논문 2010-47CI-4-9

옵티컬 그리드 환경에서 DAG 계층화를 통한 스케줄링 알고리즘

(Scheduling Algorithm using DAG Leveling in Optical Grid Environment)

윤 완 오*, 임 현 수*, 송 인 성*, 김 지 원*, 최 상 방**

(Wan-Oh Yoon, Hyun-Soo Lim, In-Seong Song, Ji-Won Kim, and Sang-Bang Choi)

요 약

그리드 시스템에서 리스트 스케줄링 기반의 알고리즘을 사용한 태스크 스케줄링은 프로세서의 완전 연결된 환경에서 낮은 시간 복잡도와 높은 효율성을 보여준다. 하지만 기존 알고리즘은 태스크 간의 통신비용 및 옵티컬 그리드 환경에서 통신이 이루어지는 경로인 lightpath의 구성 과정을 충분히 고려하지 않았다. 본 논문에서는 옵티컬 그리드 환경에 최적화 된 방향성 비순환 그래프(Directed Acyclic Graph, DAG)를 계층화하여 태스크의 할당 우선순위를 결정하는 계층화 선택 알고리즘인 LSOG (Leveling Selection in Optical Grid)을 제안한다. 이 알고리즘은 동일한 계층 내 태스크들의 할당 우선순위를 결정할 때 부모 태스크와 통신비용이 가장 큰 태스크를 먼저 수행한 뒤 각각의 네트워크에서 태스크 간의 통신이 이루어지는 가장 짧은 길이의 경로를 고려한다. 이 과정은 옵티컬 그리드 환경에서 링크 리소스 사용을 최적화하여 스케줄링 과정의 통신비용을 개선시킨다. 기존의 알고리즘 중 ELSA (Extended List Scheduling Algorithm)와 SCP (Scheduled Critical Path) 알고리즘을 LSOG와 비교한 결과 CCR 값의 증가와 네트워크 환경이 원활함에 따라 전체 스케줄링 성능이 향상되는 것을 확인하였다.

Abstract

In grid system, Task scheduling based on list scheduling models has showed low complexity and high efficiency in fully connected processor set environment. However, earlier schemes did not consider sufficiently the communication cost among tasks and the composition process of lightpath for communication in optical gird environment. In this thesis, we propose LSOG (Leveling Selection in Optical Grid) which sets task priority after forming a hierarchical directed acyclic graph (DAG) that is optimized in optical grid environment. To determine priorities of task assignment in the same level, proposed algorithm executes the task with biggest communication cost between itself and its predecessor. Then, it considers the shortest route for communication between tasks. This process improves communication cost in scheduling process through optimizing link resource usage in optical grid environment. We compared LSOG algorithm with conventional ELSA (Extended List Scheduling Algorithm) and SCP (Scheduled Critical Path) algorithm. We could see the enhancement in overall scheduling performance through increment in CCR value and smoothing network environment.

Keywords: optical grid environment, task scheduling, leveling, communication, routing

I. 서 론

최근 과학기술이 발전함에 따라 일기 예보 예측 모형, 유체 역학, 우주의 미래를 예측, 전자의 파동 시뮬레이션 등의 다양한 분야에서 많은 계산을 필요로 하는 컴퓨팅 환경이 요구되고 있다. 이러한 컴퓨팅 환경을 구축하기 위해서는 대규모의 데이터베이스와 처리 알고리즘이 필수적이다. 이러한 작업들은 많은 양의 데이터베이스와 복잡한 알고리즘 계산으로 인해 단일 지역 시스템에서 수행하기가 어렵다. 따라서 이를 해결하기 위한 방법으로 그리드 시스템(grid system)이 제안되었다[1].

* 학생회원, ** 평생회원, 인하대학교 전자공학과
(Dept. of Electronic Engineering Inha University)
접수일자: 2010년4월13일, 수정완료일: 2010년7월7일

그리드 시스템은 단일 지역 시스템에서 수행되기 어려운 대용량의 컴퓨팅 문제를 분산된 컴퓨팅 자원을 활용하여 빠른 결과를 얻기 위해 개발되었으며, 그리드 리소스와 리소스 관리 시스템, 그리드 스케줄러 등으로 구성된다[2]. 분산 컴퓨팅을 수행하는 그리드 시스템의 특성상 리소스의 스케줄링이 전체 시스템의 성능을 좌우한다. 이에 따라 리소스의 효율적인 스케줄링 알고리즘이 주목받고 있으며 다양한 연구가 진행되고 있다. 그중 리스트 스케줄링(list scheduling)기법인 HEFT (Heterogeneous Earliest Finish Time)는 완전 연결(fully-connected)된 네트워크 환경에서 낮은 시간 복잡도와 짧은 전체 수행 시간을 갖는 가장 대표적인 스케줄링 알고리즘이다[5~7]. 이와 같은 기존의 알고리즘들은 완전 연결된 네트워크 환경에서 태스크의 계산비용 최소화에만 연구가 집중됐다. 그러나 그리드 시스템에서 호스트 간에 처리된 데이터 및 상태 정보의 교환은 네트워크 의존성이 강하기 때문에 태스크 계산비용뿐만 아니라 태스크 간의 통신비용도 고려해야 한다[3]. 이에 따라 네트워크 환경의 현실성을 고려한 옵티컬 그리드 환경(optical grid environment)에서의 스케줄링 알고리즘 연구가 진행되었다. 옵티컬 그리드 환경은 옵티컬 링크(optical link), 옵티컬 스위치(optical switch), 액세스 링크(access link)와 같이 3가지의 네트워크 구성 요소들을 갖는다. 옵티컬 링크는 데이터를 전송하는 경로이며, 옵티컬 스위치는 옵티컬 링크를 서로 연결한다. 그리고 액세스 링크는 옵티컬 링크를 통해 전달된 데이터를 그리드 리소스로 전달한다[4].

