few small input data sets into workload programs. By applying estimation methods to these samples, we can build analytic models that precisely estimate communication rates for huge input data sets. The primary advantage is that, since the models depend only on data set size not on the specifications of target systems or workloads, they can be utilized to various systems and applications. In addition, the fact that the algorithmic behavioral characteristics of workloads are reflected into the models entitles them to model diverse other performance metrics. In this paper, we built models for cache miss rates which are the main causes of data communication in shared memory multiprocessor systems. The results present excellent prediction error rates; below 1% for five cases out of 12, and about 3% for the rest cases.

Key Words : Modeling, Prediction Model, Data Communication, Statistical Estimation Methods

1. 서 론

분석적 모델링은 컴퓨터 하드웨어나 소프트웨어 등의 시스템을 설계하고 개발하는 과정에서 널리 사용되어 왔다. 분석적 모델을 이용하면 완전한 시스템을 구축하지 않고도 목표 시스템의 성능은 물론 장단점까지도 쉽고 빠르게 분석할 수 있다[1, 2]. 이러한 장점 때문에 분석적 모델링은 컴퓨터뿐만 아니라 모든 과학과 공학 전반의 연구에 있어서 오랫동안 중요한 분야로 자리매김 해왔다. 그러나 때로는 시스템의 수학적인 해석 절차가 매우 까다로울 수 있다. 특히 고려하는 변수들이 많아지면 수학적인 해석이 더욱 복잡해짐은 물론 모델을 도출하는 과정에서 수많은 가정을 수반하여야만 한다. 이와 같이 복잡한 과정은 모델을 일반화하는데 넘을 수 없는 장벽이 된다. 이러한 이유로 인하여 많은 분야에서 다양한 모델링 기법이 제안되어 왔지만 일반적으로 정확도가 낮으며, 정확도가 높은 경우라 하더라도 모델들을 적용할 수 있는 범위가 상당히 제한적이다.

본 논문에서는 구축 과정이 간단하면서도 결과에 대한 정확도가 매우 높은 분석적 모델을 구축하는 기법을 제안한다. 이 기법은 쉽게 수집할 수 있는 몇 가지 성능 데이터를 표

※ 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2002-003-D00258)
† 준 회 원 : 동국대학교 정보통신공학과 석사과정
†† 정 회 원 : 동국대학교 정보통신공학과 부교수(주저자)
논문접수 : 2005년 3월 21일, 심사완료 : 2005년 6월 1일

본으로 활용하여, 특별한 분석과정 없이, 통계적인 추정식을 이끌어내는 방식이다. 즉, 수집된 표본 데이터에는 주어진 입력에 대한 시스템의 고유한 반응양식이 내재되어 있게 마련인데, 소수의 표본 데이터에 통계적인 추정기법을 적용함으로써 그 반응양식을 수학적인 공식으로 표현해내는 기법으로써 다음과 같은 논리적인 배경을 갖는다.

일반적으로, 실험을 통하여 측정된 데이터의 값은 입력 데이터 등 다양한 변수들에 의하여 결정된다. 이 때, 입력데이터 등이 독립변수라면, 측정된 데이터는 독립변수들의 값에 따라 값이 결정되는 종속변수이다. 즉, 독립변수들을 $\mathbb{X} = (x_1, ..., x_m)$, 독립변수들이 종속변수의 값에 영향을 미치는 방식을 나타내는 계수를 $\mathbb{A} = (a_1, ..., a_M)$라 하면, 종속변수인 측정 데이터는 다음과 같은 다항식으로 표현된다.

$$y(\mathbb{X}) = f(\mathbb{X};\mathbb{A}) = \sum_{k=1}^{M} a_k X_k(\mathbb{X}) \qquad \text{(식 1)}$$

여기서 $X_k(x)$는 선형함수는 물론 지수함수나 삼각함수 등 비선형함수들을 모두 포함한다. 이와 같은 관점에서 보면 $\mathbb{A} = (a_1, ..., a_M)$의 값을 알아내어 (식 1)을 완성할 수 있다면, 이 식에 독립변수들의 실제 값을 대입함으로써 우리가 원하는 종속 변수의 값을 쉽고 빠르게 얻을 수 있는 예측 모델로 활용할 수 있다.

위와 같은 사실을 바탕으로 공유메모리 다중 프로세서 시스템에서의 통신 빈도를 예측할 수 있는 모델을 구하였다. 이 구조의 시스템에서 프로세서간의 통신을 유발하는 가장 중요한 요인인 캐시접근실패에 대한 예측 모델을 구하기 위하여, 우선 서로 다른 크기의 작은 입력 데이터를 작업부하 프로그램에 부과하여 각각의 경우에 대하여 캐시접근실패 빈도를 측정한다. 그리고 이들 측정 데이터에 최소 제곱 추정법(Least-Square Estimation)[3]과 로버스트 추정법(Robust Estimation)[4]을 순차적으로 적용함으로써 캐시접근실패 빈도에 대한 분석적 모델을 구한다. 이 모델은 입력 데이터의 크기를 유일한 변수로 가지며 작업부하나 컴퓨터 시스템의 사양에는 영향을 받지 않는다. 이 모델의 변수에 새로운 입력 데이터의 크기 값을 대입하면 그에 대한 캐시접근실패빈도를 쉽게 구할 수 있다.

본 논문에서 제안하는 모델은 작업부하나 목표시스템의 구조적인 사양에 무관하게 입력 데이터의 크기에만 의존하므로 다양한 작업부하와 목표시스템에 그대로 적용할 수 있는 장점이 있다. 또한 목표시스템 상에서의 작업부하의 알고리즘적인 동적특성이 수학적인 공식으로 반영되므로 데이터 통신이외의 성능 데이터를 모델링하는 데에도 다양하게 적용할 수 있다.

본 논문에서 제안하는 방법으로 구해진 예측 모델은 12번의 실험 중 5번의 경우에는 1% 미만, 나머지 경우에는 약 3% 내외의 정확한 예측오차율을 보였다.

본 논문은 다음과 같이 구성되었다. 2장에서는 관련된 연구 사례를 요약하고, 3장에서는 본 논문이 제안하는 모델링 기법을 소개하며, 4장에서는 통계적인 추정기법인 최소 제곱 추정법과 로버스트 추정법을 다룬다. 5장에서는 실험 환경과 더불어 본 논문이 제안하는 기법을 적용한 예측 모델 및 예측결과를 소개한다. 마지막으로, 6장에서 본 논문 결과에 대한 토론을 제시하면서 결론을 짓는다.

2. 관련 연구

과학과 공학 분야에서 발생하는 문제를 컴퓨터 시스템을 이용하여 해결하는 가장 일반적인 방식은 시간과 공간에서 발생하는 연속적인 문제들을 이산화를 통한 수치 해석적 접근방식에 기초한 알고리즘을 이용하여 시뮬레이션하는 것이다[5]. 시뮬레이션의 정확성을 향상시키기 위해서는 문제 공간을 확장하거나 이산화 과정에서의 조밀도를 높여야 한다. 이에 따라 데이터의 양과 계산의 양이 급증하며, 빠른 계산을 위해서는 궁극적으로 다중 프로세서 시스템을 사용하여야 한다. [6]과 [7]에서는 과학과 공학의 여러 분야에서 대표적으로 다루어지는 문제들을 공유메모리 시스템을 이용하여 해결할 수 있도록 구축한 응용 프로그램들을 소개한다. [5]에서는 이 문제들을 다중 프로세서 시스템을 이용하여 해결할 때, 데이터의 이산화 조밀도, 시뮬레이션을 위한 시간 구간 등에 따르는 오차를 분석하였다. 본 논문에서는 [6, 7]에서 소개된 병렬 프로그램들을 작업부하로 사용한다.

