

모바일그리드 환경에서 성능예측 기반의

작업 스케줄링 기법¹⁾

송성진⁰⁺, 진성호⁺, 정대용⁺, 정광식⁺⁺, 유현창⁺²⁾

⁺ 고려대학교 대학원 컴퓨터교육학과,

{white⁰, wingtop, karat, yuhc}@comedu.korea.ac.kr

⁺⁺ 한국방송통신대학교 컴퓨터과학과

kchung0825@knou.ac.kr

Performance Prediction based Job Scheduling Method in Mobile Grid

Sungjin Song⁺, SungHo Chin⁺, DaeYong Jung⁺, KwangSik Chung⁺⁺, HeonChang Yu⁺

⁺ Dept. of Computer Science Education, Korea University

⁺⁺ Dept. of Computer Science, Korea National Open University

요 약

최근 들어, 모바일 장치의 성능이 향상되고 보급률이 증대됨에 따라 모바일 장치를 그리드 자원으로 이용하기 위한 모바일 그리드가 등장하였다. 그러나 모바일 장치가 가지는 무선기기로써의 제약사항 즉, 무선 통신의 불안정성, 이동으로 인한 연결 끊김 등의 문제와 배터리의 제약은 모바일 그리드를 구성하는데 많은 어려움을 야기한다. 따라서 본 논문에서는 이러한 제약사항을 극복할 수 있는 환경적인 요소를 고려하여 학교나 회사와 같이 안정적인 무선통신 환경을 제공하고 배터리 충전이 용이한 네트워크 그룹을 가정하였다. 그리고 제한된 성능을 발휘하는 모바일 장치에서 독립적인 소규모 작업의 효율적인 수행을 위해 성능예측 기반 작업 스케줄링 기법을 제시하였다. 이 기법은 네트워크 그룹 내의 모바일 장치의 이용 패턴이 규칙적으로 나타내는 특성을 이용한다.

제안하는 스케줄링 기법에서는 하나의 네트워크 그룹의 성능을 그 그룹에 속한 모바일 장치들의 성능의 합으로 정의하고 시간에 따라 변화하는 모바일 장치들의 성능을 예측하기 위해 기존에 수집된 성능 정보의 통계를 이용한다. 그리고 본 기법은 그리드와 네트워크 그룹, 네트워크 그룹과 모바일 장치 사이의 작업 분배시 예측된 성능 정보를 이용함으로써 네트워크 그룹의 사용률을 높여 전체 작업의 최종 응답시간을 줄일 수 있다.

1. 서 론

그리드 컴퓨팅[1]은 대량의 데이터 처리나 복잡한 문제의 해결을 위한 분산시스템 환경으로 인식되어 우주과학, 생명과학, 경제분석 등 여러 분야에 적용되고 있다[2][3]. 최근에 노트북이나 PDA 또는 HPC (Handheld Personal Computer) 등과 같은 모바일 장치의 보급이 일반화 되고, 빠른 기술의 발전으로 성능 향상이 뒷받침되면서 모바일 장치를 그리드의 자원으로 활용하는 모바일 그리드가 제안되었다[4]. 그리고 모바일 그리드의 도전 과제로서 모바일 장치가 갖는 제한적 성능, 배터리의 제약, 무선통신의 불안정성, 이동으로 인한 연결 끊김 등의 문제가 제기되었다[4][5].

본 논문에서는 위와 같은 문제를 해결하기 위하여 모

바일 그리드의 이용 환경을 AP(Access Point)와 모바일 장치(MD: Mobile Device)로 구성되는 모바일 네트워크(MNet: Mobile Network)와 MNet의 집합으로 만들어지는 회사나 학교와 같은 네트워크 그룹(NG: Network Group)을 가정하였다. 이 환경은 (i) 모바일 장치 이용자 다수가 밀집해서 생활하기 때문에 자원의 확보가 쉽고, (ii) 주요 생활공간이기 때문에 배터리의 충전이 용이하여 배터리의 제약이 적고, (iii) 강의실이나 사무실 같은 주요 활동 장소에 각각 AP가 설치되어 있어서 안정적인 무선 통신 환경을 제공하고, (iv) 모바일 장치가 이동하는 경우에도 이 환경 내에서의 이동하는 경우에는 지속적으로 모바일 그리드의 자원으로 이용 가능하고, (v) 시간에 따른 모바일 장치의 이용 현황 통계[6][7]를 통해 볼 때, 모바일 장치의 이용 패턴이 규칙적으로 나타나므로 네트워크 그룹의 성능을 예측할 수 있게 한다. 따라서 네트워크 그룹은 모바일 그리드의 자원으로 이용하기에 적합한 환경이다. 그리고 NG를 이용한 모바일 그리드 환경을 구축하기 위해서는 다음과 같은 사항들을 고려되어야 한다.

