데스크탑 그리드 시스템에서 자원의 가용성과 신뢰도를 이용한 결과 검증 메커니즘♣

김홍수*, 길준민+, 윤준원╪, 최장원╪, 김성석╫, 이상근*
*고려대학교 컴퓨터학과
+대구가톨릭대학교 컴퓨터교육과
╪한국과학기술정보연구원 슈퍼컴퓨팅센터
╫서경대학교 전자상거래학과
e-mail : hera@disys.korea.ac.kr*, jmgil@cu.ac.kr+, jwyoon@kisti.re.kr╪, jwchoi@kisti.re.kr╪, sskim03@skuniv.ac.kr╫, yalphy@korea.ac.kr*

Result Verification Mechanism using Volunteer Availability and Reliability in Desktop Grid Systems

Hong-Soo Kim*, Joon-Min Gil+, Jun-Weon Yoon╪, Jang-Won Choi╪, Sung-Suk Kim╫, Sang-Keun Lee*
*Dept. of Computer Science and Engineering, Korea University
+Dept. of Computer Science Education, Catholic University of Daegu
╪Supercomputing center, Korea Institute of Science and Technology Information
╫Dept. of E-Business, Seokyeong University

요 약

데스크탑 그리드 시스템에서 각 자원에서 수행한 작업 결과에 대한 정확성 검증은 중요한 문제이다. 작업 결과에 대한 정확성을 보장하기 위해서 결과 검증 기법으로 투표기반 기법이나 신뢰기반 기법이 사용되어 왔다. 그러나 이러한 기법들은 동적인 연산 환경에 효과적으로 대처하지 못하여 낮은 확장성과 높은 연산 지연을 초래할 수 있는 단점을 갖고 있다. 이러한 단점을 해결하고자 본 논문에서는 자원제공자에 의해 초래될 수 있는 잘못된 연산 결과에 대해 각 자원제공자의 신용도(credibility)를 동적으로 평가하여 연산 결과의 정확성을 판단하는 적응적 결과 검증 메커니즘(Adaptive Result Verification Mechanism: ARVM)을 제안한다. 또한, 제안하는 ARVM 은 자원제공자의 가용성(availability) 정보를 이용하여 결과 검증 시에 요구되는 연산 지연을 최소화한다.

1. 서론

데스크탑 그리드 컴퓨팅은 인터넷에 연결된 데스크탑 컴퓨터들의 유휴 컴퓨팅 시간을 이용하여 고성능 과학 응용을 수행하기 위한 컴퓨팅 패러다임이다. 데스크탑 그리드 컴퓨팅은 자원제공자 또는 작업자라고 불리는 연산 노드들이 휘발적 속성에 따라서 자유롭게 연산에 참여하고 탈퇴하는 특징을 가진다. 또한, 각 노드들은 개별적인 관리자가 존재하고 다른 컴퓨팅 환경들(성능, 네트워크, 휘발성 등)을 가지는 이질적인 환경이다.

데스크탑 그리드 시스템 (DGS)은 악의적인 자원제공자들의 잘못된 연산 결과에 의해 야기되는 사보타주(Saboteur)에 노출되어 있다. 만약 악의적인 자원제공자가 서버에 잘못된 연산 결과를 보낸다면, 하나의 응용에 해당하는 모든 연산 결과는 취소된다. 예를 들어, SETI@Home 은 완료된 연산 결과의 수를 속이는 소수 자원제공자들의 악의적인 행동에 피해를 본 사례가 보고되었다[2, 6].