캐시접근실패는 공유메모리 시스템에서 프로그램의 성능을 좌우하는 핵심적인 요소이다. 캐시접근실패는 발생하는 원인에 따라 구분되는데 본 논문에서는 [8]에서 제안하는 구분방식을 채택하였다. 최초접근실패(cold miss)는 프로세서가 하나의 메모리 블록을 처음 접근할 때 발생하는 접근실패를 말한다. 공유데이터접근실패(sharing miss)는, 무한한 크기의 캐시를 가정할 때, 한 프로세서의 캐시에 저장되어 있던 데이터에 대한 원격 프로세서의 쓰기 동작으로 인하여 원래의 프로세서 캐시에 있던 블록이 무효화된 후 원래의 프로세서가 그 데이터를 접근할 때 발생하는 접근실패이다. 이를 정확히 말하자면 진성공유데이터접근실패(true sharing miss)라 한다. 이에 비하여 가성공유데이터접근실패(false sharing miss)는 실제로 공유되지는 않지만 실제로 공유되는 데이터와 함께 무효화된 블록에 저장되어 있던 데이터를 원래 프로세서가 접근할 때 발생한다. 마지막으로 교체접근실패(replacement miss)는 유한한 크기의 캐시에 있어서 발생하는 나머지 캐시접근실패를 통칭한다.

다중프로세서 시스템의 성능에 관한 분석적 모델링 방법을 적용한 연구 사례는 매우 다양하다. 예를 들어, [9]에서는 대용량의 병렬시스템의 전체적인 성능을 다루었으며, [10]은 프로세서의 수보다 많은 수의 독립적인 작업들을 수행하는 병렬 시스템의 성능을 Markov 체인을 이용한 분석적 모델로 표현하였다. 또한 다중 프로세서의 대표적인 구조인 공유메모리 시스템의 성능을 좌우하는 핵심요소인 캐시메모리에 초점을 둔 분석적 모델링을 다룬 연구도 매우 다양하게 소개되고 있다[10, 11, 12]. 이러한 모델들은 작업부하로 사

용된 각 응용 프로그램들의 데이터 공유에 관한 구조적인 특성을 기초로 하였다. 예를 들어, [11]이 제시하는 구조적 모델은 수정된 공유 블록을 접근하는 확률에 기초하였으며, 이로부터 얻어진 파라미터들로부터 캐시접근실패율을 포함한 몇 가지 성능에 대한 공식을 소개하였다. 또한 [12]는 캐시의 크기와 맵핑 방법 등 다양한 하드웨어적인 특성을 파라미터로 하는 분석적 성능 모델을 제시한다.

이와 같은 모델들은 파라미터에 따른 성능변화를 보여주거나 동일한 파라미터 값을 가지는 두개의 다른 컴퓨터의 구조를 비교하는데 도움을 준다. 하지만 이러한 모델들은 몇 가지 공통적인 문제점을 안고 있다. 첫째, 이 모델들은 특정한 작업부하 프로그램의 특징에 기초하여 구축되었다. 따라서 모델을 구할 때 사용하였던 작업부하 프로그램이외의 다른 프로그램에는 적용할 수 없다. 둘째, 모델의 파라미터들은 블록의 크기, 입력 데이터의 크기, 그리고 프로세서 수 등 목표 시스템의 구조적인 사양에 따라 그 값이 변한다. 따라서 새로운 사양을 갖는 목표시스템에 대해서는 동일한 모델을 적용할 수 없다. 셋째, 캐시접근실패의 종류를 충분히 구별하지 않았다. 이로 인하여 상이한 원인에 의하여 발생되는 캐시접근실패가 구분되지 않고 동일한 방법으로 모델링되어 이러한 모델을 이용하여 성능을 예측할 때 그 정확도에 제한이 있다.

3. 모델링 과정

실험을 통하여 측정된 데이터의 값은 실험 환경, 입력 데이터 등 다양한 변수들에 의하여 결정된다. 이 때, 실험 환경이나 입력데이터 등 결과에 영향을 미치는 모든 항목을 독립변수라면, 측정된 데이터는 독립변수들의 값에 따라 결정되는 종속변수라고 볼 수 있다. 따라서 m개의 독립변수를 $\mathbb{X} = (x_1, ..., x_m)$라 하고, 독립변수들이 종속변수의 값에 영향을 미치는 정도를 나타내는 M개의 계수를 $\mathbb{A} = (a_1, ..., a_M)$라 하면, 종속변수인 측정 데이터는 (식 1)과 같은 다항식으로 표현된다.

한편, 과학이나 공학 분야의 응용 프로그램들은 시간과 공간적으로 연속적인 문제를 이산화 과정을 거친 후 일련의 수학적인 해결 절차를 컴퓨터 명령어로 대치한 것으로써 그 안에는 고유한 알고리즘이 내재되어 있다. 프로그램이 실행되는 동안 발생하는 데이터 접근은 프로그램의 알고리즘적인 특성에 따라 일정한 패턴을 보인다. 데이터 접근 패턴은 이와 같이 프로그램에 내재되어 있는 특성에 따라 결정되므로 입력되는 데이터의 물리적인 크기와는 무관하다.

또한 프로그램의 복잡도를 결정하는 유일한 요소는 그 프로그램에 입력되는 데이터의 크기이다. 복잡도란 프로그램이 실행되는 동안 발생하는 데이터 접근 횟수로 정의되므로 [13] 프로그램에서의 데이터 접근 횟수는 오직 입력 데이터의 크기에 따라 결정된다. 한편, 프로그램이 다중 프로세서 시스템에서 실행될 때, 각 프로세서가 자신의 캐시에 저장되어 있지 않은 데이터를 필요로 할 때에는 프로세서간의 통신이 발생한다. 이 때, 통신의 횟수는 데이터 접근 횟수의 일부이며, 따라서 이 또한 오직 입력 데이터의 크기에 따르는 함수이다.

이상의 논의를 토대로 다음과 같은 두 가지 유용한 사실을 이끌어 낼 수 있다. 첫째, 데이터 접근의 횟수나 데이터의 공유로 인하여 야기되는 통신의 횟수 및 이와 관련된 모든 수치는 입력 데이터의 크기를 유일한 독립변수로 갖는 종속변수이다. 둘째, 독립변수(=입력 데이터의 크기)와 종속변수(=캐시접근실패 수)를 매개하여 주는 함수는 프로그램에 내재되어 있는 고유한 알고리즘적 특성에 따라 결정된다. 즉, 프로그램의 성능을 의미하는 종속변수인 y는 입력 데이터의 크기를 N이라고 할 때 $y(N) = y(N;a_1, ..., a_M)$로 표현된다. 제 4장에서 소개하는 최소 제곱 추정법과 로버스트 추정법을 사용하여 $\mathbb{A} = (a_1, ..., a_M)$의 값을 알아낼 수 있다. 그러면 이 식에 독립변수에 우리가 원하는 특정한 값을 대입함으로 해서 성능 결과를 쉽게 알 수 있으므로 (식 1)이 바로 예측 모델이 된다.

