데스크탑 그리드 환경에서 데스크탑 가용성 기반 마코브 작업 스케줄링 기법

변은정[O], 박학수[‖], 박찬열[‖], 최장원[‖], 정순영[‡], 황종선[†]
고려대학교 컴퓨터학과[†], 한국과학기술정보연구원[‖], 고려대학교 컴퓨터교육학과[‡]
{vision[O], hwang[†]}@disys.korea.ac.kr, {hspark, chan, jwchoi}@kisti.re.kr[‖], jsy@comedu.korea.ac.kr[‡]

Markov Job Scheduling Scheme based on Desktop Availability in Desktop Grid Computing Environment

EunJoung Byun[O], ChongSun Hwang[†]
Dept. of Computer Science & Engineering, Korea University
HarkSoo Park[‖], ChanYeol Park[‖], JangWon Choi[‖]
Supercomputing Center, KISTI
SoonYoung Jung[‡]
Dept. of Computer Science & Education, Korea University

요 약

데스크탑 그리드(Desktop Grid) 환경에서는 자원 제공자인 데스크탑의 자율적 연산 참여와 탈퇴를 허용하기 때문에 기존 데스크탑 그리드 시스템들은 연산 도중 잦은 중단으로 인해 연산의 완료 시간이 지연되고, 연산 수행의 신뢰성이 저하되며 연산의 완료를 보장하지 못하고 있다. 기존의 데스크탑 그리드 시스템들은 이러한 데스크탑의 동적 특성을 반영하지 못하고 있을 뿐만 아니라 데스크탑의 연산 수행 양식을 고려하지 않아 시스템의 안정성과 성능이 저하되었다. 신뢰성 높은 연산 수행을 지원하고, 데스크탑의 비예측적 연산 수행 속성을 극복하기 위해서는 데스크탑의 동적인 특성인 휘발성(volatility)을 고려한 스케줄링이 필요하다. 본 논문에서는 휴리스틱 접근을 통해 동적으로 가변하는 데스크탑의 상태를 보다 정확하게 모델링 하는 가용성 기반 마코브 작업 스케줄링 기법을 제안한다. 제안 기법은 데스크탑의 가용성을 기반으로 과거 연산 수행에 대한 패턴을 확률 모델링하여 미래 연산 수행 유형을 예측함으로써 연산 수행 도중의 불안정한 자원 제공 현상을 완화시키며 안정적인 스케줄링을 지원하여 시스템의 신뢰성과 성능을 향상시킨다.

1. 서 론

그리드 컴퓨팅[1]은 가상 조직을 이용하여 상이한 기관에 분산된 다양한 고성능의 자원에 대한 지속적인 접근을 제공한다. 데스크탑 그리드 컴퓨팅[2]은 그리드 컴퓨팅의 일종으로 인터넷 기반 병렬 처리를 통해 높은 처리율과 성능을 지원하는 시스템을 구축하는 컴퓨팅 방식이다. 데스크탑 그리드는 데스크탑의 성능 향상과 인터넷의 발전을 기반으로 글로벌 컴퓨팅, 자원제공자 컴퓨팅, P2P 그리드 컴퓨팅, 공공 컴퓨팅 등으로 불리며 발전해 왔으며, 대규모 응용의 연산을 인터넷에 연결된 대규모의 유휴 컴퓨팅 자원에 작업을 병렬 처리한다. 데스크탑 그리드는 저비용으로 슈퍼컴퓨터 이상의 높은 처리 능력을 제공하고, 보다 용이한 데스크탑의 참여를 지원한다. 복잡한 설치 과정과 설정, 기술적인 배경 지식 등을 요구하는 그리드와 달리 데스크탑 그리드 컴퓨팅은 일반 인터넷 사용자들도 특별한 기술적 지식이나 복잡한 설치 과정 없이 보다 편리하게 시스템에 참여, 이탈할 수 있는 높은 자율성과 접근성을 제공한다.