1) 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-311- D00173)

2) 교신저자

- (a) 어떤 작업을 모바일 그리드에 할당할 것인가?
- (b) 주어진 작업을 모바일 장치에 어떻게 분배할 것인가?
- (c) 어떠한 장애가 발생하고, 어떻게 대처할 것인가?

(a)의 경우, MD는 데스크탑에 비해 낮은 성능을 가지며, NG에서 이탈하는 시간을 예측하기 어렵다. 따라서 독립적으로 처리할 수 있으며, 짧은 시간에 해결 가능한 소규모 작업이 모바일 그리드에 적합하다.

(b)의 경우, NG 내부에 안정적인 고성능 프록시 서버(Proxy: Proxy Server)를 두어서 가용한 자원 및 성능 정보를 수집하고, 기존의 유선 그리드에 있는 스케줄러(Scheduler)가 그리드 환경에 주어진 작업 중에서 (a)에 해당하는 작업을 PS에서 측정된 성능을 고려하여 Proxy에 분배한다. Proxy는 주어진 작업을 MD 각각의 성능에 따라 작은 작업들의 집합인 작업 그룹(Job Group)을 분배한다[8].

(c)의 경우, Scheduler와 Proxy는 안정적이기 때문에 장애가 발생하지 않는다고 가정하면, 장애가 발생할 요인은 무선 통신의 장애나 MD의 이동이나 전원 꺼짐으로 인한 MD의 이탈을 생각할 수 있다. 이탈로 인한 문제를 해결하기 위해서 Proxy는 작업을 할당하는 동시에 시간 제한을 정하고, 제한된 시간 내에 MD의 응답이 없는 경우 다른 장치에 같은 작업을 부과하여 처리할 수 있도록 한다. MD가 제한된 시간 내에 복귀하거나 NG 내부의 다른 MNet로 이동했음을 Proxy에 알린 경우에는 해당 작업 그룹의 남은 작업들은 다른 MD에 분산하여 빠르게 처리하게 한다.

모바일 그리드에 주어진 작업을 효율적으로 처리하기 위해서는 Proxy에서 NG에 할당된 작업이 예상 시간 내에 처리되어야 한다. 그러나 일반적인 모바일 환경에서는 MD가 NG에 참여하거나 이탈하는 것의 예측이 불가능하기 때문에 NG의 성능 정보는 시간에 따라 빠르게 변동하고, NG에 할당된 작업의 양과 NG의 성능 사이에 차이가 발생하게 되어 비효율적인 부분이 나타난다. 예를 들어 NG의 성능이 측정된 값보다 좋아지면 작업을 배정받지 못한 장치가 나타나므로 자원의 낭비가 일어나고, 성능이 나빠지면 작업을 처리할 자원이 부족하게 되어 작업 지연이 나타난다. 그러므로 효율적인 작업할당을 위해서 Scheduler는 NG의 성능 변화를 예측하여 이를 반영하여야 한다.

따라서 본문에서는 위의 사항들을 고려한 모바일 그리드 시스템의 구성을 제시하고, 로그 통계를 기반으로 성능을 예측하는 기법과 이를 통한 효율적인 스케줄링 기법에 대하여 나타낸다. 논문의 구성은 다음과 같다. 2장에서는 이에 대한 관련 연구를 서술하고, 3장에서는 본 논문에서 적용된 모바일 그리드 시스템의 환경과 구성요소에 대해 알아보고, 4장에서 NG의 성능을 예측하는 기법과 예측된 성능을 이용한 스케줄링 기법을 알아본다. 그리고 5장에서는 결론과 향후 과제를 언급한다.