이상의 모델링 과정은 다음과 같이 요약된다.

① 서로 다른 작은 크기의 입력 데이터$(N_1, ..., N_k)$를 작업부하 프로그램에 부과하여 각 경우마다 캐시접근실패 횟수$(y_1, ..., y_k)$를 측정함으로써 k개의 표본점 $((N_1, y_1), ..., (N_k, y_k))$을 수집한다.

② 제 4장에서 소개하는 최소 제곱 추정법과 로버스트 추정법을 사용하여, $y(N) = y(N;a_1, ..., a_M) = \sum_{k=1}^{M} a_k X_k(N)$을 최적화 하는 $\mathbb{A} = (a_1, ..., a_M)$을 구한다.

③ 위의 식에 우리가 알고자하는 입력 데이터의 크기(N_{large})를 대입하여 이 때의 캐시접근실패 횟수(y_{large})를 예측한다.

4. 통계적 추정(Statistical Estimation)

4.1 곡선 적합(Curve Fitting)

곡선 적합이란 측정된 데이터들을 일정한 좌표공간에 위치시켰을 때, 이 점들에 대하여 최적으로 알맞은 곡선의 함수를 찾는 통계적인 방법이다. 이러한 곡선의 식을 구하는 방법에는 여러 가지가 있으나[3] 본 논문에서는 측정 데이터에 영향을 미치는 독립변수가 단지 프로그램의 입력 데이터의 크기뿐이므로 단일 변수를 위한 곡선 적합법을 고려한다.

일반적으로, 추정식(estimator) $y(x) = y(x;a_1, ..., a_M)$은 (식 2)와 같이 독립 변수 x와 미지의 계수 $a_j (j = 1, ..., M)$를 포함하는 선형조합(linear combination)이다.

$$y(x) = a_1 + a_2 x + ... + a_M x^{M-1} = \sum_{k=1}^{M} a_k X_k(x) \quad \text{(식 2)}$$

(식 2)에서 $X_k(x)$는 기저 함수로써 선형함수는 물론 삼

각함수나 지수 함수 등과 같이 x에 대한 비선형 함수일 수도 있다. (식 2)의 계수들을 찾아내는 방법을 모수 추정(parameter estimation)이라고 하며, 이는 N개의 표본점에 대하여 추정식 $y(x)$가 표본점들의 위치를 최적으로 표현할 수 있도록 M개의 계수 또는 모수의 값들을 조정하는 과정을 말한다. 이론적으로 추정식 $y(x)$가 나타내는 곡선이 모든 표본점들과 정확하게 일치할 확률은 매우 낮다. 본 논문에서는 이 때 발생하는 오차를 최소화하기 위한 방안으로써 최소 제곱 추정법과 로버스트 추정법을 사용한다.

최소 제곱 추정 (Least Square Estimation)

모든 측정된 N개의 표본점 (x_i, y_i)에 대한 측정오류가 $y(x)$ 주변에 형성되는 독립적이고 동일하게 무작위로 분포된 가우시안 정규분포를 갖는다고 가정하자. (식 3)에서와 같이, N과 Δy가 상수일 때, 모든 표본점들의 발생확률은 각 표본점의 발생 확률의 곱과 같다.

$$P = \prod_{i=1}^{N} \left\{ \exp\left[-\frac{1}{2} \left(\frac{y_i - y(x_i)}{\sigma^2} \right)^2 \right] \Delta y \right\} \qquad \text{(식 3)}$$

그러므로 주어진 N개의 표본점들에 가장 근접한 추정식을 구하는 것은 다음의 (식 4)를 최소화 하는 $a_j (j = 1, ..., M)$의 값을 찾는 것과 동일하다.

$$\sum_{i=1}^{N} \left[\frac{y_i - y(x; a_1, ..., a_m)}{\sigma_i} \right]^2 \qquad \text{(식 4)}$$

이에 대한 해결방법은 다음과 같다[3].

$$a_j = \sum_{k=1}^{M} [\alpha]_{jk}^{-1} \beta_k \qquad \text{(식 5)}$$

$$\text{이 때, } \alpha_{kj} = \sum_{i=1}^{N} \frac{X_j(x_i) X_k(x_i)}{{\sigma_i}^2},$$

$$\text{또는 } A_{ij} = \frac{X_j(x_i)}{\sigma_i} \text{ 일 때, } [\alpha] = A^T \cdot A \qquad \text{(식 6)}$$

$$\text{그리고 } \beta_k = \sum_{i=1}^{N} \frac{y_i X_k(x_i)}{{\sigma_i}^2},$$

$$\text{또는 } b_i = \frac{y_i}{\sigma_i} \text{ 일 때, } [\beta] = A^T \cdot b \qquad \text{(식 7)}$$

σ_i는 i번째 표본점의 오차인데, 그 값을 알 수 없다면 일반적으로 $\sigma_i = 1$ 과 같이 상수로 정해질 수 있다[3].

4.3 로버스트 추정(Robust Estimation)

흔하지는 않지만, 표본점을 수집할 때 알지 못하는 원인으로 인하여 일정한 패턴을 따르는 대부분의 표본점과는 달리 그들과 현격하게 격리되어 있는 이탈점(outlier)이 관찰되는 경우가 있다. 이러한 경우 이탈점으로 인하여 추정식이 상당히 왜곡된다. 로버스트 추정식이란 이러한 이탈점의 영향을 최소화함으로써 추정식의 왜곡률을 최소화하기 위한 모수 추정기법이다. 따라서 로버스트 추정식은 추정식이 이상적으로 최적화 되어있다고 가정할 때, 최소 제곱 추정식보다는 이탈점에 덜 민감하다.

(식 3)과는 달리, 로버스트 추정식의 유사함수는 (식 8)과 같다[4].

$$P = \prod_{i=1}^{N} \{ \exp[-\rho(y_i, y(x_i; a_1, ..., a_M))] \Delta y \} \qquad \text{(식 8)}$$

(식 8)에서 ρ는 가중함수로써, 지역적인 추정식의 경우, 실제로 측정된 y_i와 각 점이 어떤 가중 인자 σ_i가 적용될 때 예측되는 $y(x_i)$값 사이의 오차에 의존한다. 널리 사용되는 가중함수에는 여러 가지가 있으나 본 논문에서는 $\rho(z) = ln(1 + z^2)$그리고 $\psi(z) = 2z/(1 + z^2)$ 로 하는 Caushy의 함수를 사용하였다. σ_i를 알 수 없을 경우, 이는 반드시 추정 되어야 하는데 본 논문에서는 [4]에서 소개된 방법 중에서 $\sigma_i = \sqrt{(y_i - y(x_i; a))^2}/N$ 을 사용하였다.