데스크탑 그리드 컴퓨팅에서 인터넷에 연결된 데스크탑은 휘발성과 같은 데스크탑의 자율적, 동적 연산 참여 특성은 데스크탑의 연산 유형을 예측하기 어렵게 만들며 잦은 연산 중단을 야기하고, 연산 수행 시간을 지연시킨다. 휘발성은 데스크탑의 고장(failure)이나, 자율적인 연산 참여와 이탈 또는 사용자의 간섭에 의해 일시적, 장기적으로 연산 수행이 중단되거나, 네트워크가 일시적으로 단절되는 현상으로 데스크탑의 사용이 불가한 상태를 의미한다. 데스크탑의 휘발성은 지속적으로 안정적인 연산 수행을 보장할 수 없도록 하여 연산의 신뢰성을 저하시키고, 연산 시간을 지연시켜 결국 전체 연산 수행 시간 역시 증가시킨다.

본 논문에서는 자원의 휘발성으로 인해 발생되는 문제를 해결하기 위해 가용성(availability)을 기반으로 마코브 체인을 이용하여 데스크탑의 상태를 확률 모델링하는 가용성 기반 마코브 작업 스케줄링 기법을 제안한다. 제안 기법은 데스크탑의 확률 모델링을 통해 가용성을 보다 정확히 측정하고 가용성의 지속성을 예측함으로써 작업에 대해 적합한 가용성과 지속성을 가지는 데스크탑을 선정하고 안정적 스케줄링을 지원한다.

2. 관련 연구

기존의 데스크탑 그리드 컴퓨팅 시스템들의 스케줄링 기법은 크게 FCFS(First Come First Serve) 스케줄링 기법, 선행처리자 우선할당 스케줄링(Eager Scheduling) 기법, 데스크탑의 성능과 연산 참여 시간을 고려한 스케줄링 기법, 확률적 스케줄링(Stochastic Scheduling) 기법 등 네 가지로 분류된다.

첫 째, FCFS 기법은 먼저 온 데스크탑에 먼저 작업을 할당하는 방식으로 XtremWeb, Korea@Home[5]등에서 이용되었다. XtremWeb에서는 단순한 FCFS 기법에 기반한 작업 가로채기 기법으로 스케줄링하며, 데스크탑에 결함이 발생하여 작업이 중단되면 중단된 작업을 재스케줄링한다. 그러나 FCFS 기법은 데스크탑 그리드 컴퓨팅의 다양하고 동적인 특성을 고려한 연산 할당 기법 및 연산 수행 모델이 없어 비효율적이다.

둘 째, 선행처리자 우선할당 스케줄링 기법은 FCFS의 변형된

형태로 작업을 먼저 종료한 데스크탑에게 미완료 작업을 재할당하는 방식이다. 선행처리자 우선할당 기법은 적응적 병렬성을 위한 단순한 기법으로 자주 이용되어 왔으며 Bayanihan[3], Charlotte, Javelin 등 많은 시스템에 적용되었다. 선행처리자 우선할당 스케줄링 기법은 데스크탑의 불안정한 자원 제공 및 일시적인 연산 간섭이나 네트워크 단절 등과 같은 데스크탑 그리드 컴퓨팅의 동적인 특성을 고려하지 않고, 단순히 작업을 먼저 종료한 데스크탑에게 추가적으로 작업을 할당하고 있어 데스크탑이 연산 수행에 실패할 경우, 연산 지연 시간이 증가되고, 연산의 완료를 보장하지 못하며, 처리할 수 있는 응용이 제한적인 단점이 있다. 선행처리자 우선할당 기법에서는 데스크탑의 연산 수행 실패로 인한 반복적 재할당으로 인해 안정적 연산 수행을 보장하지 못하고, 전체 연산 수행 시간이 지연된다. Javelin++[4], Javelin 3 등은 선행처리자 우선할당 기법을 개선하여 진보된 형태로 적용하였으나, 부가적인 오버헤드가 크고, 작업 관리의 복잡도가 증가되는 단점이 발생하고, 연산 지연 및 전체 연산 수행 시간 증가 문제도 여전히 남아 있다.