기존 HEFT 기반의 알고리즘들은 완전 연결된 네트워크 환경을 갖는다는 가정 하에 DAG의 bottom-level만을 고려하여 태스크 할당 우선순위를 결정하였다. 그러나 이러한 방법은 링크의 가용 상태에 따라 통신 경로가 변경되거나 통신 자체가 지연되는 상황을 고려하지 못하는 단점이 있다. 따라서 링크의 가용 상태까지 고려한 옵티컬 그리드 환경에서 낮은 시간 복잡도와 전체 수행 시간(makespan)을 갖는 HEFT 알고리즘을 이용한 연구가 진행되었으며, 가장 대표적으로 ELSA (Extended List Scheduling Algorithm)와 SCP (Scheduled Critical Path) 알고리즘이 있다[8~9]. 그러나 ELSA와 SCP 알고리즘은 옵티컬 그리드 환경을 고려하지 않은 HEFT의 bottom-level 기반의 태스크 할당 우선순위를 그대로 적용하여 옵티컬 그리드 환경의 링크들에 의해 통신이 지체되는 문제점을 안고 있다.

본 논문에서는 HEFT의 bottom-level 기반의 태스크 할당 우선순위 대신 DAG를 계층화한 뒤 계층 내 태스크들의 우선순위를 결정하는 계층화 선택 알고리즘 LSOG (Leveling Selection algorithm in Optical Grid environment)를 제안한다. 이 알고리즘은 DAG를 계층화한 뒤 동일 계층 내 태스크들의 할당 우선순위를 부모와의 통신비용을 기준으로 정렬한 뒤 통신비용이 큰 태스크부터 차례대로 리소스에 할당한다. 이 방법을 통해 태스크 간의 종속성(dependency)을 위배하지 않는 범위 내에서 통신비용이 큰 순서대로 태스크를 할당함으로써 네트워크 리소스가 통신을 수행하지 않는 유휴 시간을 줄일 수 있다.

본 논문에서 제안하는 LSOG 알고리즘과 기존의 ELSA 및 SCP 알고리즘을 대상으로 여러 개의 파라미터 값을 변화시켜 각각의 알고리즘에 대해 성능 비교를 하였다. 프로그램의 전체 수행 시간 및 DAG 상 임계경로(Critical Path, CP)의 길이 대비 실제 전체 수행 시간의 비율인 SLR (Schedule Length Ratio) 값을 기준으로 변화하는 SLR 값을 비교하여 기존 알고리즘보다 제안된 알고리즘의 성능이 향상됨을 확인하였다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 본 논문에서 사용될 환경 및 문제를 설명하고 Ⅲ장에서는 기존 태스크 스케줄링 방법과 다양한 라우팅 알고리즘을 살펴본다. 그리고 Ⅳ장에서는 본 논문에서 제안한 LSOG 알고리즘을 설명한다. Ⅴ장에서는 기존 옵티컬 그리드 시스템에서의 스케줄링 알고리즘과 LSOG 알고리즘을 시뮬레이션을 통해 비교 분석한 뒤 Ⅳ장의 결론으로 본 논문을 마무리한다.

Ⅱ. 옵티컬 그리드 환경

이 장에서는 본 논문에서 제안한 스케줄링을 수행하기 위해 필요한 옵티컬 그리드 환경에 대해서 설명하고, 옵티컬 그리드 환경에서의 스케줄링 요소를 정의한다.

2.1 옵티컬 그리드 구성요소

기본적인 옵티컬 환경 $Op(R, N, L)$은 그림 1과 같이 그리드 리소스, 옵티컬 스위치, 그리고 링크로 구성된다. 그리드 리소스 k는 r_k, 그리드 리소스의 집합은 $R = \{r_1, r_2, \cdots, r_n\}$로 표현한다. 그리드 리소스는 태스크를 수행할 수 있는 여러 수행 타입(execution type)을 가지며 $type(r_k)$로 표현한다. 그림 1에서 그리드 리소

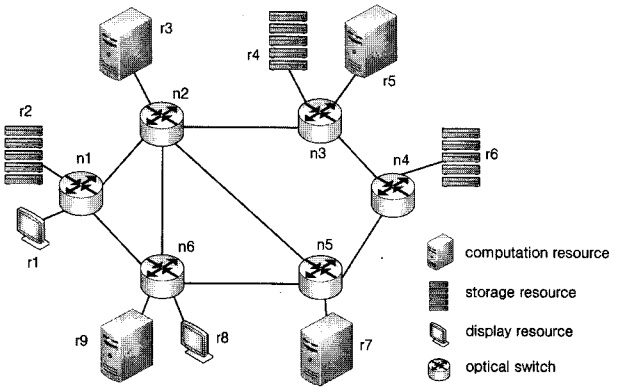

그림 1. 옵티컬 그리드 환경
Fig. 1. Optical grid environment.

스는 계산, 저장, 디스플레이의 3가지 수행 타입($type$)으로 구분되며, 계산 리소스 r_3, r_5, r_7, r_9의 타입은 $type(r_3, r_5, r_7, r_9) = 1$, 저장 리소스 r_2, r_4, r_6의 타입은 $type(r_2, r_4, r_6) = 2$, 디스플레이 리소스 r_1, r_8의 타입은 $type(r_1, r_8) = 3$으로 정의한다. 그리드 리소스는 자신의 수행 타입과 동일한 수행 타입의 태스크만을 수행할 수 있다.