(식 8)의 유사함수를 최대화 하는 것은 $a = a_i, ..., a_M$ 일 때, a에 대하여 다음의 (식 9)를 최소화하는 것과 같다.

$$\sum_{i=1}^{N} \rho\left(\frac{y_i - y(x_i; a)}{\sigma_i} \right) \qquad \text{(식 9)}$$

또한 (식 9)를 최소화하는 것은 다음과 같은 M차 연립방정식의 해법과 같다[4].

$$\psi = \frac{d\rho}{dz} \text{ 일 때, } \sum_{i=1}^{N} \psi\left(\frac{y_i - y(x_i; a)}{\sigma_i} \right) X_j(x_i) = 0,$$

$$j = 1, 2, ..., M \qquad \text{(식 10)}$$

가중 최소 제곱법(weighted least squares method)[4]이라고 불리는 이 해법은 반복적이며 최소 제곱법으로 얻어지는 a의 초기 측정치로부터 시작한다. [4]에서, A가 $A_{ij} = \frac{X_j(x_i)}{\sigma_i}$인 $N \times N$행렬이고, $Y = [y_i]$, W는 (식 11)과 같이 정의되는 w_i의 대각행렬일 때, $a = (A^T W A)^{-1} A^T W Y$ 이 된다.

$$w_i = \frac{(\psi(y_i - y(x_i; a)))/\sigma_i}{(y_i - y(x_i; a))/\sigma_i}, \; i = 1, 2, ..., N \qquad \text{(식 11)}$$

5. 실험 및 예측 모델

5.1 실험 환경 및 시뮬레이션 방법

N이 임의의 프로그램에 입력되는 데이터의 크기를 나타낼 때 때, 프로그램의 복잡도는 *big-oh*[13]표기법을 이용하여 $O(f(N))$과 같이 표현된다. $f(N)$은 특별히 정의된 것은 아니지만, 이상적인 컴퓨터 환경에서 명령어 수 또는 데이터 접근 수를 의미한다[13]. 본 논문에서도 작업부하 프로그램의 데이터 접근수(REF, number of references)와 캐시 접근실패 수를 표현하기 위해 이 방법을 사용한다.

시뮬레이션에서는 Lund University에서 개발되어진 프로그램-구동 시뮬레이터인 CacheMire 패키지를 사용하였다[14]. CacheMire를 사용하면 병렬 프로그램에 대한 명령어의 실행과 데이터에 대한 접근 및 동기화 명령들이 시뮬레이션이 수행되는 동안 실시간으로 얻어진다. 목표시스템은 8개의 프로세서와 무한 크기의 프로세서 캐시, 블록크기는 32KByte, 그리고 Illinois 일관성 프로토콜[8]을 사용하여 캐시접근실패의 수를 측정하였다. 이 때 [8]에서 제안하는 방식대로 최초접근실패(CM, cold miss), 진성공유데이터접근실패(TSM, true sharing miss), 가성공유데이터접근실패(FSM, false sharing miss) 및 교체접근실패로 구분하였다.

〈표 1〉 실험에 사용된 벤치마크와 추정식과 예측을 위한 입력 데이터의 크기

작업부하 프로그램	표본 입력 데이터의 크기	캐시접근실패 예측을 위한 입력 데이터의 크기		공유 데이터
LU	48, 64, 80, 96, 112, 128, 144	256	512	A, L, thisPivot
MP3D	2K, 4K, 8K, 16K, 32K, 64K, 128K	256K	512K	Particles, Cells, Ares
OCEAN	34, 66, 130	258	514	q_multi, rhs_multi 등 25개
FFT	2^6, 2^8, 2^{10}, 2^{12}, 2^{14}	2^{16}	2^{18}	x, trans
BARNES	512, 768, 1K, 1260, 1536, 1792, 2K	4K	8K	bodytab, ctab, ltab
RADIX	2K, 4K, 6K, 8K, 10K, 12K, 14K	512K	1M	key, densitiies, ranks

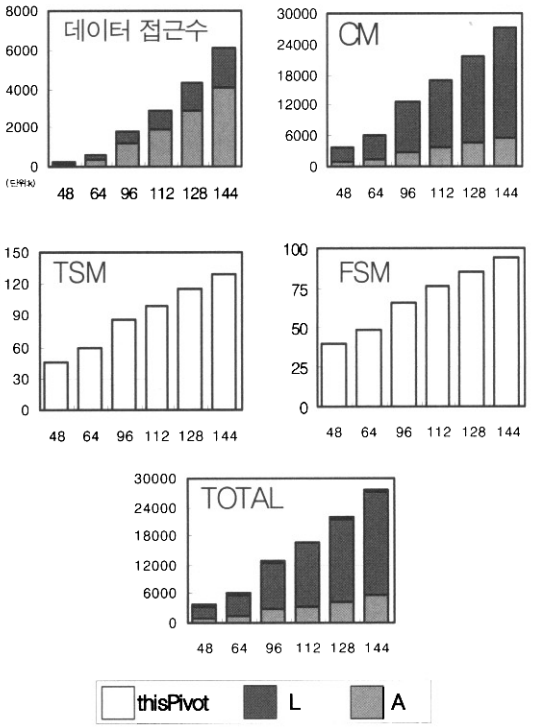
본 논문에서는 SPLASH[6]의 LU, MP3D, OCEAN과 SPLASH-2[7]의 FFT, BARNES, RADIX 등 모두 6개의 벤치마크 프로그램을 시뮬레이션 하였다. 이들 중 4개 프로그램은 7개, OCEAN은 3개, FFT는 5개의 작은 크기의 입력 데이터를 적용하여 표본점을 각각 수집하였다. <표 1>은 각 벤치마크에 사용된 표본점을 수집하기 위한 작은 입력 데이터의 크기, 캐시접근실패 빈도를 예측하고자 하는 큰 입력 데이터의 크기 및 공유 데이터 변수를 보여준다. 표에 나오지 않은 공유 데이터들은 성능에 거의 영향을 미치지 않는다.

다음 절부터는 각 벤치마크들에 대한 설명과 더불어, (그림 1)부터 (그림 6)까지는 입력 데이터의 크기(x-축)에 따르는 데이터접근 수와 각 종류의 캐시접근실패 수(y-축)를 보여준다. <표 2>부터 <표 7>에서는 곡선 적합에 의한 예측 모델과 함께 큰 크기의 데이터를 입력했을 때에 대한 예측 결과를 보여준다. 표에서 수학적 다항식으로 표현되는 예측 모델들은 시뮬레이션을 통하여 수집된 표본점들에 본 논문의 제 4장에서 소개한 통계적 추정기법을 C-언어를 사용하여 자체적으로 구현한 프로그램을 사용하여 구하였다. 한편, 예측 오차율은 다음과 같이 계산되었다.

$$\text{예측 오차율 } (\%) = \frac{\text{시뮬레이션 결과} - \text{예측된 결과}}{\text{시뮬레이션 결과}} \times 100 \quad \text{(식 10)}$$

5.2 LU

LU는 밀집정방행렬의 LU-분해를 수행하는 프로그램이다. N은 행렬의 행(또는 열)의 수이며 프로그램내의 *for-loop*가 반복되는 횟수이기도 하다. 공유 데이터는 두개의 $N \times N$ 행렬인 A, L과 하나의 N크기 벡터(*thisPivot*)가 있다. 그러므로 $O(N^2)$의 최초접근실패가 예측된다. *for-loop*에서 $i = 0, 1, ..., N-1$으로 변함에 따라, i번째 반복시기에서 A는 i번째 열부터 우측의 $N-i$번째 열까지 한번씩 접근되며, L의 i번째 열은 $N-i$번 접근되어진다. 그러므로 이들에 대한 데이터 접근 수는 $O(N^3)$이다. 두 개의 큰 배열 A와 L에 대한 접근은 최초접근실패만을 야기하고 *thisPivot*에 대한 접근에서 $O(N)$개의 공유데이터접근실패가 발생한다.