셋 째, 데스크탑의 성능과 연산 참여 시간을 고려한 기법은 WBC(Web-Based Computing)에서 데스크탑의 성능과 데스크탑의 연산 참여 시간을 데스크탑의 작업량과 연관시켰다. 그러나 데스크탑과 작업 간의 연관은 단순히 데스크탑의 속도만을 반영하여 연산의 의존성, 지역성, 가용성 등을 반영하지 못하는 한계점을 가지고 있으며 연산 중단이나 결함이 발생할 경우 재스케줄링 및 재연관에 대한 기법은 제시하고 있지 않다.

넷 째, 확률적 스케줄링 기법[6]은 확률적 변수를 이용하여 선험적인 데스크탑의 성능 변이를 표현하여 스케줄링에 적용하는 방식이다. 그러나 기존 기법의 확률적 변수는 단순한 값과 순간적 예측만을 제공하여 확률적 예측의 정확성을 항상 보장하기 어렵고 동적 데스크탑의 특성 역시 반영하지 못하고 있다.

이와 같이 대부분의 기존 연구들은 불안정한 데스크탑의 연산 수행 유형 및 일시적인 연산 간섭이나 네트워크 단절 등으로 인한 휘발성과 같은 데스크탑 그리드 컴퓨팅의 동적인 특성을 반영하거나 이를 해결하기 위해 가용성 등을 고려한 스케줄링 기법 등이 거의 없다. 따라서 지속적인 연산 및 연산의 완료성을 보장하지 못하고, 지연 시간의 증가에 따른 전체 연산 수행 시간의 증가로 시스템의 성능이 저하시키고 있다.

3. 가용성 기반 마코브 모델링

기존의 데스크탑 그리드 컴퓨팅 시스템들은 스케줄링 시에 데스크탑들의 가용성(availability)을 고려하지 않거나[3][4], 단순한 통계적인 방법[2]으로 가용성을 고려하여 데스크탑의 연산 실패 발생 시에 연산의 완료를 보장하지 못하고, 연산 수행 패턴을 예측하지 못하였다. 데스크탑의 자율적, 동적 연산 수행 패턴에 대한 분석이 선행된 후, 작업이 요구하는 가용성에 부합하는 데스크탑을 선정하여 작업을 할당하는 것이 필요하다.

본 논문에서는 데스크탑의 연산 수행 패턴, 즉 가용성을 예측하기 위해 통계 모델이나 단순한 정점 확률 값을 사용하지 않고, 시간 상태에 따른 데스크탑의 동적 상태 변화를 마코브 체인을 이용하여 확률 모델링함으로써 보다 정확한 데스크탑의 가용성을 제공하고, 예측의 신뢰성을 향상시킨다. 이로써 연산 실패율을 저하시킴은 물론 연산 실패로 인한 지연 시간을 감소시키고, 연산의 완료성을 향상시킨다. 데스크탑의 가용성을 보다 정확하게 분석하여 작업을 할당하므로 참여와 이탈이 자유로운 데스크탑 그리드 컴퓨팅의 동적 상태에 능동적으로 대처하는 신뢰성 높은 데스크탑 시스템을 구축하도록 지원한다.

3.1 마코브 모델링

데스크탑의 시간 변화에 따른 상태 변화를 표현하기 위해 각 데스크탑의 상태를 30분마다 검사하여 기록하며, 하루를 주기로 데스크탑의 상태 변화를 모델링한다. 따라서 하루는 총 48개의 시간 단위로 모델링되고, 각 시간 단위의 상태와 시간 단위 간의 상태 전이 확률을 각각 계산한다.