기존 연구에서 연산 결과에 대한 결과 검증은 주로 투표기반 결과 검사 기법과 신뢰기반 결과 검사 기법이 사용되어져 왔다. 먼저, 투표기반 결과 검사 기법은 몇몇 노드들(일반적으로 3 개 이상의 노드에게 할당)에게 복제된 작업을 중복하여 수행하게 하고 각각 수행된 결과를 비교하는 방법을 사용한다. 만약 같은 연산 결과가 과반수 이상이라면, 최종 연산 결과로 받아들인다. 이 기법은 간단하고 직관적이지만 연산 자원들의 낭비로 인해 비효율적이다. 반면에 신뢰기반 기법들 [3, 5]은 기본적으로 복제를 하지 않고 자원제공자들의 신뢰성 값을 이용해서 해당 자원제공자에 의해 수행된 결과를 판단하는 기준으로 삼는다. 자원제공자들의 신뢰성 값은 작은 문제를 (또는 샘플) 주기적으로 보내어 그 결과를 이미 알려져 있는 연산 결과와 비교를 통해서 계산된다. 이러한 결과 검사 기법들은 FCFS (First-Come First Serve) 스케줄링[1]을 기반으로 수행이 되는데 결과 검증 관점에서 FCFS 스케줄링 기법은 자원제공자들의 참여와 탈퇴가 자유로운 동적인 연산 환경에서는 높은 연산 지연을 초래한다.

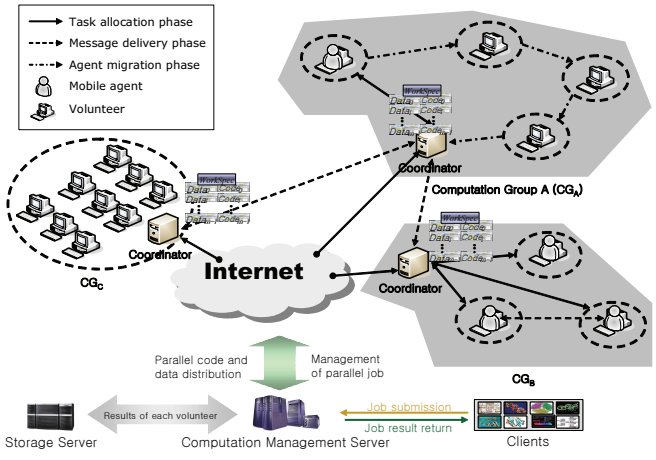
이러한 문제점을 극복하기 위하여 본 논문에서는 적응적 결과 검증 메커니즘(Adaptive Result Verification Mechanism: ARVM)을 제안한다. ARVM 에서는 각 자원제공자들의 신용도와 가용성을 평가하기 위해 이동에이전트에 기반한 랜덤 샘플링 방법을 이용한다. 이 방법은 자원제공자가 연산을 수행하기 이전에 결과가 이미 알려진 샘플 작

♣ 본 연구는 한국과학기술정보연구원(KISTI) " 컴퓨터연계활용 기반구축(Korea@Home 부문)" 사업의 지원에 의해서 수행되었음.

업 객체를 미리 자원제공자에게 분배하고, 해당 자원제공자에 의해 계산된 샘플 작업의 결과를 원래 작업 결과와 비교하여 자원제공자의 신용도를 판단한다. 이때, 자원제공자의 가용성도 함께 평가된다. 이런 방식으로 얻어진 신용도와 가용성 값은 자원제공자를 분류하고 연산 그룹을 구성하는 요소로서 사용된다. 따라서, ARVM 은 이러한 자원제공자의 가용성과 신용도 정보를 작업 할당에 활용함으로써 결과에 대한 정확성을 보장하고 결과 검증으로 인한 연산 지연을 줄일 수 있다.

2. 데스크탑 그리드 시스템 모델

그림 1 은 본 논문의 시스템 환경에 대한 전체 구조를 보여준다. 이 그림에서 보는 것처럼 DGS 모델은 클라이언트(client), 연산 관리 서버(CMS: Computation Management Server), 저장 서버(storage server), 조정자(coordinator), 그리고 자원제공자(volunteer)로 구성된다. 클라이언트는 CMS 에 자신의 응용을 위탁한다. 조정자는 스케줄링, 연산 그룹 관리, 그리고 에이전트 관리 등의 역할을 수행한다. 자원제공자는 자신의 유휴 컴퓨팅 시간에 그리드 응용을 수행하기 위하여 자신의 자원을 제공한다.