(그림 1) LU의 시뮬레이션 결과

〈표 2-(a)〉 예측 모델: 추정식

항목	규모	예측 모델
REF	$O(N^3)$	$1.994N^3+8.764N^2-178.70N+3849.08$
Total Misses	$O(N^2)$	$1.250N^2+12.988N-63.308$
CM	O(N)	$1.250N^2+7.030N-36.233$
TSM	O(N)	$5.396N-14.143$
FSM	$O(N^2)$	$0.562N-12.932$
캐시접근 실패율		$\frac{1.250N^2+12.988N-63.308}{1.994N^3+8.764N^2-178.70N+3849.08}$

〈표 2-(b)〉 예측 모델과 예측 결과

LU	큰 입력 데이터의 크기 및 예측 결과					
	288 × 288			512 × 512		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	48358061	48321296	0.076	270273517	269893536	0.140
Total Misses	107349	107349	0.000	334255	334247	0.002
CM	105669	105662	0.005	331243	331225	0.005
TSM	1532	1539	-0.457	2736	2748	-0.439
FSM	148	148	0.000	276	274	0.725
캐시접근 실패율	0.2220	0.2220	0.000	0.1237	0.1238	-0.080

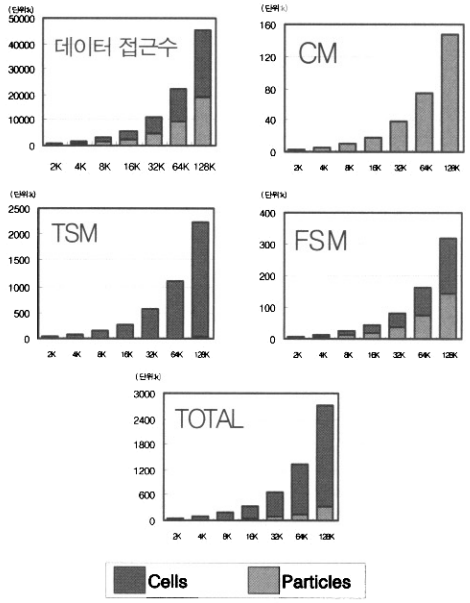
<표 2>에서 보듯이, 큰 크기의 입력 데이터에 대한 예측 오차는 극히 작다. 이것은 LU가 가지고 있는 알고리즘적인 특성으로 인하여 데이터 접근패턴과 프로세스간의 통신이 매우 규칙적이기 때문에 예측이 쉽기 때문이다.

5.3 MP3D

MP3D는 우주공간에서의 입자의 운동에 대하여 *Particles* 배열 안에 있는 분자들의 위치와 속도를 반복적으로 계산하는 시뮬레이션 프로그램이다. 입력 데이터 크기 N은 1차원 배열 *Particles*안의 분자 수를 나타낸다. 또 다른 1차원 배열 *Ares*는 시간이 경과함에 따라 추가되는 분자들을 나타내며 이배열의 크기 또한 N에 비례한다. 3차원 좌표공간을 이산적으로 나타내기 위한 배열 *Cells*의 크기는 N에 무관한 고정 크기이다. 그러므로 MP3D의 최초접근실패의 수는 $O(N)$으로 예측된다.

이 프로그램 안에서의 시뮬레이션 반복회수는 5회($O(1)$)이며, 각 반복시기마다 *Particles* 배열 안의 모든 분자들이 접근되고 *Cells* 배열 공간 안에서 이동 한다($O(N)$ 접근). 분자가 움직일 때 마다 3차원 좌표상의 입자들의 위치 변화를 나타내기위하여 *Cells* 요소들이 갱신되므로, 매 반복시기마다 *Cells*의 각 요소들은 $O(N)$번 접근된다. *Ares* 배열의 각 분자들 또한 *Cells* 배열 공간 안에서 이동하여 단 한번의 충돌검사를 수행한다($O(N)$회 접근). 그러므로 MP3D에서 전체적인 데이터 참조회수는 $O(N)$이고 진성공유데이터 접근실패 수는 또한 $O(N)$이다.

프로세서가 접근하는 공유데이터의 메모리에서의 위치는 대부분 예측이 불가능하다. 하나의 프로세서가 하나의 분자를 *Cells* 공간 안에서 이동시킬 때, 임의로 생성된 수의 값에 따라 프로세서가 충돌을 시험해야 하는지 말아야 하는지가 결정되기 때문이다. 이러한 충돌들은 *Particles* 배열에서 공유데이터접근실패를 발생시키는 원인이다. 더욱이 *Cells* 배열의 각 원소의 메모리 상의 위치는 분자의 공간좌표에 의해 선택되고 프로세서나 분자의 메모리 상의 위치와는 무관하다. 이러한 불규칙적인 특성 때문에 심각한 예측 오류가 가능하다. 그러나 로버스트 추정식은 이러한 불규칙성을 통계적인 처리방법으로 잘 극복하였다. <표 3>에서 오차율이 1% 미만의 매우 정확한 결과를 보여주고 있다.



(그림 2) MP3D의 시뮬레이션 결과

〈표 3-(a)〉 예측 모델: 추정식

항목	규모	예측 모델
REF	O(N)	$348.845N+41097.4$
Total Misses	O(N)	$21.452N-3650.30$
CM	O(N)	$1.522N-23.046$
TSM	O(N)	$17.321N-1507.18$
FSM	O(N)	$2.609+5180.57$
캐시접근실패율		$\frac{21.452N-3650.3}{348.845N+41097.4}$

〈표 3-(b)〉 예측 모델과 예측 결과

MP3D	큰 입력 데이터의 크기 및 예측 결과					
	256K개의 분자			512K개의 분자		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	91485470	91488736	-0.003	182937948	182936384	0.000
Total Misses	5625800	5627779	-0.035	11219864	11251459	-0.281
CM	398842	399127	-0.071	798626	798277	0.043
TSM	4542127	4539559	0.056	9102790	9180176	0.248
FSM	684831	689093	-0.622	1318448	1373006	-4.138
캐시접근 실패율	6.1494	6.1513	-0.030	6.1332	6.1504	-0.280

5.4 OCEAN

OCEAN은 큰 규모의 바다에서 물분자의 운동을 움직임을 일정한 횟수만큼 반복적으로 시뮬레이션 하는 프로그램이다. 각 반복시기마다 크기가 고정된 여러 개의 2차원 격자 모양으로 이산적으로 모델링된 공간 배열을 이용하여 다양한 미분 방정식이 계산되며, 각 격자는 해양분지를 수평으로 나눈 구역을 나타낸다. 이 벤치마크에 사용된 알고리즘은 다중 격자 Gauss-Seidel의 Successive Over Relaxation 방법이다. 주요 데이터 구조는 물분자의 운동을 모델링하기 위한 방정식과 연관된 이산 데이터를 저장하는 25개의 2차원 배열이다. 그 중에서 *q_multi*와 *rhs_multi*는 여러 층으로 이루어진 격자를 나타내는 배열이며 각각의 크기는 $O(N^2)$이다. 각 반복시기마다 $O(N^2)$의 두 격자배열이 접근된다. 그러므로 데이터 접근의 수는 $O(N^2)$이다. 프로세서간의 데이터 공유는 오직 각 프로세서에게 할당된 데이터의 구획 경계에서만 나타나므로 진성공유데이터접근실패의 수는 $O(N^2)$이다.