데스크탑의 상태는 유휴 상태 *I*(idle), 사용 상태 *U*(use), 정지 상태 *S*(stop)로 구분된다. *I* 상태는 20분 이상 연산 수행 도중에 사용자의 간섭 없이 연산을 수행할 수 있는 상태이다. *U* 상태는 사용자의 일시적 간섭이나 자원 점유로 인해 연산을 수행할 수 없는 상태이다. *S*는 자원의 물리적 결함(crash)이나 시스템 종료, 네트워크 단절로 연산을 수행할 수 없는 상태이다.

데스크탑의 가용 상태는 매 30분마다 검사하여 자원의 시간대별 상태를 마코브 체인에 갱신하고 확률을 재계산한다.

기본적인 마코브 체인은 **그림 1**과 같이 모델링된다. **그림 1**에 나타난 바와 같이 I_i는 i번째 시간 단위의 I상태를 의미한다. 전이 확률 P_{ij}는 i번째 시간 단위의 I상태에서 j번째 시간 단위의 I상태로 전이하는 확률을 의미한다. 전이 확률 행렬 P는 j시간 단위에서 k시간 단위로의 상태 전이 확률들을 표현한다.

$$P = \begin{array}{c} \\ I_j \\ U_j \\ S_j \end{array} \begin{array}{c} \begin{array}{ccc} I_k & U_k & S_k \end{array} \\ \begin{pmatrix} P_{I,I} & P_{I,U} & P_{I,S} \\ P_{U,I} & P_{U,U} & P_{U,S} \\ P_{S,I} & P_{S,U} & P_{S,S} \end{pmatrix} \end{array}$$

상태 전이 확률 행렬에서 각 행의 합은 1이 된다.

이와 같이 데스크탑의 각 시간대에 나타날 수 있는 상태들의 확률로 각 데스크탑의 각 시간대의 가용 상태를 예측할 수 있다. 예측의 정확도를 향상시키기 위해 마코브 모델 내의 각 상태의 신뢰도 값을 계산한다. 마코브 체인 모델에서 데스크탑의 가용성은 실제 연산을 수행할 수 있는 I 상태를 중심으로 I를 지나가는 모든 경로들의 확률 값을 통해 표현된다. I 상태의 신뢰도는 가용 상태를 표현한다. 또한 비터비(viterbi) 알고리즘[7]을 적용하여 최확치(most probable value) 및 최적 상태 열을 추출하여 시간 변화에 따른 상태 변화의 예측을 지원한다.

I 상태를 지나가는 모든 경로의 확률은 전방향 상태 신뢰도 FSC (Forward State Credit)와 후방향 상태 신뢰도 BSC (Backward State Credit)의 곱으로 구할 수 있다. 전방향 상태 신뢰도는 이전 시간대에서 현재 I 상태로 전이하는 모든 상태의 전이 확률과 이전 상태의 신뢰도 값의 곱들의 합으로써 다음 식과 같이 계산된다. 후방향 상태 신뢰도는 현재 I 상태에서 다음 시간대의 상태로 전이하는 확률과 다음 상태의 신뢰도의 곱들의 합으로써 다음 식과 같이 계산된다. 현재 I 상태를 중심으로 구해진 전방향 상태 신뢰도와 후방향 상태 신뢰도의 곱이 I의 상태 신뢰도 SC(State Credit)가 된다.

- 전방향 상태 신뢰도:

$$FSC\_I_i = SC\_I_{i-1} * PI_{i-1}I_i + SC\_U_{i-1} * PU_{i-1}I_i + SC\_S_{i-1} * PS_{i-1}I_i$$

- 후방향 상태 신뢰도:

$$BSC\_I_i = SC\_I_{i+1} * PI_iI_{i+1} + SC\_U_{i+1} * PI_iU_{i+1} + SC\_S_{i+1} * PI_iS_{i+1}$$

- I_i를 지나가는 모든 경로의 확률값, 즉 I_i의 상태 신뢰도:

$$SC\_I_i = FSC\_I_i * BSC\_I_i$$

결론적으로 SC_I_i 값은 I_i를 대표하는 I_i의 신뢰도 값이 된다.