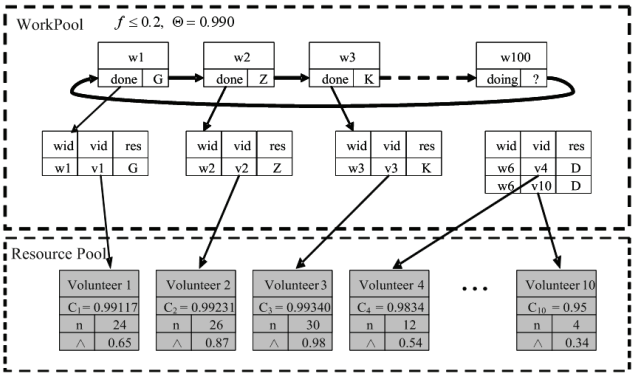


(그림 1) 데스크탑 그리드 컴퓨팅 환경

본 논문의 DGS 는 다음과 같은 단계로 수행된다:

1. ***자원 등록 단계***: 자원제공자들은 CMS 로 자신의 정보들(CPU 능력, 메모리 능력, OS 유형 등)을 등록한다. CMS 는 해당 정보를 해당 조정자에게 보낸다.
2. ***응용 위탁 단계***: 클라이언트는 CMS 에게 응용을 위탁한다.
3. ***작업 할당 단계***: CMS 는 작업을 분류한 후 각 조정자들에게 작업 묶음을 할당한다.
4. ***부하 균형 단계***: 이 단계는 주기적으로 이루어지는데, 하나의 응용 전체에 대한 수행 시간을 높이기 위해서 조정자 간에 작업의 양을 조정한다.
5. ***스케줄링 단계***: 조정자는 자신의 자원 풀과 작업 풀을 기반으로 작업을 자원에게 할당한다.
6. ***결과 수집 단계***: 자원제공자는 작업을 수행한 후 그 결과를 조정자에게 되돌려준다.
7. ***응용 완료 단계***: 조정자는 자신에게 할당된 응용을 모두 수행하면 CMS 에게 결과와 함께 완료 메시지를 보낸다.

본 논문에서는 작업 풀 기반 분산 마스터-워커 모델을 가정한다. 이 모델은 조정자에 의해 관리하는 분산된 연산 그룹을 기반으로 하기 때문에 인터넷기반 그리드 시스템에서 좀 더 확장적이면서 신뢰적인 연산 수행이 가능하다. 그림 1 에서 본 것처럼, 응용은 상호독립적으로 구성되는 순차적인 작업 객체들로 분할된다. 또한, 본 논문에서는 각기 다른 데이터에 대해서 동일한 코드를 수행하는 SPMD (Single-Program Multiple-Data) 모델을 가정한다.



(그림 2) 신용도와 가용성 기반의 작업 풀과 자원 풀 모델

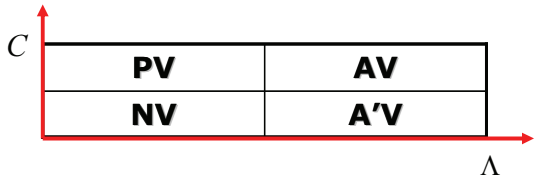
3. 결과 검증 기법

3.1 개요

본 논문에서는 이동 에이전트를 이용해서 연산을 수행하는 주체로 이용하고 자원제공자의 가용성과 신용도를 측정하는 도구로 이용한다. "ODDUGI" [7]는 이동에이전트 시스템으로서 DGS 환경에서 연산 그룹을 지칭하는 다중 지역으로 구성이 된다. 본 시스템은 이동에이전트, 자원제공자, 연산 그룹, 그룹 조정자, 응용 서버로 구성된다. 이러한 환경에서 이동에이전트는 가용성과 신용도를 계산하기 위한 샘플 작업 객체를 수행하고, 그리고 고정된 에이전트는 작업 객체(연산)를 수행하는 역할을 수행한다.