OCEAN은 두 가지 측면에서 분석적으로 모델링하기 까다로운 프로그램이다. 우선, 프로그램에서 사용하는 데이터가 많아 많은 양의 메모리를 요구한다. 이로 인하여 본 연구에서는 단지 세 개의 작은 입력 데이터에 대한 표본점만을 수집할 수 있었다. 이것은 2차 다항식에 있는 세 개의 미지의 계수를 추정하기위한 최소한의 요구량이다. 뿐만 아니라, OCEAN의 수행시간은 입력 데이터의 값에 크게 의존하는데 이는 계산되어 갱신된 수치들의 변화량의 총합이 일정한 임계치보다 작을 때 까지 반복적으로 수행되기 때문이다. 반복 횟수도 입력 데이터의 값에 따라 다르므로 표본점들의 측정오차가 상당히 크다. 더욱이 수집된 표본점이 최소 요구량인 단 3개밖에 없으므로 예측모델의 정확도가 매우 낮을 것이다. 만일 예측하고자 하는 큰 입력 데이터에 대하여 OCEAN 프로그램이 불규칙적인 반응을 보인다면 예측 결과 또한 나쁘게 된다. 그럼에도 불구하고, 우수한 캐시 접근실패율에 대한 예측 오차율을 보이는 모델을 도출할 수 있었다. 두 번의 예측 오차율이 공히 4%미만이다.

〈표 4-(a)〉 예측 모델: 추정식

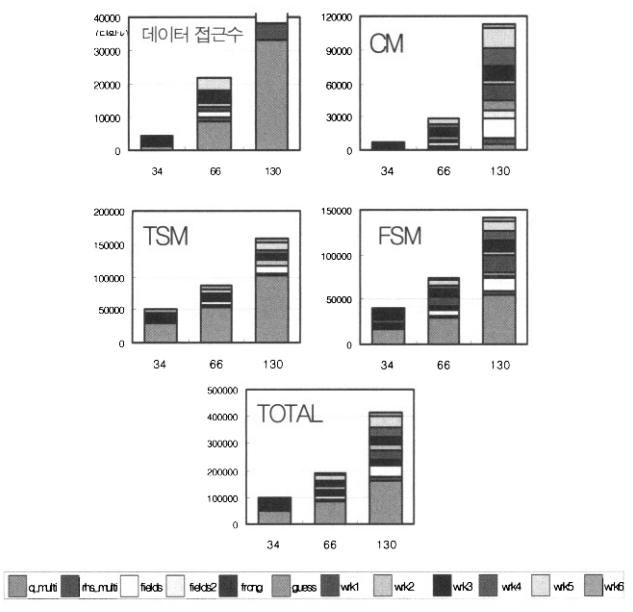
항목	규모	예측 모델
REF	$O(N^2)$	$4576.93N^2-53728.1N-1562387$
Total Misses	$O(N^2)$	$6.911N^2+2137.86N+9554.16$
CM	$O(N^2)$	$6.911N^2+79.348N+229.352$
TSM	O(N)	$1021.86N+6299.61$
FSM	O(N)	$1036.65N+3052.20$
캐시접근 실패율		$\frac{6.911N^2+2137.86N+9554.16}{4576.93N^2-53728.1N-1562387}$

〈표 4-(b)〉 예측 모델과 예측 결과

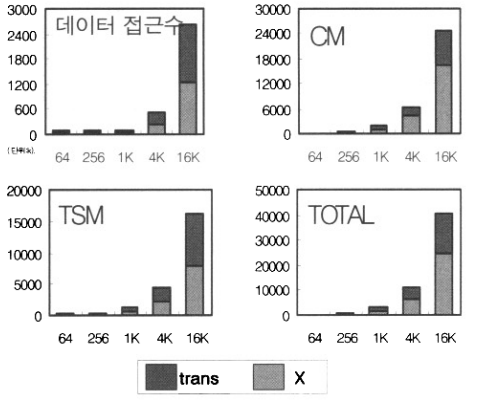
OCEAN	큰 입력 데이터의 크기 및 예측 결과					
	258 × 258			514 × 514		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	329996291	316955936	3.951	1277460884	1235256448	3.303
Total Misses	1030064	1021176	0.862	2919052	2934399	-0.525
CM	482821	480757	0.013	1867445	1867001	0.023
TSM	275316	269940	1.952	519512	531538	-2.315
FSM	273927	270479	1.258	532095	535860	-0.707
캐시접근 실패율	0.312	0.322	-3.204	0.228	0.237	-3.938

5.5 FFT

FFT는 radix-$\sqrt{N}$ 여섯 단계 FFT 알고리즘의 1차원 버전이다. 중요한 입력 데이터는 두 개의 $\sqrt{N}\times\sqrt{N}$ 배열로써, X는 데이터 포인트를, *trans*는 roots-of-unity를 담고 있다. 작은 크기의 1차원 배열인 *umain*과 *umain2*는 공유되지 않으므로 캐시접근실패의 수는 무시한다. $\sqrt{N}\times\sqrt{N}$의 배열을 반영하기 위하여 입력 데이터의 크기를 $n=\sqrt{N}$으로 한다. FFT는 반복적이지 않으며 X행렬과 *trans*행렬($O(N^2)$ 크기)의 각 원소들은 $O(1)$번 접근된다. 그러므로 전



(그림 3) OCEAN의 시뮬레이션 결과



(그림 4) FFT의 시뮬레이션 결과

〈표 5-(a)〉 예측 모델: 추정식

항목	규모	예측 모델
REF	$O(n^2)$	$197.952n^2-2614.8n+24184.5$
Total Misses	$O(n^2)$	$2.315n^2+22.908n-0.672$
CM	$O(n^2)$	$1.878n^2+15.903n-0.474$
TSM	$O(n^2)$	$0.437n^2+7.005n-0.198$
FSM	0	0
캐시접근 실패율		$\frac{2.315n^2+22.908n-0.672}{197.952n^2-2614.8n+24184.5}$

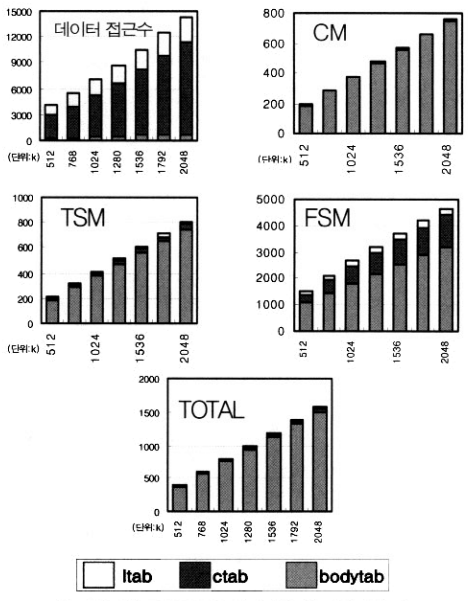
〈표 5-(b)〉 예측 모델과 예측 결과

FFT	큰 입력 데이터의 크기 및 예측 결과					
	2^{16}			2^{18}		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	11993088	12327778	-2.790	52166656	50577336	3.046
Total Misses	157692	157632	0.038	619504	618805	0.112
CM	127228	127169	0.045	501232	500538	0.138
TSM	30464	30463	0.001	118272	118267	0.003
FSM	0	0	0	0	0	0
캐시접근 실패율	1.3149	1.2787	2.753	1.1875	1.2234	-3.023

체 데이터 접근 수와 최초접근실패 수는 $O(N^2)$이고, 진성공유데이터접근실패 수 역시 $O(N^2)$이다. <표 5>에서 보듯이, 예측된 데이터접근실패율과 실측된 데이터접근실패율에 대한 오차는 4% 미만이다.