2. 관련 연구

모바일 환경에서 자원을 찾아 작업을 처리하는 스케줄

링 기법에 관한 연구는 일반적으로 MD의 가용성을 예측하고, 가용한 시간을 고려하여 작업을 처리하는 방법을 이용한다. 가용한 시간을 구하는 방법으로 일정한 속도를 유지하며 고정된 방향으로 움직이는 MD를 가정하여 통신이 가능한 범위를 벗어나는 시간을 예측하는 기법[9], MD가 임의의 속도와 방향으로 움직일 수 있음을 고려하여 확률적으로 통신이 가능한 범위를 벗어나는 시간을 예측하는 기법[10], 또는 배터리의 잔량을 통해 MD를 이용할 수 있는 시간을 예측하는 기법[11] 등이 있다. 이 기법들은 MD의 이동성과 연결 끊김에 관한 대안을 제시하였지만, 짧은 가용 시간과 장치의 밀집도가 낮아서 그리드의 자원으로 이용하는데 어려움이 있기 때문에 이들이 가정한 환경은 모바일 그리드에 적합하지 않다.

본 논문에서는 MD의 밀집도가 높아서 자원 확보가 용이하고, 각 장치들의 이용이 지속적으로 나타나는 학교나 회사와 같은 네트워크 환경을 모바일 그리드에 적합한 환경으로 보고, 이 환경에서 MD의 이용 패턴이 규칙성을 보이고 있음[6][7]을 이용하여 월별, 주별, 일별, 시간별 로그 데이터 통계를 내고, 이를 이용하여 성능을 예측하는 기법을 사용한다.

3. 시스템 구성

모바일 그리드 환경은 그림 1에서 보는바와 같이 크게 AP와 MD의 집합인 모바일 네트워크(MNet), MNet의 집합인 네트워크 그룹(NG), 기존의 유선 그리드 등 세 부분으로 구분할 수 있다. 예를 들어 무선 통신이 가능한 강의실이나 연구실이 MNet에 해당하고, NG를 구성한다. 기존의 그리드 입장에서 보면 하나의 NG는 그리드 내에 존재하는 하나의 노드에 해당한다.

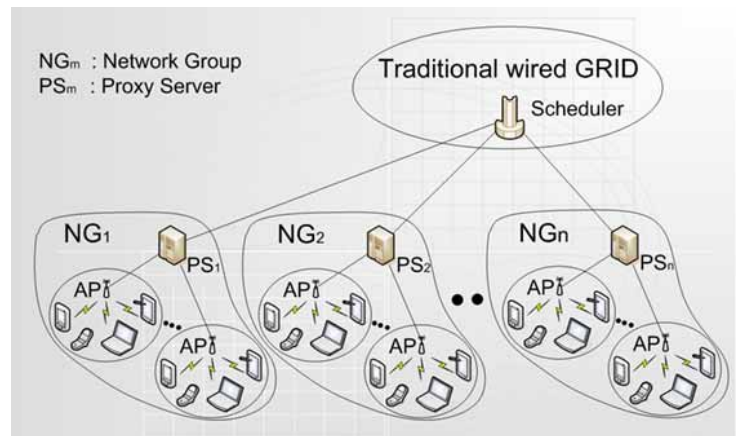


그림 1 모바일 그리드 시스템 구성도

그리고 MD 간의 이질성과 낮은 통신 대역폭, 높은 연결 지연 등의 문제로 인하여 MD와 그리드 환경을 직접적으로 통합하기 어렵기 때문에 이를 중재하기 위해 Proxy를 두었다[5].

모바일 그리드 시스템에서 MD와 Proxy의 구조를 그림 2와 같이 제시하며, 각각의 구성요소는 다음과 같은 기능을 한다.

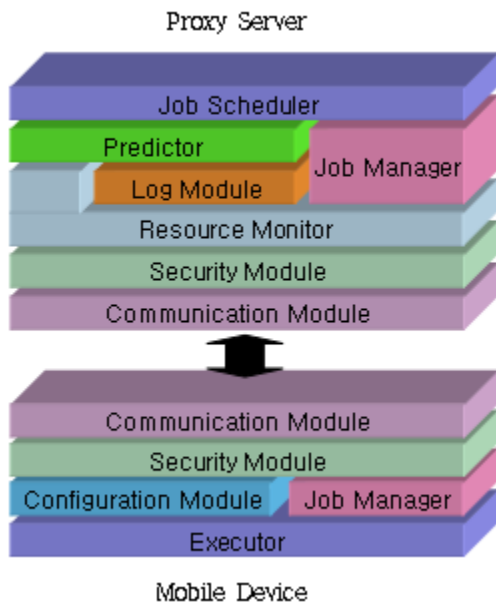


그림 2 프록시 서버와 모바일 장치의 구조

Communication Module은 보안된 통신을 처리하고, Security Module은 MD의 인증과 작업 분배 및 결과 수집에 관한 보안을 처리한다.