기본적으로, 본 논문에서는 결과 검증을 위한 정확성을 보장하기 위하여 신뢰성이 낮은 자원제공자에 대해서 복제 정책을 시행한다. 본 논문에서 사용하는 복제 정책은 다음과 같다. 스케줄링 기능을 담당하는 조정자는 자원제공자 i 의 신용도 C_i 값과 신용도의 임계값 θ 를 비교한다. 만약 $C_i > \theta$ 라면 자원제공자 i 에게 작업 객체를 할당한다. 그렇지 않으면, 조정자는 다음으로 큰 C_i 값을 갖는 자원제공자를 선택하고 다시 θ 와 비교하여 복제 여부를 결정한다. 여기서, θ 는 각 응용마다 서로 다른 값을 가질 수 있는데 허용 에러율 ε 에 의해 결정된다. 본 논문에서는 $\theta = 1 - \varepsilon$ 로 계산된다.

그림 2 는 조정자에 위치한 작업 풀과 자원 풀을 보여준다. 이 그림에서 작업 풀은 작업들의 집합을 관리하는데, 하나의 작업에 대해서 w_{id} 는 작업 객체, v_{id} 는 자원제공자, 그리고 res 는 자원제공자에 의해 수행된 연산 결과 값을 나타낸다. 또한, 자원 풀은 자원들의 집합을 관리하는데, 각 자원들은 신용도 C_i 와 샘플링 수 n, 그리고 가용성 Λ_i 를 갖는다. 아울러, 본 논문에는 자원제공자들이 수행한 연산에 대해 잘못된 결과를 되돌려줄 에러율 f 를 가정한다. 그림 2 의 예에서는 f 가 0.2 이고 허용 에러율 ε 가 0.001 이므로 θ 는 0.990 을 가정하였다. 그림에서 조정자의 신용도 값(C_3) 이 0.99349 인 자원제공자 v_3 에게 작업 객체 w_3 를 할당한다면, 조정자는 신용도 값이 임계값보다 크기 때문에 복제 없이 작업 객체를 v_3 에게 할당한다.



(그림 3) 가용성(Λ)과 신용도(C)에 따른 자원제공자 분류

3.2 자원제공자 분류와 정의

데스크탑 그리드 컴퓨팅 환경에서 성능은 자원제공자들의 연산 참여와 탈퇴 같은 동적인 특징에 의해 영향을 받는다[8]. 따라서, 자원제공자의 가용성은 연산 수행에 대한 신뢰성 뿐만아니라 시스템의 성능 향상에 있어서 중요한 고려사항이다. 더욱이, 본 논문에서는 자원제공자에 의해 수행된 결과의 정확성을 보장하기 위해 자원제공자의 신용도를 계산한다. 본 논문에서는 그림 3 에서 보는 것처럼 연산 결과에 대한 정확성과 함께 높은 성능을 보장하기 위해 가용성과 신용도라는 두 가지 요소에 따라 자원제공자들을 네 가지 유형으로 분류한다. 먼저, 다음과 같이 자원제공자의 가용성과 신용도를 정의한다.

정의 1. *자원제공자 신용도* (C). 신용도는 자원제공자에 의해 수행된 연산 결과에 대한 정확성을 판단하는 요소이다.

정의 2. *자원제공자 가용성* (Λ). 가용성은 자원제공자가 연산 결함에서 올바르게 동작할 확률이다.

$$C_i = \begin{cases} 1-\dfrac{f}{n}, & \text{if } n>0 \\ 1-f, & \text{if } n=0 \end{cases} \qquad (1)$$

$$\Lambda_i = \frac{MTTCF}{MTTCF + MTTCR} \qquad (2)$$

수식 (1)에서 C_i 는 자원제공자 v_i 의 신용도이고 n 은 ASMA 에 의해 결과를 올바르게 되돌려준 샘플링 개수이다. f 는 자원 풀에서 자원제공자를 무작위로 선택했을 때 악의적인 자원제공자일 확률이다. 만약 n 이 0 이면 $1-f$ 로 신용도를 계산한다. 수식 (2)에서 ***MTTCF*** 는 연산 탈퇴나 네트워크 결함 등의 평균 결함 시간이고 ***MTTCR*** 은 평균 연산 수행 시간을 말한다. 즉, 가용성은 결함 시간과 실제 연산 수행 시간을 더한 전체 시간에 대한 실제 수행 시간의 비율이다.