5.6 BARNES

BARNES는 Barnes-Hut의 계층적인 N-body알고리즘을 사용하여 3차원 공간에서 입자들의 이동을 일정한 횟수만큼 반복하여 시뮬레이션 하는 프로그램이다. 크기 N의 1차원



(그림 5) BARNES의 시뮬레이션 결과

〈표 6-(a)〉 예측 모델: 추정식

항목	규모	예측 모델
REF	O(N*log*N)	$1491.8N\log N+3278.6N-2741459\log N+6796905$
Total Misses	O(N+*log*N)	$34.943N+11965.7\log N-24249.6$
CM	O(N+*log*N)	$28.532N-108.061\log N+4471.05$
TSM	O(N+*log*N)	$5.088N+10919.2\log N-25920.3$
FSM	O(N+*log*N)	$1.323N+1154.58\log N-2800.35$
캐시접근 실패율		$\frac{34.943N+11965.7\log N-24249.6}{1491.8N\log N+3278.6N-2741459\log N+6796905}$

〈표 6-(b)〉 예측 모델과 예측 결과

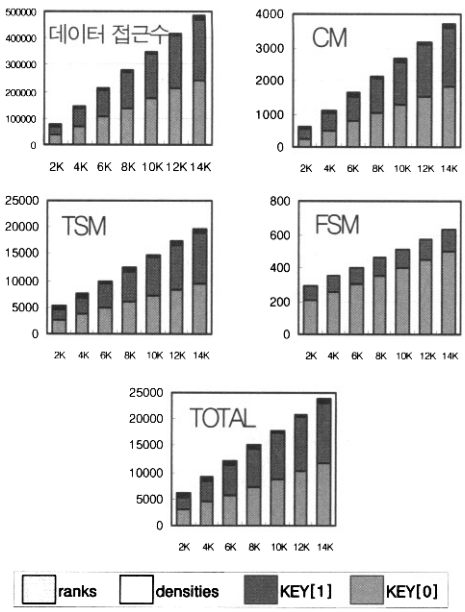
BARNES	큰 입력 데이터의 크기 및 예측 결과					
	4096			8192		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	33733353	33701928	0.093	74726397	74732568	-0.008
Total Misses	162663	166743	-2.592	315040	314845	0.061
CM	115856	120907	-4.360	239227	237740	0.062
TSM	39415	38597	2.073	62819	63078	-0.413
FSM	7392	7239	2.062	12994	13045	-0.400
캐시접근 실패율	0.482	0.497	-2.592	0.421	0.421	0.071

배열 *bodytab*은 시뮬레이션 될 입자들의 정보를 담고 있다. *ltab*배열과 *ctab*배열은 octree구조이다. $O(N)$개의 원소로 구성된 *ltab*은 트리의 터미널 노드에 있는 입자들에 대한 배열이고, $O(\log N)$개 요소로 구성된 *ctab*은 트리의 터미널 노드가 아닌 노드에 대한 배열이다. 세 가지 배열의 크기에 따라 최초접근실패 수는 $O(N+\log N)$이다. 각 반복시기마다 *bodytab*의 모든 $O(N)$개의 입자들이 octree의 $O(\log N)$개 비터미널 *ctab*의 원소들을 부분적으로 순회하고 나서 *ltab*원소들에 접근한다. 그러므로 *ctab*과 *bodytab*에 대한 참조는 각각 $O(N\log N)$과 $O(N)$이다. 전체 데이터 접근 수는 $O(N\log N)+O(N)+O(\log N)+O(1)$이고 진성공유데이터접근실패 수는 $O(N)+O(\log N)+O(1)$이다.

5.7 RADIX

RADIX는 정수형 데이터에 대한 radix sort를 수행하는 프로그램이다. 각 반복시기마다 키 값들의 r-번째 비트를 근으로 하여 정렬한다. 대부분의 데이터 접근은 1차원 배열인 *key[0]*와 *key[1]*의 배열에서 이루어지므로 최초접근실패의 수는 $O(N)$이다. 반복의 회수는 가장 긴 비트의 길이의 키와 그 근에 의존하는데, 두 인자 모두 프로그램 실행 중에 결정되므로 반복회수는 일정하지 않다. 매 반복시기마다 접근되는 *key[0]*과 *key[1]*배열은 지역적인 히스토그램을 계산하기 위해 접근되고 부분적으로 정렬되어 다른 배열에 저장된다. 그러므로 데이터 접근 수와 진성공유데이터접근실패 수는 모두 $O(N)$이다.

가성공유데이터접근실패의 경우는 진성공유데이터접근실패보다 예측이 훨씬 어렵다. 실제로 *key*에 대한 가성공유데



(그림 6) RADIX의 시뮬레이션 결과

〈표 7-(a)〉 예측 모델: 추정식

항목	규모	예측 모델
REF	O(N)	34.009N+219.030
Total Misses	O(N)	1.437N+3575.31
CM	O(N)	0.749N+1416.06
TSM	O(N)	0.684N+1640.14
FSM	O(N)	0.004N+519.106
캐시접근 실패율		1.437N+3575.31 / 34.009N+219.030

〈표 7-(b)〉 예측 모델과 예측 결과

RADIX	큰 입력 데이터의 크기 및 예측 결과					
	4096			8192		
	실험 결과	예측 결과	오차율 (%)	실험 결과	예측 결과	오차율 (%)
REF	17301504	17831184	-3.061	34603008	35662148	-3.060
Total Misses	755830	756904	-0.142	1509531	1510233	-0.046
CM	393235	394264	-0.261	786957	787113	-0.019
TSM	359634	360205	-0.158	718224	718769	-0.076
FSM	2961	2435	17.752	4350	4351	-0.037
캐시접근 실패율	4.368	4.225	2.833	4.362	4.234	2.924

이터접근실패의 예측은 어렵지만 발생수가 매우 작으므로 전체적인 예측결과에 큰 영향을 미치지 않는다.

5.8 다른 벤치마크 프로그램

본 논문에서 선택한 벤치마크 프로그램들은 SPLASH[6]와 SPLASH-2[7] 패키지에 포함된 것이다. SPLASH의 프로그램 중에서 Pthor와 LocusRoute는 회로 입력 값에 대한 명세가 없어서 충분한 크기의 입력 데이터를 구할 수 없었으므로 실험할 수 없었다. 또한 SPLASH에 있는 Cholesky, Radiosity와 기타 SPLASH-2 프로그램들은 프로세스의 실행이 동적으로 스케줄되는 특성을 가지므로 본 논문에서 제안하는 방법으로는 모델링이 불가능하다.