옵티컬 스위치는 n_i, 옵티컬 스위치의 집합은 $N = \{n_1, n_2, \cdots, n_n\}$로 나타낸다. 옵티컬 스위치는 링크를 서로 연결하며 통신의 경로를 결정한다. 링크는 옵티컬 링크와 액세스 링크로 구분된다. 옵티컬 링크는 옵티컬 스위치를 서로 연결하며 l로 표기하고 옵티컬 스위치 n_1과 n_2를 연결하는 옵티컬 링크는 l_{12}로 나타낸다. 그리고 액세스 링크는 옵티컬 스위치와 그리드 리소스를 연결하며 l_a로 표기하고 r_3에 데이터를 전달하는 액세스 링크는 l_{a3}로 표기한다. 링크의 집합은 $L = \{l_1, l_2, \cdots l_n, l_{a1}, l_{a2}, \cdots l_{an}\}$와 같이 표현한다.

2.2 옵티컬 그리드 환경에서의 스케줄링 요소

옵티컬 그리드 환경에서 스케줄링을 수행하기 위해서는 옵티컬 그리드 환경에 적합하게 변형시킨 DAG가 필요하다. 옵티컬 그리드 환경에서 DAG는 $DAG(V, E)$로 표현한다. 여기서 V는 태스크(본 논문에서는 노드와 태스크를 같은 뜻으로 간주한다.) v의 집합이고 이러한 노드(node)의 집합은 $V = \{v_1, v_2, \cdots, v_n\}$로 표기한다. i번째 노드 v_i가 리소스 k에서의 수행 시간은 $c(v_i, r_k)$로 표기한다. 노드는 자신을 가리키는 노드 명칭 이외에 두 가지의 속성을 갖는다. 그림 2에서 각각의 원은 노드를 의미하고 노드 v_0에서 숫자 2는 노드가

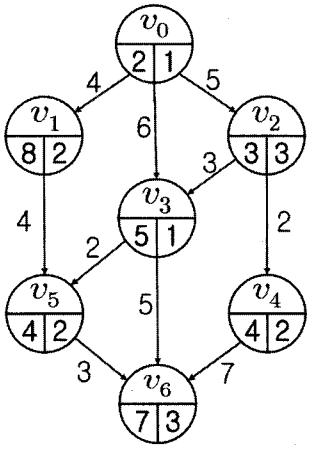

그림 2. 옵티컬 그리드에서 DAG
Fig. 2. DAG in optical grid.

할당될 수 있는 모든 그리드 리소스에서의 평균 계산비용으로써 v_i의 평균 계산비용은 $\bar{c}(v_i)$로 나타낸다. 노드 v_0에서 숫자 1은 노드의 수행 타입으로 이는 v_0가 계산 리소스 r_3, r_5, r_7, r_9에서 수행될 수 있음을 의미한다.

노드 간에 연결된 화살표는 노드 간의 통신을 의미하며 에지(edge)라 하고, v_i와 v_j을 잇는 에지는 e_{ij}로 나타낸다. 에지의 집합은 $E = \{e_1, e_2, \cdots, e_n\}$로 표기한다. 에지에 포함된 숫자는 노드 간 통신비용이며 $d(e_{ij})$로 나타낸다. 즉, 노드 v_0와 v_1을 잇는 에지 e_{01}의 통신비용이 4라면 $d(e_{01}) = 4$로 표기한다.

노드 v_i에서 v_j로의 방향성을 갖는 에지 e_{ij}가 존재한다면 v_j는 v_i의 자식 노드라 하며 v_i는 v_j의 부모노드라고 부른다. 예를 들어 노드 v_3의 부모 노드는 v_0, v_2이며, $pred(v_3) = \{v_0, v_2\}$로 나타내고 노드 v_3의 자식 노드는 v_5, v_6이며, $succ(v_3) = \{v_5, v_6\}$로 나타낸다. 노드 v_0와 같이 부모 노드가 없는 노드는 시작 노드, 노드 v_6와 같이 자식 노드가 없는 노드는 종료 노드라고 하며 스케줄링은 시작 노드부터 시작되어 종료 노드에서 끝난다.

$Op(R, N, L)$와 $DAG(V, E)$가 주어진다면 스케줄링에 필요한 계산비용, lightpath, EFT (Earliest Finish Time)를 고려할 수 있다. 먼저 r_3에 할당된 노드 v_0의 수행 시작 시간은 $t_s(v_0, r_3)$, 수행 종료 시간은 $t_f(v_0, r_3)$로 나타낸다. 수행 종료 시간과 수행 시작 시간의 차는 계산비용이라 하고 $w(v_0, r_3)$로 표현한다. 마찬가지로 e_{01}이 링크에 할당된 시간은 $t_s(e_{01})$, 할당

을 마친 시간은 $t_f(e_{01})$로 나타낸다. 그리고 할당된 통신이 점유한 모든 링크의 집합을 lightpath라고 한다. 예를 들어 그림 2의 DAG에서 노드 v_0와 v_1은 그림 1에서 각각 리소스 r_3과 r_4에 할당되었다면 통신은 r_3에 대한 액세스 링크 l_{a3}과 r_4에 대한 액세스 링크 l_{a4}, 옵티컬 스위치 n_2과 n_3를 잇는 옵티컬 링크 l_{23}를 통해 이루어진다. 즉, e_{01}의 lightpath는 $lp(e_{01})$ = {l_{a3}, l_{23}, l_{a4}}로 표현한다.

스케줄링 알고리즘은 하나의 노드를 가장 효율적으로 수행할 수 있는 리소스에 할당하는 과정을 반복한다. 즉, 노드 v_i가 리소스 r_k에서 가장 최소의 비용으로 수행을 마칠 수 있는 시간을 찾는 과정이 해당되며 그 시간을 가리켜 $EFT(v_i, r_k)$라 하고, 이것을 r_k에서 v_i의 EFT라고 한다.