본 논문에서 자원제공자는 그림 3 과 같이 가용성과 신용도에 따라서 네 가지로 분류된다. 첫째, AV(Affirmative Volunteer)는 높은 신용도와 함께 높은 가용성 값을 갖는 자원제공자이다. 둘째, PV(Passive Volunteer)는 높은 신용도와 낮은 가용성 값을 가지는 자원제공자이다. 셋째, A'V(Active Volunteer)는 비교적 낮은 신용도와 높은 가용성 값을 가지는 자원제공자이다. 넷째, NV(Negative Volunteer)는 신뢰적이지도 않고 가용성도 낮은 자원제공자를 말한다.

3.3 자율적 샘플링 기법

본 논문의 DGS 는 "ODDUGI" 이동에이전트 시스템[7]을 이용하여 자원제공자의 신용도와 가용성을 검사하는 자율적 샘플링 기법을 제안한다. 이전 연구들[3, 10]은 중앙 관리 서버가 각 자원제공자들에게 샘플을 보낸 후 결과를 되돌려 받는 방법으로서 서버의 부하 증대와 함께 높은 통신 비용을 초래하였다. 한편, 본 논문의 자율적 샘플링 기법은 조정자에서 생성된 이동에이전트가 연산 그룹의 자원제공자들에게 이주하면서 자원제공자들을 평가한다. 따라서 이동에이전트를 통해 각 자원제공자들에게 이주하여 샘플링함으로써 기존 연구에 비해 통신 비용 측면에서 효과적이다. 결과적으로, 이동에이전트기반 자율적 샘플링 기법은 낮은 통신 비용으로 정확성을 보장할 수 있다.

3.4 연산 복제 기법

DGS 에서 작업 완료율은 개별 자원제공자의 가용성에 의존하는데 높은 작업 완료율을 성취하기 위해 연산 복제 기법이 일반적으로 사용되어 왔다[8]. 복제는 신뢰적인 연산뿐만 아니라 자원제공자의 결함에 대비할 수 있는 방법 중에 하나이다. 본 논문에서는 자원제공자의 가용성과 신용도 값에 따라서 복제 개수를 결정하는데 자원제공자의 이러한 두 가지 값에 따라 조정자가 차별화된 복제 개수를 결정한다. 조정자는 자원제공자의 연산 수행 이력(computation history)을 기반으로 평균 가용성 $\overline{\Lambda_i}$ 를 다음과 같이 계산한다.

$$\overline{\Lambda_i} = \frac{\sum \Lambda_k'}{K} \qquad (3)$$

여기서, K 는 자원제공자 i 에서 완료된 작업의 전체 개수를 나타내며, Λ_k' 는 전체 K 개의 완료작업 중 k 번째 작업의 완료 시간을 나타낸다.

수식 (3)을 이용하여 작업에 대한 복제 개수 n 은 다음과 같이 계산한다.

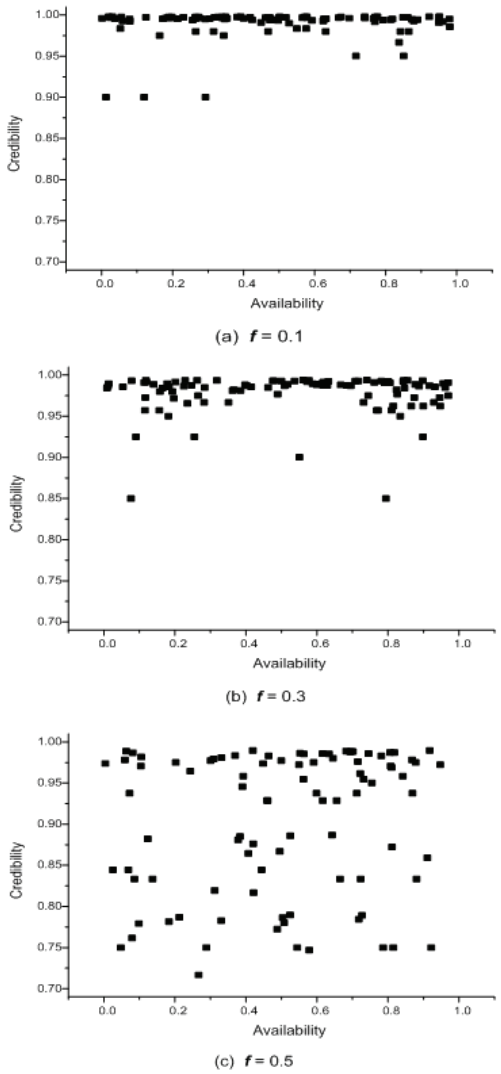
$$n = \frac{\overline{\Lambda_i}}{\Psi_i \times C_i} \qquad (4)$$