마코브 모델 내에서 처음 상태의 전방향 상태 신뢰도와 마지막 상태의 후방향 상태 신뢰도는 각각 일어나는 빈도의 비율로서 구한다. 예를 들어, **그림 1**에서 I_1, U_1, S_1 의 빈도를 각각 f_I_1, f_U_1, f_S_1 이라고 하면, I_1 의 전방향 신뢰도는 f_I_1, f_U_1, f_S_1의 총합에 대한 f_I_1 의 비율 값이다.

$$FSC\_I_1 = \frac{f\_I_1}{f\_I_1 + f\_U_1 + f\_S_1}, \quad BSC\_I_{48} = \frac{f\_I_{48}}{f\_I_{48} + f\_U_{48} + f\_S_{48}}$$

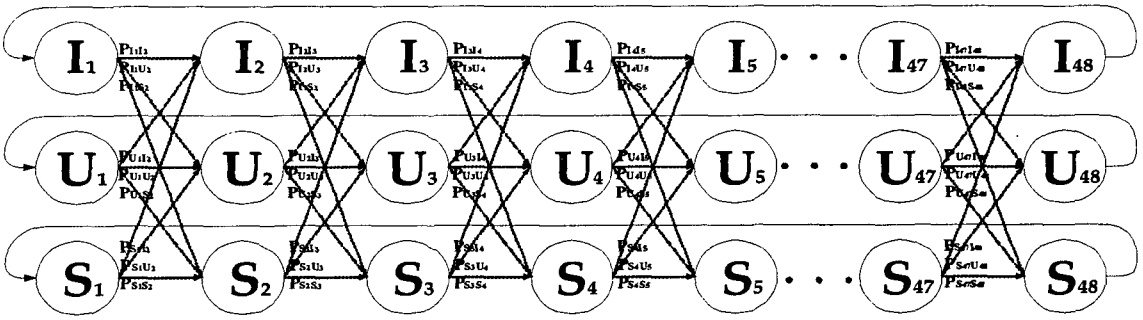

그림 1. 가용성 기반 마코브 모델

```
// Estimation Step
For(i=1 ; 48 ; i++)
{
    Calculate_FSC( )
    {   If(I1) then
            FSC_I1=Frequency(f_I1);
        Else then
            FSC_Ii=SumOfPreviousSC(SC_Ii-1);
    }
    Calculate_BSC( )
    {   If(I1) then
            BSC_I48=Frequency(f_I48);
        Else if(I48) then
            BSC_Ii=SumOfNextSC(SC_Ii+1);
    }
    Calculate_SC( )
    {   SC_Ii = FSC_Ii * BSC_Ii;
    }
}
// Eligibility step
For(i=1 ; 48 ; i++)
{   Compare(SC_Ii, Job_SC_Ii);
    MatchIf(SC_Ii, Job_SC_Ii)
        {   EligibleDeskTop = Select(SC_Ii, DesktopList);
            AllocateWorkLoad(EligibleDeskTop);
        }
}
//Execution step
ExecuteWorkLoad(WorkLoad);
If(Completed) then
    Return(Results);

//Evaluation step
EvaluateAccuracy( )
{               TotalPrediction=    AccuratePrediction    +
InaccuratePrediction;
    Accuracy = AccuratePrediction / TotalPrediction;
}
UpdateMarkovModel(Accuracy);
```

그림 2. 마코브 작업 스케줄링 알고리즘

3.2 마코브 작업 스케줄링 기법

데스크탑 그리드 환경에서 안정적인 시스템을 환경을 제공하고, 높은 성능을 보장하기 위해서 데스크탑의 동적, 비예측적 특성을 고려한 스케줄링은 매우 중요한 고려사항이다. 마코브 작업 스케줄링 기법은 제안된 가용성 기반 마코브 모델을 기반으로 스케줄링하여 시간의 변화에 따라 가변하는 데스크탑의 연산 수행에 대한 불확실성, 비예측성, 휘발성 문제를 완화시킨다. 마코브 작업 스케줄링 기법은 스케줄링 시에 실시간으로 예측에 대한 정확성을 반영하여 마코브 모델을 갱신, 적용한다.