Proxy의 Resource Monitor는 MD의 참가/이탈을 감지하고, 각 장치의 자원 정보를 수집한다. 수집된 정보는 Log Module을 통해 기록되고, Predictor에서 현재의 자원 정보와 기존에 기록된 정보의 통계를 이용하여 NG의 성능 변화를 예측하고, 예측된 값을 Scheduler가 이용할 수 있도록 전달한다. Job Manager는 그리드에서 분배 받은 작업을 유지 관리하고, Job Scheduler는 Job Manager의 작업을 MD의 성능에 맞게 분할하여 분배하고, 결과를 수집한다.

MD의 Configuration Module은 MD의 자원 정보를 관리하고 Proxy에 정보를 제공한다. Job Manager는 MD에 배분된 작업을 유지 관리하고, Executor가 Configuration Module의 정보를 통해 Job Manager의 작업을 순차적으로 수행하여 처리 결과를 Proxy로 전송한다.

4. 스케줄링 알고리즘

스케줄링은 Scheduler에서 NG의 Proxy로 분배하는 것과 NG내의 Proxy에서 MD로 분배하는 두 부분으로 나누어진다. 전자는 일정 시간 이후에 예상되는 NG의 성능을 예측하고, 이를 고려하여 스케줄링에 적용하는 방법을 이용한다. 후자는 [8]의 스케줄링 기법을 응용하여 Proxy에 분배된 작업을 NG 내의 MD들의 성능에 맞는 크기의 작업 그룹으로 만들어 분배하는 방법을 이용한다.

4.1. 성능정보 수집 및 통계

본 연구에서는 NG에서 MD의 이용 패턴이 요일 및 시간에 따라 일정한 형태가 나타남을 이용하여 NG의 성능 기록을 통해 통계적으로 예측한다. 이를 위해 NG의 성능

을 측정하여 기록하고, 기록을 토대로 평균 성능을 예측하는 작업이 필요하다.

NG의 성능정보를 수집하는 방법은 그림 3과 같이 나타낼 수 있다. MDinfo는 MD의 성능 정보이고, APinfo는 Proxy에서 관리하는 AP 목록의 각각의 AP에 연결된 MD 정보를 나타낸다.

MD는 두 가지 이벤트를 발생시키는데, 하나는 NG의 Proxy에 로그인하여 MD의 성능정보를 Proxy에 제공하는 것이고, 다른 하나는 MD의 이동으로 AP가 변경되면서 발생하는 Handoff 이벤트이다. 이 때 Proxy에 상태 정보를 반영하여 성능정보 수집과 통계에 반영한다.

Proxy에서는 일정 주기로 AP의 정보를 수집하여 MD의 이동여부를 탐지하고, 이를 바탕으로 성능정보를 구하고, 기존 통계와 함께 다음 주기의 성능을 예측하여 Scheduler에 전송한다.

```

struct MDinfo { /* Mobile Device information */
    performance; /* performance information */
}

struct APinfo { /* AP information */
    MDinfo list; /* Mobile Device list */
}

struct AP_List { /* AP list */
    APinfo list; /* AP list */
}

EVENT mobile_device_handoff() { /* Mobile Device */
    send notification to proxy;
    change proxy's APinfo;
}

EVENT mobile_device_login() { /* Mobile Device */
    send MDinfo to proxy;
}

EVENT mobile_device_login() { /* Proxy */
    change proxy's APinfo;
}

do { /* Proxy */
    foreach(AP list) {
        get APinfo from AP;
        check incoming/outgoing MD list;
        calculate incoming/outgoing/total performance;
    }
    log performance information;
    next period's performance prediction;
    send performance information to Grid's scheduler;
    wait until next period (t + d);
} while(true)
    
```