여기서, Ψ_i 는 자원제공자 i 가 연산 결함 없이 작업을 완료했을 때의 수행 시간을 나타내며, C_i 는 자원제공자 i 의 신용도로서 수식 (1)로부터 얻어진다.

4. 구현 및 성능 평가

본 논문에서 제안하는 ARVM 의 성능을 평가하기 위해서 Korea@Home 데스크탑 그리드 시스템[4] 위에 "ODDUGI" 이동에이전트 시스템 [7]을 구현하였다. Korea@Home 은 인터넷에 연결된 데스크탑 컴퓨터들의 유휴 컴퓨팅 시간을 이용하여 그리드 응용을 수행하는 데스크탑 그리드 시스템이다. 성능 평가를 위해 사용된 응용은 생명공학 분야에서 가상 탐색 기술을 기반으로 신약 후보 물질을 찾아내는 응용이다. 이 응용에서 하나의 작업은 결함이 없는 자원제공자에서 평균적으로 약 16 분이 소요되었다.

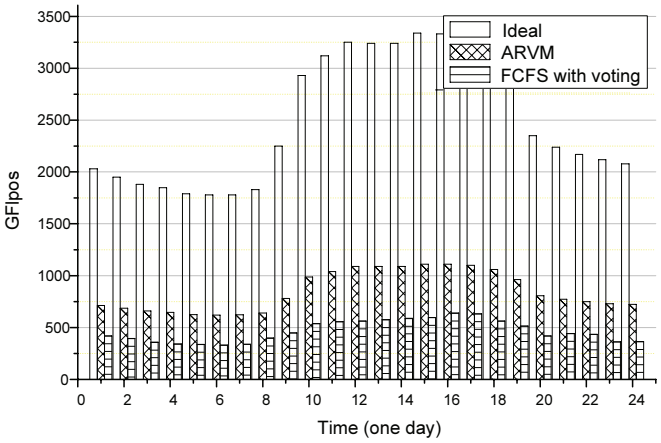
그림 4 는 일주일 동안의 실측 데이터를 기반으로 에러율 f 가 0.1, 0.3, 0.5 일 때 신용도와 가용성에 따른 자원제공자들의 분포를 보여준다. 그림 4 (a)는 에러율이 0.1 일 때의 분포로서 대체로 자원제공자들이 신뢰적으로 연산을 수행하는 반면 가용한 자원은 골고루 분포함을 알 수 있다. 그림 4 (b)의 경우에는 에러율이 0.3 일 때 자원제공자들의 분포를 나타내는데 (a)의 경우보다 신뢰적인 연산 수행은 감소하지만 대체로 비슷한 분포를 갖는다. 그림 4 (c)에서는 에러율이 0.5 일 때 분포로서 신뢰적인 연산을 수행하는 자원제공자들의 정도가 0.7~1.0 까지 골고루 분포함을 알 수 있다. 이 그래프에서 보는 것처럼 결과적으로 에러율이 높을수록 자원제공자들은 신용도와 가용성의 값의 변화에 의해 폭넓게 분포함을 알 수 있다.