III. 관련 연구

본 논문에서 제안한 LSOG 알고리즘의 성능 비교를 위해서 옵티컬 그리드 환경을 고려한 기존 스케줄링 방법인 ELSA와 SCP 알고리즘, 그리고 ELSA와 SCP의 기반이 되는 HEFT 알고리즘에 대해 설명한다. 또한 링크의 할당에서 사용되는 라우팅(routing) 알고리즘에 대해서도 살펴본다.

3.1 HEFT

HEFT 알고리즘은 노드 우선순위 결정 단계, 프로세서 선택 단계의 두 단계로 이루어진 알고리즘이다. 첫 번째 단계인 우선순위 결정 단계는 모든 노드의 bottom-level을 계산하여 값이 클수록 높은 우선순위를 갖는다. 노드 v_i의 bottom-level은 식 (3.1)에서 $B_{lv}(v_i)$로 나타내며 재귀적으로 계산된다. 여기서 $\bar{c}(v_i)$는 v_i가 할당될 수 있는 모든 리소스에서의 평균 계산비용이다. 현재 노드의 bottom-level 값은 현재 노드와 자식 노드 간의 통신비용 $d(e_{ij})$와 자식 노드의 bottom-level $B_{lv}(v_j)$를 더하여 가장 큰 값을 선택하고 다시 자신의 평균 계산비용과 더하여 결정한다.

$$B_{lv}(v_i) = \bar{c}(v_i) + \max_{v_j \in succ(v_i)}(d(e_{ij}) + B_{lv}(v_j)) \quad (3.1)$$

두 번째 단계인 프로세서 선택 단계에서는 삽입 기반 정책을 이용하여 노드를 할당한다. 삽입 기반 정책은 이미 할당된 두 노드 사이에 다른 노드를 할당할 공간이 있고, 선행 제약 조건을 위반하지 않는다면 빈 공간에 노드를 추가로 할당하는 방법이다. 이 알고리즘의 시간 복잡도는 $O(pv^2)$이며, p는 프로세서 수이다.

3.2 ELSA

ELSA는 HEFT를 옵티컬 그리드 환경에 맞게 적용한 알고리즘으로 노드 할당 우선순위 결정 과정은 HEFT와 동일하다. 하지만 최적 프로세서 선택 과정에서는 다익스트라(dijkstra) 알고리즘을 사용한 적응 라우팅 알고리즘을 통해 lightpath를 결정하는 과정이 추가된다. 이는 완전 연결된 프로세서의 집합을 대상으로 한 HEFT와 달리 ELSA가 옵티컬 그리드 환경에 적용되는 알고리즘이기 때문이다.

ELSA 알고리즘의 시간 복잡도는 다음과 같이 기술된다. DAG에서 모든 노드의 bottom-level을 계산할 때 $O(vlogv+e)$, 리소스와 링크에 노드와 에지를 할당할 때 $O(r(v+e(routing)))$, 라우팅 과정에서 다익스트라 알고리즘을 사용할 때 $O(nlogn+l)$의 시간 복잡도를 갖는다. 여기서 n는 옵티컬 스위치의 개수이며 전체 시간 복잡도는 $O(vlogv+e)$ + $O(r(v+e(nlogn+l)))$이다.

3.3 SCP 알고리즘

SCP 알고리즘은 ELSA를 통해 완성된 스케줄링 테이블에서 종료 노드를 포인터 노드로 지정한다. 이 포인터 노드에 가장 늦게 통신을 수행하는 노드를 거슬러 올라가 포인터를 옮겨가며 시작 노드까지 각 경로상의 노드를 검색한다. 이 과정에서 검색된 노드는 SCP 노드가 되며 SCP 노드의 할당 우선순위는 종속성을 위배하지 않는 범위에서 최대한 앞당기고 나머지 노드들의 우선순위는 그 이후로 조정한다.

노드들의 할당 우선순위가 결정된 다음 새로운 스케줄링 리스트가 만들어지면 이 리스트를 ELSA와 동일한 방법으로 다시 스케줄링을 한다. 그리고 새로운 할당 우선순위를 갖는 SCP 스케줄링과 ELSA 스케줄링 중 전체 수행 시간이 더 짧은 알고리즘을 최종 선택한다. 따라서 ELSA와 같거나 더 짧은 전체 수행 시간을 얻을 수 있다. SCP 알고리즘의 시간 복잡도는 $O(vlogv+e)$ + $O(r(v+e(nlogn+l)))$으로 ELSA 알고리즘의 시간 복잡도와 동일하다.

3.4 라우팅 알고리즘

라우팅 알고리즘은 고정 라우팅(Fixed Routing, FR), 고정-대체 라우팅(Fixed-Alternate Routing, FAR), 적응 라우팅(Adaptive Routing, AR)의 3가지로 분류된다. 우선 고정 라우팅은 미리 정해놓은 하나의 경로만을 사용하여 통신을 수행한다. 만약 lightpath의 링크 중 어느 하나라도 사용할 수 없는 경우에는 그 링크가 사용 가능할 때까지 기다려야 한다. 고정-대체 라우팅은 네트워크 리소스의 사용 측면에서 비효율적인 고정 라우팅의 단점을 보완하기 위해 사용한다. 이 라우팅은 미리 경로가 계산된 테이블에 이전 경로 이외에 여러 개의 대체 경로를 저장하여 사용 가능한 경로를 검색한다. 적응 라우팅은 다른 리소스와 통신이 필요한 경우 매번 새롭게 사용 가능한 경로를 검색하여 최단 경로를 결정한다. 그러므로 적응 라우팅은 옵티컬 그리드 환경과 같이 동적으로 변화하는 환경에 적합하다. ELSA와 SCP 알고리즘, 그리고 본 논문에서 제안한 LSOG 알고리즘에서 자신의 부모 태스크와 통신이 필요한 경우 최단 경로를 결정할 때 적응 라우팅을 사용하였다[10~11].