그림 3 네트워크 그룹의 자원정보 통계 알고리즘

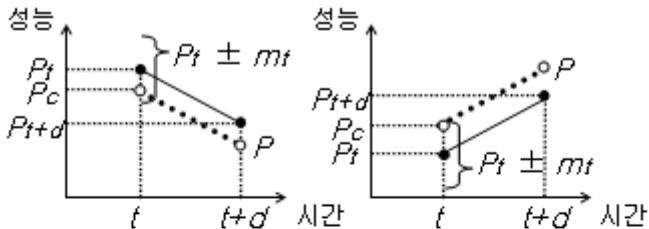
4.2. 성능예측

NG의 성능예측은 최근 n 주간 같은 요일, 같은 시간의 성능정보를 이용하여 현재시간 t 의 평균 P_t 와 표준편차

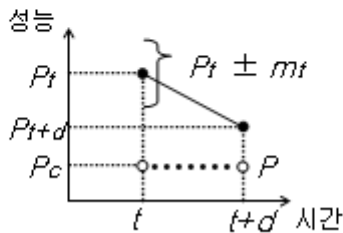
M_t , 다음 주기인 $t+d$ 의 평균 P_{t+d} 와 표준편차 M_{t+d} 값과 현재시간 측정된 성능 P_c 를 이용하여 시간 $t+d$ 의 성능 P 를 다음과 같이 예측한다.

$$\begin{cases} P = \frac{P_{t+d}}{P_t} P_c \quad (P_c \geq P_t - \alpha M_t) & \dots\dots\dots ① \\ P = P_c \quad (P_c < P_t - \alpha M_t) & \dots\dots\dots ② \end{cases}$$

여기에서 $P_t - \alpha M_t$ 조건은 통계범위를 나타내는데, 그림 4와 같이 t 시간의 평균과 표준편차를 이용하여 현재의 성능이 일반적인 경우인지를 판단하는 기준이고, α 는 통계범위를 조정하는 계수다. 만약 현재의 성능이 통계범위에 포함된다면 ①식을 사용하고, 그렇지 않은 경우 ②식을 사용한다. ①식은 성능의 평균값의 변화량을 이용해 다음 주기의 성능을 구하는 식이다. 일반적으로 같



(a) 현재 성능이 통계 범위 이상인 경우



(b) 현재 성능이 통계 범위 미만인 경우

그림 4 성능예측 기법

은 요일, 같은 시간에서 성능 값은 통계범위 내에 있겠지만, 공휴일이나 임시 휴일 같이 특별한 일이 있는 경우에는 평균보다 훨씬 낮은 값을 나타내게 된다. ②식은 이렇게 특별한 경우에 성능의 값을 예측할 수 없기 때문에 현재의 성능을 그대로 성능의 예측 값으로 이용함을 나타낸다.

그러나 평균 성능의 변화량을 이용하여 예측한 값은 NG에 참가한 MD의 유동성에 따라 크게 달라질 수 있다. 시간 t 에서 P_t 의 성능을 가진 NG가 d 시간 이후에 P_{t+d} 의 성능을 낸다고 할 때, P_{t+d} 의 성능은 NG에서 MD가 이탈하여 발생한 성능의 감소(P_{out})와 NG에 새롭게 참가한 MD로 인한 성능의 증가(P_{in})의 합으로 다음 식과 같이 나타낼 수 있다.

$$P_{t+d} = P_t + P_{in} - P_{out} \dots\dots\dots ③$$

MD의 이탈은 해당 장치에 할당된 작업을 처리하지 못하고 다른 장치에서 처리해야 하므로 작업처리시간이 지

연되어 성능의 저하로 나타난다. 따라서 실질적인 성능 예측은 MD의 유동성을 고려해야 한다.

MD의 유동성은 [7]의 연구에서와 같이 지속적인 이용자와 단속적인 이용자로 나타나고, 이들의 비율은 MD의 이용자 수의 증감에 관계없이 일정한 비율로 나타난다. 따라서 각 주기에서 MD의 이탈 비율 β 가 일정하게 나타나므로 β 의 통계정보로 P_{out} 의 값을 ④식과 같이 얻고, 이를 이용하여 NG의 성능 P 를 ⑤식과 같이 얻는다.

$$P_{out} = \beta P_c \dots\dots\dots ④$$

$$\begin{cases} P = \frac{P_{t+d}}{P_t} P_c - \beta P_c \quad (P_c \geq P_t - \alpha M_t) & \dots\dots\dots ⑤ \\ P = P_c \quad (P_c < P_t - \alpha M_t) & \end{cases}$$