(그림 4) 신용도와 가용성에 의한 자원제공자들의 분포

그림 4 에서의 분포를 가지고 본 성능평가에서는 자원제공자들을 각각 다른 비율의 세 가지 경우에서 네 가지 유형(AV, PV, A'V, NV)으로 나누어 그림 5 와 같이 하루 동안의 성능을 얻어 내었다. 각각의 에러율 f 마다 연산 완료 시간을 측정한 결과 신용도의 경우 NV 는 0.0~0.98, PV 는 0.98~1.0, A'V 는 0.0~0.98, 그리고 AV 는 0.98~1.0 으로 설정하고 가용성의 경우에는 NV 가 0.0~0.9, PV 가 0.9~1.0, A'V 는 0.0~0.9, 그리고 AV 가 0.9~1.0 인 경우에 가장 빠른 연산 완료 시간을 보였다. 따라서, 본 논문에서는 이 경우를 가지고 그림 5 와 같이 성능 평가를 하였다.

그림 5 에서 성능 평가는 각 에이전트가 조정자에게 보내온 작업 결과로서 측정하였다. 조정자는 한 시간 단위로 Linpack benchmark [9]에 의해 DGS 의 성능을 평가한다. 그림 5 에서 이상적 성능(Ideal)은 연산 결함이 없다는 가정하에 측정된 성능이다. 그리고 본 논문에서 ARVM 은 위에서 가장 좋은 성능을 보인 자원제공자의 비율을 적용하여 성능을 측정하였다. 투표기반 FCFS 기법에서는 자원제공자가 작업풀에 작업을 요청하면 일정한 복제 개수 (여기에서는 3 개의 복제를 가정)와 함께 작업을 수행하여 성능을 측정하였다. 측정 결과는 기존의 FCFS 를 사용한 기법보다 본 논문의 기법이 모든 시간대에서 약 2 배 정도의 성능 향상을 보였다. 이는 본 논문에서 제안하는 ARVM 이 기존 기법에 비해 결과 검증에서 발생 될 수 있는 연산 지연을 감소시키는 것을 시사한다.



(그림 5) ARVM 과 투표 기반 FCFS 기법과의 성능 비교

5. 결론

본 논문에서는 DGS 에서 연산 결과에 대한 검증을 위하여 ARVM 메커니즘을 제안하였다. ARVM 은 결과 검증으로 인한 큰 성능 저하를 막고 결과에 대한 정확성을 보장하는 기법이다. 성능 평가에서 본 것처럼 기존에 결과 검증을 위해 사용하였던 투표기반의 FCFS 기법들에 비해 좋은 성능을 보임을 알 수 있었다.

참고문헌

[1] M. O. Neary and P. Cappello, "Advanced Eager Scheduling for Java Based Adaptively Parallel Computing," *Concurrency and Computation: Practice and Experience*, Vol. 17, Iss. 7-8, pp. 797-819, Feb. 2005.

[2] D. Molnar, The SETI@home Problem, http://turing.acm.org/crossroads/columns/onpatrol/september2000.html

[3] L. Sarmenta, "Sabotage-Tolerance Mechanism for Volunteer Computing Systems," *Future Generation Computer Systems*, Vol. 18, No. 4, pp. 561-572, Mar. 2002.

[4] KOREA@Home homepage, "http://www.koreaathome.org".

[5] S. Zhao, V. Lo, and C. G. Dickey, "Result Verification and Trust-Based Scheduling in Peer-to-Peer Grids," *Proc. of the 5th IEEE Int. Conf. on Peer-to-Peer Computing*, pp 31-38, Sept. 2005.

[6] SETI@Home homepage, "http://setiathome.ssl.berkeley.edu".

[7] S. Choi, M. Baik, H. Kim, E. Byun, and C. Hwang, "Reliable Asynchronous Message Delivery for Mobile Agent," *IEEE Internet Computing*, Vol. 10, Iss. 6, pp. 16-25, Dec. 2006.

[8] D. Kondo, M. Taufer, C. L. Brooks, H. Casanova, and A. Chien, "Characterizing and Evaluating Desktop Grids: An Empirical Study," *Proc. of the 18th Int. Parallel and Distributed Processing Symp.*, pp 26-35, April 2004.

[9] J. Dongarra, "Performance of various computers using standard linear equations software," *ACM SIGARCH Computer Architecture News*, Vol. 20 , pp. 22-44, June 1992.