Ⅳ. 제안된 스케줄링 알고리즘

본 논문에서 제안하는 계층화 선택(Leveling Selection in Optical Grid, LSOG) 알고리즘은 DAG 계층화 단계, 우선순위 결정 단계, 노드와 통신 링크의 할당 단계로 구성된다. 먼저 노드들의 우선순위를 결정하기 전에 DAG 계층화를 수행한다[12]. 계층화는 DAG에서 시작 노드와 가까운 노드일수록 그리드 리소스에 먼저 할당될 수 있는 최소한의 조건이 된다. 이 조건에서 종속성이 없는 동일한 계층의 노드들은 우선순위가 결정되고, 마지막으로 우선순위가 높은 노드부터 적절한 리소스에 할당하기 위해서 라우팅 알고리즘을 이용하여 통신의 할당이 이루어진다.

4.1 DAG 계층화 단계

LSOG 알고리즘의 첫 번째 단계는 너비 우선 검색(Breath First Search, BFS)을 통해 DAG를 계층화하는 것이다[13]. 그림 3에서 노드들은 계층화되어 자신만의 계층을 가지며, 결정된 계층의 순서에 따라 수행한다. 노드 안의 $level$ 다음에 표기된 숫자는 노드의 계층을 의미하며 노드 v_i의 계층은 $level(v_i)$로 표기한다. 그림 3에서 v_0의 계층은 $level(v_0)$ = 1이며 v_6의 계층은 $level(v_6)$ = 5가 된다.

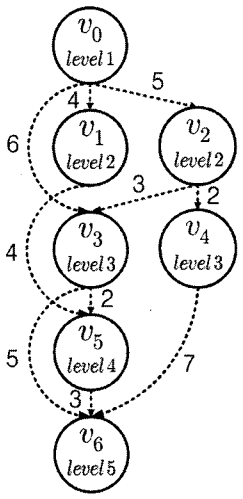

그림 3. 계층화된 노드들
Fig. 3. Nodes leveling complete.

```
//select_prio_in_level( DAG )
  set U_wait = V, Q_prio = Null
  for each level do
    extract v_j from U_wait
      which has max(d_max(v_j)) in level
    insert v_j to Q_prio
  end for
```

그림 4. 계층 내 노드 우선순위 결정 단계의 의사코드
Fig. 4. Pseudo code of phase of node priority decision within a level.

4.2 계층 내 노드 우선순위 결정 단계

그림 4는 LSOG 알고리즘의 두 번째 단계로 동일 계층의 노드들에 대해 할당 우선순위를 결정하는 과정에 대한 의사코드이다. 계층화 단계를 통해 결정된 계층 내의 노드들에 대한 할당 우선순위를 결정하기 위해서 각각의 노드는 모든 부모 노드와의 통신비용 중 가장 큰 값을 선택한다. v_i의 부모 노드들 중 통신비용이 가장 큰 값은 $d_{\max}(v_i)$로 나타낸다. 그리고 동일 계층 내에서 각 노드의 $d_{\max}(v_i)$를 기준으로 다시 큰 값을 갖는 순서대로 할당 우선순위를 정한다. 예를 들어 그림 3의 $level3$에서 $d_{\max}(v_3)$ = 6, $d_{\max}(v_4)$ = 2에 따라 할당 우선순위는 v_3, v_4가 되고, 이와 같은 방법으로 전체 노드에 대해 결정된 계층의 순서에 따라 할당 우선순위를 결정하면 v_0, v_2, v_1, v_3, v_4, v_5, v_6가 된다. 통신비용이 큰 노드에 높은 우선순위를 할당하는 것은 결정된 선후 노드 관계 내에서 통신에 의한 네트워크

```
//assign_NodeAndComm( DAG, Op )
  for each v_j∈Q_prio do
    for each r_k∈R do
      if type(r_k) = type(v_j) then
        calculate c(v_j, r_k)
        find available path about e_ij when
          v_i∈pred(v_j) by Dijkstra algorithm
        if available path at that time then
          assign e_ij to the link
        else if no available path
          delay assigning e_ij to the link
        end if
        calculate EFT(v_j, r_k)
      end if
    end for
    select r_k to minimize EFT(v_j, r_k)
  end for
```

그림 5. 노드와 통신 링크 할당 단계의 의사코드
Fig. 5. Pseudo code of node and communication link assigning phase.

리소스를 최대한 점유된 상태로 유지하기 위함이다. 따라서 네트워크 리소스는 최소한의 유휴 시간을 갖으며 그리디(greedy)한 성질을 만족시키는 형태로 전체 스케줄링이 이루어진다[13].