4.3 스케줄링 알고리즘

스케줄링은 Scheduler에서 NG의 Proxy로 작업을 분배하는 부분과 각 NG의 Proxy에서 MD로 작업을 분배하는 부분으로 구분된다. Scheduler에서 NG의 Proxy로 작업을 분배하는 방법은 일정 주기로 각 NG에서 보내오는 성능 정보와 다음 주기의 성능예측 정보를 수집하고, 예측된 성능 정보를 기준으로 해당 주기(d 내에 처리할 수 있는 작업을 각 NG에 부과한다. Scheduler의 스케줄링 알고리즘은 그림 5와 같다.

```

struct NGinfo { /* Network Group information */
    P_t; /* current performance */
    P; /* predicted performance */
}

struct NG_List { /* NG list */
    NGinfo list; /* NG information list */
}

EVENT receive_NGinfo () {
    receive NGinfo from the proxy;
    update NG_List;
}

EVENT receive_result () {
    receive the result of job from the proxy;
    process the result;
}

do { /* Grid Scheduler */
    get current time c; /* between t and t+d */
    foreach (NG_List) {
        if (NG has no job) {
            Job_size = P × ((t+d) - c);
            send Job(size = Job_size) to NG;
        }
    }
    wait until next period (t+d);
} while(true)
    
```

그림 5 그리드 스케줄러의 스케줄링 알고리즘

NG의 Proxy에서 NG에 속한 MD에 작업을 분배하는 방법은 각 MD에서 로그인 시에 전송한 성능 정보(P_{MD})를 기준으로 일정 시간(d') 사이에 처리 할 수 있는 작업을 분배하여 처리한다. Proxy의 스케줄링 알고리즘은 그림 6과 같다.

```

struct MDInfo { /* Mobile Device information */
    performance; /* shared resource information */
}

struct APInfo { /* AP information */
    MDInfo list; /* Mobile Device list */
}

struct AP_List { /* AP list */
    APInfo list; /* AP list */
}

EVENT receive job () {
    receive the job from Grid's scheduler;
    job_processing ();
    send the result of job to Grid's scheduler;
}

EVENT receive result () {
    receive the result of job from mobile device;
    process the result;
}

job_processing () {
    get current time c; /* between t and t+d */
     $d' = \frac{(t+d) - c}{m}$ ;
    while (until t+d) {
        foreach (AP_List) {
            foreach (MD list) {
                if (MD has no job) {
                     $Job_{size} = P_{MD} \times d'$ ;
                    send Job(size =  $Job_{size}$ ) to MD;
                }
            }
        }
        wait until next period ( $c + d'$ )
    }
}
    
```

그림 6 프록시 서버의 스케줄링 알고리즘

4.4. 성능 평가

본 연구에서 제안하는 스케줄링 기법의 성능 평가를 위한 시뮬레이션에서 고려하는 환경 요소는 다음과 같다. 모바일 장치는 성능에 따라 MD_{10} , MD_5 , MD_1 의 세 부류로 구분하고, 각각의 NG는 1000개의 MD가 소속되어 있고, 모바일 그리드 전체에는 3개의 NG가 존재한다. NG의 평균 성능과 표준편차는 [7]의 'Fall/Winter 2003/4의 주간 모바일 장치의 이용 자료'를 참조하였다. 테스트를 위한 NG의 실제 장치의 이용 패턴은 표준편차를 이용하여 95% 신뢰수준에서 임의의 값을 선택하였다. 또한 비교 평가를 위하여 초기에 알려진 성능 비율에

맞게 정적으로 전체 작업을 배분하는 기본 스케줄링 기법(BS: Base Scheduling), 일정 시간 간격으로 성능 정보를 이용하여 동적으로 작업을 배분하는 최적화 스케줄링 기법(BFS: Best-Fit Scheduling), 그리고 제안하는 성능예측을 기반한 작업 스케줄링 기법(PPS: Performance Prediction based Scheduling) 등을 같은 환경에서 실행하였다.

그림 7은 10^8 개의 작업을 BS, BFS, PPS 기법으로 각각 시뮬레이션 한 결과를 나타낸다. BS의 경우 작업수행 초기에는 같은 시간에 더 많은 작업을 처리하여 효과적인 것으로 보이지만, 시간이 흐름에 따라 성능의 변화를 받