6. 결 론

과학이나 공학 분야의 응용 프로그램들의 경우, 프로그램의 복잡도를 결정하는 요소는 오직 그 프로그램에 사용되는 입력 데이터의 크기이며, 일반적으로 입력 데이터 크기가 변화함에 따라 데이터 접근 양식에는 일정한 패턴이 유지된다. 이러한 사실을 토대로 몇 개의 표본점을 수집한 후 통계적인 추정방법을 적용하여 다중 프로세서 시스템에서 발생하는 데이터 통신 빈도에 대한 모델링 기법을 제안하였다. 이 방법으로 구축된 모델은 목표시스템에서의 작업부하의 알고리즘적인 동적 특성이 수학적인 공식으로 반영되므로 대용량의 입력 데이터를 부과했을 때의 통신 빈도를 정확히 예측할 수 있었다.

프로그램의 복잡도가 입력 데이터의 크기에 대한 다항함수로 표현될 때, 표본점들에 대한 적합곡선을 구하기 위하여, 로버스트 파라미터 추정기법을 사용하였으며, 그 결과 매우 정확한 캐시접근실패율에 대한 예측을 할 수 있었다. 이러한 모델링 전략은 프로그램의 실행 중 행동양식이 상당히 불규칙하거나(MP3D, OCEAN, BARNES), 표본점의 수가 충분하기에는 부족한 경우에도(OCEAN, FFT) 매우 잘 적용된다. 예측오차는 모두 4% 미만이며 절반의 경우 1% 미만의 오차를 보인다. 단, 프로세스의 실행순서가 동적으로 결정되는 경우에는, 데이터 접근 양식에 있어서 일정한 패턴이 유지되지 않으므로, 본 모델링 방법이 적용되지 않는다.

공유메모리 시스템에서 캐시접근실패 이외에도 데이터 무효화 신호나 블록을 메모리에 되쓰는 경우가 발생하기도 하지만 이들은 캐시접근실패보다는 빈도가 상대적으로 매우 낮으므로 본 연구에서는 다루지 않았다. 하지만, 이들의 발생원인도 캐시접근실패가 발생하는 원인과 밀접하게 관련되므로 본 논문에서 제안하는 예측모델을 적용할 경우 동일한 수준의 정확도를 보일 것이다.

지금까지 그림에 나타난 그래프들에서 보여주듯이, 벤치마크 안의 공유 변수들의 공유효과는 매우 상이하다. 만일 각각의 변수에 대한 동적 특성을 고려한 모델링이 사용되었다면 모델을 유도해내는 과정은 보다 복잡해졌을 것이지만, 보다 정확한 예측을 할 수 있었을 것이다. 또한 본 논문에서는 다중 프로세서 시스템에서 입력 데이터의 크기에 따른 데이터 통신의 빈도의 변화를 다루었다. 이는 다중 프로세서 시스템에서 고정된 수의 프로세서 환경에서 메모리의 크기의 증가에 따른 것이다. 현재, 보다 완벽한 성능의 모델을 만들기 위하여, 프로세서의 수의 변화를 반영할 수 있는 모델링에 대한 연구를 하고 있다.

끝으로, 본 논문에서 제안하는 공유메모리 기반의 다중프로세서 시스템에서의 데이터 통신에 대한 예측 모델을 구축하는 방법은 분산 구조를 가진 병렬 컴퓨터 시스템에도 그대로 적용될 수 있다. 특히 메시지-패싱 기법을 사용하는 분산 시스템에서는 프로그램의 원천 코드에 데이터 통신을 유발하는 send와 receive 명령어들이 명시적으로 사용되므로 데이터 통신의 양을 측정하거나 예측하기가 더욱 쉽다. 이러한 이유로, 분산 시스템에 본 논문에서 제안하는 모델을 적용하였을 경우 예측결과의 품질에 대한 평가가 낮아질 수 있으므로, 본 논문에서는, 목표 시스템을 공유메모리 기반의 대칭형 다중프로세서 시스템으로 정하였다.

참 고 문 헌

[1] D. Ferrari, "Computer Systems Performance Evaluation," Prentice-Hall.

[2] J. Tsai, A. Agarwal, "Analyzing Multiprocessor Cache Behavior Through Data Reference Modeling," Proc. of ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, pp.236-247, May, 1990.

[3] W. I. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, "Numerical Recipes," Cambridge University Press.

[4] R. L. Launer, and G. N. Wilkinson, "Robustness in Statistics," Academy Press.

[5] J. P. Singh, J. L. Hennessy, A. Gupta, "Scaling Parallel Programs for Multiprocessors: Methodology and Examples," IEEE Computer, pp.42-50, 1993.

[6] J. P. Singh, W-D Weber, A. Gupta, "SPLASH: Stanford Parallel Applications for Shared-Memory," Computer Architecture News, 20(1):5-44, March, 1992.

[7] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, A. A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Consideration," Proc. of 22nd Ann. Int. Symp. on Computer Architecture, pp.24-36, May, 1995.

[8] M. Dubois, J. Skeppstedt, L. Ricciulli, K. Ramamurthy, and P. Stenstrom, "Detection and Elimination of Useless Misses in Multiprocessors," Proc. of 20th Ann. Int. Symp. on Computer Architecture, pp.88-97, May, 1993.

[9] D. Kerbyson, A. Hoisie, H. Wasserman, "Modeling the Performance of Large-Scale Systems," IEEE Proc. on Software, 150(4), pp.214-221, Aug., 2003.

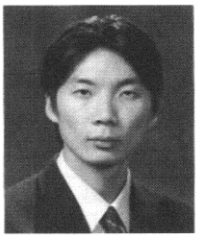
[10] G. Weerasinghe, I. Antonios, L. Lipsky, "An Analytic Performance Model of Parallel Systems that Perform N Tasks Using P Processors that can Fail," IEEE Int'l. Symp. on Network Computing and Applications, pp.310-319, 2001.

[11] M. Dubois, J. C. Wang, "Shared Block Contention in a Cache Coherence protocol," IEEE Transactions on Computers, Vol. 40, No.5, May, 1991.

[12] I. Gluhovsky, B. O'Krafta, "Comprehensive Multiprocessor Cache Miss Rate Generation using Multivariate Models," *To appear* ACM Trans. on Computer Systems, Vol.23, No.2, pp. 111-145, May, 2005.

[13] V. A. Aho, J. E. Hopcroft, J. D. Ullman, "Data Structures and Algorithms," Addison-Wesley Publishing Company.

[14] M. Brorsson, F. Dahlgren, H. Nilsson, P. Stenstrom, "The CacheMire Test Bench — A Flexible and Effective Approach for Simulation of Multiprocessors," Proc. of 26th Ann. IEEE International Simulation Symposium, pp.41-49 Apr., 1993.



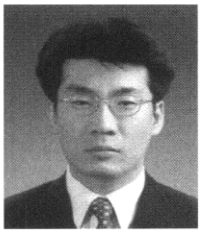
전 장 환

e-mail : b4crazy@dgu.edu

2004년 동국대학교 정보통신공학과(학사)

2004년~현재 동국대학교 정보통신공학과 (석사과정)

관심분야 : 컴퓨터 구조, 임베디드 시스템, 네트워크 컴퓨팅



이 강 우

e-mail : klee@dgu.ac.kr

1985년 연세대학교 전자공학과(학사)

1991년 University of Southern California 컴퓨터 공학(공학석사)

1997년 University of Southern California 컴퓨터 공학(공학박사)

1998~현재 동국대학교 정보통신공학과 부교수

관심분야 : 컴퓨터 구조, 임베디드 시스템, 네트워크 컴퓨팅