4.3 노드와 통신 링크의 할당 단계

LSOG 알고리즘의 마지막 단계에서는 노드 간 통신 링크 및 노드의 할당이 이루어진다. 노드와 통신 링크 할당 단계의 의사코드는 그림 5와 같으며, 세부 알고리즘은 ELSA와 동일하다[8]. 종속성에 의해 부모 노드와의 통신이 필요한 경우 결정된 lightpath의 링크들은 노드 간에 통신이 필요한 비용만큼 점유되며 모든 통신이 끝난 뒤 노드가 그리드 리소스에 할당된다[14]. 수행하려는 노드의 타입과 동일한 타입을 갖는 그리드 리소스에 노드를 할당하기 위해서 다익스트라 알고리즘을 통해 lighthpath를 할당한다. 사용 가능한 경로가 없는 경우에는 lightpath의 구성이 가능할 때까지 통신 링크의 할당을 지연시킨다. lightpath가 할당되고 나서 그리드 리소스에 노드가 할당되면 노드의 EFT를 계산할 수 있다. 위와 같은 과정을 수행하려는 노드의 타입과 동일한 모든 그리드 리소스에 동일하게 적용하여 각 리소스에 따른 EFT를 계산한다. 계산된 EFT 중 최소의 값을 갖는 리소스에 노드를 할당한다. 위와 같은 할당 단계는 노드의 계산비용이 일정하므로 lightpath의 통신비용

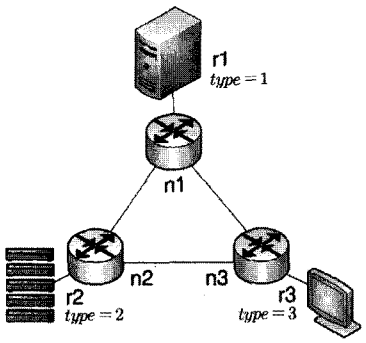

그림 6. 기본 옵티컬 그리드 환경
Fig. 6. Basic optical grid environment.

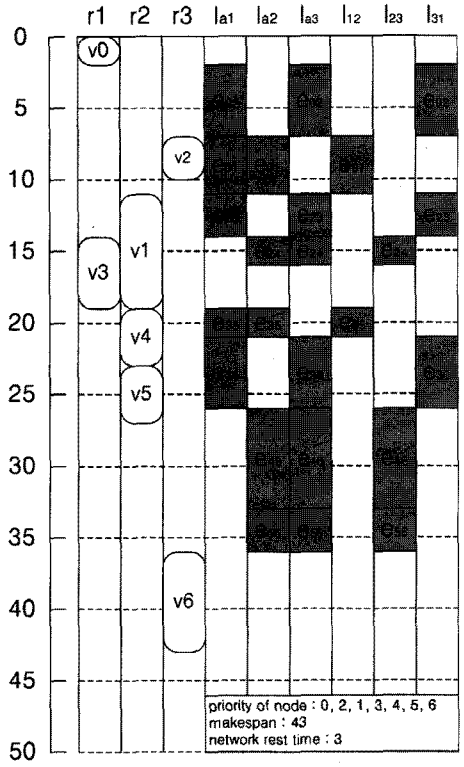

그림 7. LSOG 스케줄링 테이블
Fig. 7. LSOG scheduling table.

이 최소가 되는 리소스에 노드가 할당된다.

전체 알고리즘의 시간 복잡도는 너비 우선 검색을 이용한 DAG 계층화단계에서 $O(v+e)$, 계층 내 노드 우선순위 결정 단계에서 $O(v^2)$, 노드와 통신 링크 할당 단계에서 $O(r(v+e(n\log n+l)))$이 된다. 조밀한 그래프에서 e는 최대 v^2와 같으므로 SCP 알고리즘과 비교하면 LSOG 알고리즘의 시간 복잡도는 $O(v\log v+e) = O(v^2)$이 되므로 두 알고리즘의 시간 복잡도는 동일하다.

LSOG와 기존 SCP 알고리즘의 스케줄링 결과를 비

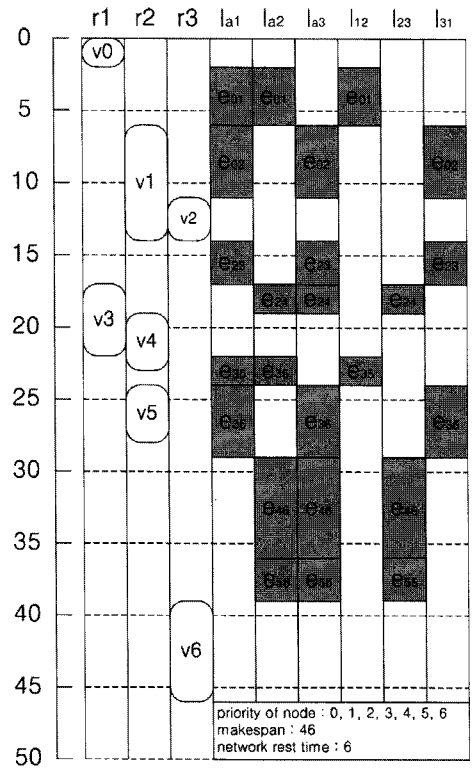

그림 8. ELSA 스케줄링 테이블
Fig. 8. ELSA scheduling table.

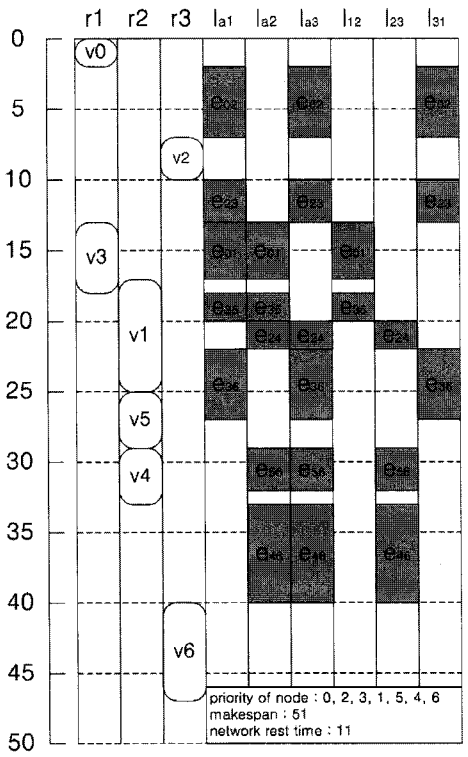

그림 9. SCP 스케줄링 테이블
Fig. 9. SCP scheduling table.