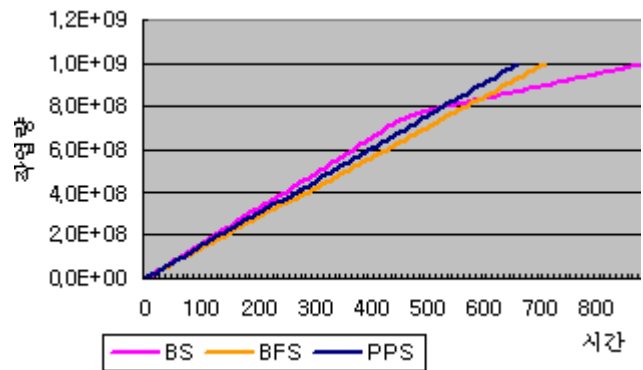


그림 7 Base, Best-Fit, Performance Prediction Scheduling 시뮬레이션 비교

영하지 못한 작업 분배로 인해 특정 NG에 작업 부하가 집중되어 최종 결과 보고가 875 단위시간으로 가장 늦었다. BFS의 경우 각 NG의 성능에 알맞게 작업을 분산시키게 되므로 BS 기법에 비해 전체 NG의 이용률을 높여 699 단위시간의 결과 보고 시간을 보임으로써 성능을 개선하였음을 보여준다. 이에 비해 PPS 기법은 성능 변화를 예측함으로써 BFS 기법보다 더욱 최적화된 성능인 651 단위시간의 결과 보고 시간을 보인다.

이상의 시뮬레이션 결과와 같이 본 논문에서 가정한 네트워크 그룹 환경에서는 PPS 기법이 가장 효율적인 스케줄링 기법임을 확인할 수 있다.

5. 결론 및 향후 과제

본 논문에서는 모바일 그리드가 갖는 제약 사항들을 해결할 수 있는 환경으로 학교나 회사와 같은 모바일 장치의 이용이 집적되고, 안정적인 이용을 나타내는 네트워크 그룹을 제시하였다. 또한 네트워크 그룹을 그리드의 자원으로 이용할 경우, 독립적이고 소규모인 작업을 처리하기에 처리하는 것이 적합함을 나타내고, 네트워크 그룹에 속한 모바일 장치의 이용 패턴이 일정하게 나타남을 이용하여 주기적으로 통계하고, 네트워크 그룹의 성능예측을 기반한 스케줄링 기법을 제시하였다.

이를 위해 성능예측을 기반한 스케줄링을 위해서 네트워크 그룹의 성능을 예측하는 방법을 제시하고, 그리드의 Scheduler와 네트워크 그룹의 Proxy 간, Proxy와 모

바일 장치 간의 두 단계로 구분하여 각각에서 작업을 스케줄링하는 알고리즘을 제시하였다. 그리고 Base Scheduling, Best-fit Scheduling과 시뮬레이션을 이용한 비교 평가를 통해 네트워크 그룹으로 구성된 모바일 그리드 환경에서 가장 효율적인 스케줄링 기법임을 보였다.

향후 연구 과제로는 모바일 장치의 높은 이질성과 다양성을 고려한 모바일 장치의 성능 측정 방법을 모색하고, 모바일 장치의 유동성으로 인한 성능 감소를 최소화할 수 있는 방법을 찾아 더 최적화된 스케줄링 기법을 제시해야 할 것이다.

참 고 문 헌

- [1] Foster and C. Kesselman, and S. Tueke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of Supercomputing Applications*, 2001.
- [2] SETI@home homepage.
- [3] The FightAIDSatHome homepage.
- [4] G. Forman and J. Zahorjan. "The Challenges of Mobile Computing", *IEEE Computer*, vol. 27, no. 4, April 1994
- [5] P. Thomas, H. Lloyd, and D. Chris. "Challenge: Integrating Mobile Wireless Device Into the Computational Grid", *MOBICOM'02*, September 23, 2002
- [6] B. Megdalena and C. Paul. "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network", *MOBISYS 1st*, May, 2003
- [7] H. Tristan, K. David, and A. Ilya, "The Changing Usage of a Mature Campus-wide Wireless Network", *MOBICOM'04*, Oct 26, 2004
- [8] M. Nithiapidary, L. Junyang, S. Nay Lin, and V. Srikumar. "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids", *AusGrid'05 Vol.44*, 2005
- [9] F. Umar and K. Wajeeha. "A Generic Mobility Model for Resource Prediction in Mobile Grids", *IEEE CTS p189-193*, 2006
- [10] Q. Min, Z. Roger, and S. Liu. "Supporting Multimedia Streaming Between Mobile Peers with Link Availability Prediction", *ACM MM'05*, Nov, 2005
- [11] C. Huang, Z. Zhu, Y. Wu, and Z. Xiao, "Power-Aware Hierarchical Scheduling With Respect to Resource Intermittence in Wireless Gids", *MLC Conference*, Aug, 2006