스케줄링은 추정 단계(Estimation), 적합 단계 (Eligibility), 실행 단계(Execution), 평가 단계(Evaluation) 순으로 진행된다.

추정 단계에서는 마코브 모델 내의 각 시간 단위에 대한 신뢰도, 즉, 전방향 상태 신뢰도, 후방향 상태 신뢰도, 상태 신뢰도를 상태 전이 확률을 이용하여 추정, 계산한다.

적합 단계에서는 작업량과 데스크탑의 가용성 및 지속성에 기반을 둔 I의 상태 신뢰도 SC_I값을 비교하여 시간대의 상태와 작업의 양에 부합하는 데스크탑을 선정하고 작업을 할당한다.

실행 단계에서는 각 데스크탑이 할당받은 작업을 수행하고, 작업 수행이 종료되면 결과 값을 반환한다.

마지막 평가 단계에서는 데스크탑의 연산 수행의 유형을 분석하여 예측의 정확성을 평가하고 마코브 모델을 갱신한다. 예측의 정확성 PA(Prediction Accuracy)는 총 예측TA(Total Accuracy) 횟수에 대한 정확한 예측 AP(Accurate Prediction) 횟수의 비율로서 구한다. 총 예측 횟수는 정확한 예측의 횟수와 비정확한 예측(Inaccurate Prediction)의 횟수의 합이다.

$$PA = AP / (IP + AP)$$

마코브 작업 스케줄링 알고리즘은 그림 2에 기술되어 있다.

4. 결론 및 향후 연구 과제

본 논문은 데스크탑 그리드 컴퓨팅 환경에서 스케줄링 시에 데스크탑의 동적, 자율적 연산 수행 특성을 고려하기 위해 데스크탑의 가용성, 즉 연산 수행 패턴에 대한 마코브 체인을 이용한 확률 모델링을 제안하였다. 이를 통해 예측의 신뢰성을 향상시키고, 작업의 크기나 종류에 따라 적합한 데스크탑을 선정하여 데스크탑의 연산 유형을 보다 정확하게 예측하였다. 가용성 기반 마코브 작업 스케줄링 기법은 데스크탑의 불안정한 자원제공과 연산 중단 현상을 완화하여, 전체 연산 수행 시간을 단축시키고, 결과적으로 연산 수행에 대한 완료성과 신뢰성을 향상시킨다.

향후, 공헌도 기반 스케줄링 기법을 Korea@Home에 구현, 적용하여 실제 환경에서 성능의 우수성을 입증할 예정이다.

참고 문헌

[1] F. Berman, G. C, Fox and J. G. Hey, "Grid Computing : Making the Global Infrastructure a Reality", Wiley, 2003
[2] Derrick Kondo, Michela Taufer, John Karanicolas, Charles L. Brooks, Henri Casanova and Andrew Chien, "Characterizing and Evaluating desktop Grids - An Empirical Study", IPDPS04, 2004.
[3] Luis F. G. Sarmenta, Volunteer Computing, Ph.D. thesis. Dept. of Electrical Engineering and Computer Science, MIT, 2001.
A.Baratloo, M.Karaul, Z.Kedem, and P.Wyckoff. "Charlotte : Metacomputing on the web", PDCS, 1996.
[4] Michael O. Neary, Sean P. Brydon, Paul Kmiec, Sami Rollins, Peter Cappello, "Javelin++:scalability issues in global computing", Concurrency : Practice and Experience Volume 12, Issue 8, pp. 727-753, 2000.
[5] "Korea@Home", http://www.KOREAatHOME.org
[6] J. Schopt and F. Berman, "Stochastic Scheduling", Proceedings of SC99, November, 1999.
[7] G. D. Forney Jr., "The Viterbi Algorithm", IEEE Proceedings of TT, Mar. 1973.