교하기 위해 그림 2의 DAG와 그림 6과 같이 하나의 수행 타입 당 하나의 리소스를 갖는 옵티컬 그리드 환경을 사용하였다. 그림 7, 8, 9에서 스케줄링 테이블의 가로축은 그리드 리소스와 링크를 나타내고 세로축은 시간을 의미한다. 그림 7에서 노드 v_0는 동일한 타입을 갖는 리소스 r_1에 할당되는 것을 시작으로 할당 우선순위를 참조하여 나머지 노드인 v_2, v_1, v_3, v_4, v_5, v_6의 순서로 그리드 리소스에 할당된다. 이 때 노드 간 통신이 필요한 경우 노드를 그리드 리소스에 할당하기 이전에 통신을 링크에 할당한다. 예를 들어 그림 7에서 노드 v_2를 자신의 수행 타입과 동일한 r_3에 할당하기 전에 부모 노드인 v_0와의 통신을 위해 e_{02}를 링크에 할당한다. e_{02}는 r_1과 r_3을 연결하는 lightpath인 $lp(e_{02}) = \{l_{a1}, l_{31}, l_{a3}\}$에 할당하며 lightpath에 통신 링크 할당이 완료되어야만 노드 v_2가 리소스 r_3에 할당될 수 있다. 이와 같은 과정을 거쳐 완성된 LSOG의 스케줄링 테이블에서 전체 수행 시간은 종료 노드인 v_6의 수행이 끝나는 43이 된다.

그림 8에서 ELSA 알고리즘의 전체 수행 시간은 46이며 그림 9에서 SCP 알고리즘의 전체 수행 시간이 47인 것을 확인할 수 있다. SCP의 전체 수행 시간이 ELSA보다 크기 때문에 SCP 알고리즘은 그림 8과 같이 ELSA의 스케줄링 테이블을 자신의 스케줄링 테이블로 결정하게 된다. 따라서 ELSA와 SCP 알고리즘 모두 전체 수행 시간은 46이 된다.

그림 7과 8에서 네트워크 리소스에 통신 링크가 할당되지 않은 유휴 시간을 비교하였을 때 LSOG 알고리즘의 유휴 시간은 e_{24}와 e_{35}의 간격인 3이 되고, ELSA의 유휴 시간은 e_{02}와 e_{23}의 간격 3과 e_{24}와 e_{35}의 간격 3이 더해진 총 6이 된다. 두 알고리즘의 전체 스케줄링 수행 시간 차이는 3이 되며, 따라서 LSOG가 네트워크 리소스의 유휴 시간을 3만큼 줄임으로써 전체 성능이 향상됨을 알 수 있다. 노드의 개수가 늘어나 스케줄링 환경이 복잡해질수록 네트워크 리소스의 유휴 시간은 증가하므로 LSOG 알고리즘이 다른 알고리즘보다 성능 향상의 폭을 넓힐 수 있는 여지가 늘어난다.

V. 성능 분석 및 평가

이 장에서는 LSOG 알고리즘의 성능을 분석하기 위해 시뮬레이션 환경을 정의하고, 제안한 알고리즘과 기

존 ELSA 및 SCP 알고리즘과의 성능 비교 결과를 기술한다.

5.1 시뮬레이션 환경

제안한 LSOG 스케줄링 알고리즘의 성능을 분석하기 위해 다양한 개수의 노드와 노드 간 통신 시간을 고려하여 임의로 생성된 DAG를 사용하였으며 대상 옵티컬 그리드 환경으로 mesh torus와 NFSnet를 가정했다[15]. 시뮬레이션에 필요한 파라미터는 표 1과 같다. 수행 타입의 개수는 3개를 적용하였고, 노드의 개수는 50, 100, 200, 300, 500으로 범위를 설정하고, 노드 당 에지의 개수는 8, 10, 12, 14, 16으로 증가시키며 DAG를 생성하였다.

$$CCR = \frac{Average\ Communication\ Cost}{Average\ Computation\ Cost} \quad (5.1)$$

CCR (Communication to Computation Ratio)은 식 (5.1)과 같으며, CCR이 커질수록 DAG에서 통신비용이 차지하는 비중이 커지므로 네트워크 리소스의 가용 상황이 중요해진다. 노드의 평균 계산비용은 2.5에서 7.5 사이에서 임의의 값을 갖는다. 여기에 그리드 리소스가 가지는 이질성(heterogeneous factor)이 더해져 노드의 계산비용이 결정된다. 노드의 개수, 노드 당 에지의 개수, 노드의 평균 계산비용, 그리고 CCR이 결정되면 에지의 평균 통신비용을 구할 수 있다. 그리드 리소스의 이질성 계수는 1 또는 2로 설정하였고 옵티컬 스위치 당 1개 또는 2개의 그리드 리소스를 연결하였다.

표 1. 시뮬레이션을 위한 입력 DAG의 파라미터
Table 1. Parameter of input DAG for simulation.

파라미터	설정 범위
노드의 수행 타입	3
노드의 개수	50, 100, 200, 300, 500
노드 당 에지의 개수	8, 10, 12, 14, 16
CCR 값	1, 1.5, 2, 2.5, 3, 5, 7
노드의 평균 계산비용	2.5 ~ 7.5
이질성 계수	1, 2
옵티컬 스위치 당 그리드 리소스 개수	1, 2

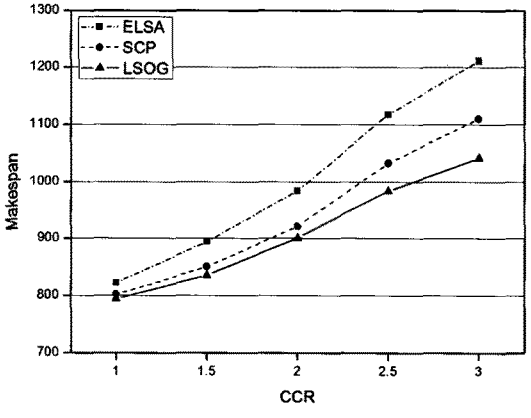

그림 10. CCR 변화에 따른 평균 전체 수행 시간
Fig. 10. Average makespan for varying CCR.

본 논문에서 제안한 LSOG 알고리즘의 스케줄링 성능 비교는 기존 알고리즘 중 ELSA와 SCP 알고리즘을 통해 이루어졌다. 성능의 척도로 정한 파라미터는 전체 수행 시간과 SLR 이다.

$$makespan = \max_{v_i \in V}(t_f(v_i)) \quad (5.2)$$

$$SLR = \frac{makespan}{\sum_{v_i \in CP} \min_{r_k \in R}\{c(v_i, r_k)\}} \quad (5.3)$$

전체 수행 시간은 종료 노드의 종료 시간이 되며 식 (5.2)와 같이 정의한다. SLR은 식 (5.3)으로 정의하며, 분모는 DAG상 임계 경로를 이루는 노드들의 최소 계산비용의 합이다. 따라서 SLR 값이 작을수록 스케줄링의 성능이 우수하다고 할 수 있다.

5.2 시뮬레이션 결과

그림 10은 ELSA, SCP, LSOG 알고리즘의 평균 전체 수행 시간이 CCR의 증가에 따라 변화하는 시간을 나타낸 것이다. 옵티컬 그리드 환경은 NFSnet이며, 각 CCR 별로 200개의 노드를 갖는 DAG 500개를 임의로 생성하였다. 생성된 DAG 500개는 각 이질성 계수의 변화에 따라 1,000번의 시뮬레이션을 수행하였다. 따라서 CCR 변화에 따라 총 5,000번의 시뮬레이션을 수행하였으며 수행한 결과 값을 이용하여 전체 수행 시간의 평균을 도출하였다. LSOG 알고리즘과 기존 알고리즘 중 성능이 나은 SCP와 비교했을 때 CCR = 1인 경우 평균 전체 수행 시간은 0.93%까지 감소하였으며 CCR 값이 순차적으로 증가함에 따라 성능 향상의 폭은 커졌다. 시뮬레이션에서 사용한 최대 CCR = 3인 경우 평균 전체 수행 시간은 6.17%까지 줄어들었다.

다음으로 평균 SLR을 비교하여 성능 향상을 분석하였다. 그림 11은 CCR = { 1, 2, 3, 5, 7 }일 때 평균 SLR이 얼마나 개선되었는지를 나타낸다. 옵티컬 그리드 환경은 mesh torus이며, 시뮬레이션의 전체 수행 반복횟수는 이전의 시뮬레이션과 동일하게 5,000번을 수

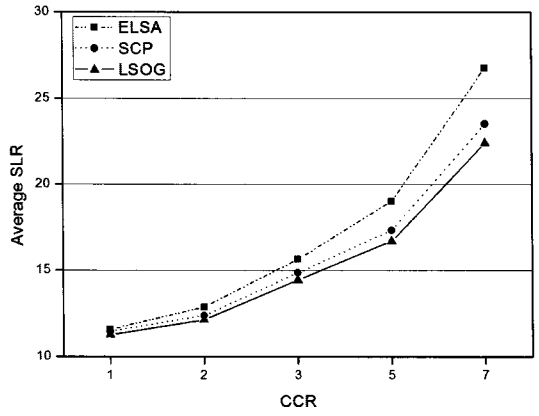

그림 11. CCR 변화에 따른 평균 SLR
Fig. 11. Average SLR for varying CCR.

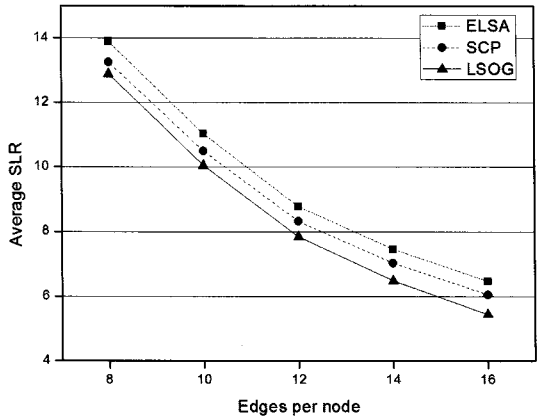

그림 12. 노드 당 에지 개수 변화에 따른 평균 SLR
Fig. 12. Average SLR for varying number of edges per node.

행하였고 수행한 결과 값으로 평균 SLR을 계산하였다. SCP에 비해 LSOG의 평균 SLR 성능 향상은 CCR = 1로 작은 경우, 즉 노드들의 계산비용과 노드 간 통신비용이 동일한 경우에는 평균 SLR의 성능 향상 폭이 1.57%로 스케줄링의 효과가 크게 나타나지 않지만 CCR 값이 7로 증가함에 따라 4.71%의 높은 성능 향상을 보인다. CCR값의 증가는 그 만큼 네트워크 리소스에 할당해야 할 통신비용이 증가한다는 의미이며, LSOG 알고리즘은 네트워크 리소스의 가용성을 최대화하는 것에 중점을 두고 있으므로 CCR값의 증가에 따라 기존 알고리즘에 비해 성능 향상의 큰 폭이 나타나게 된다.

그림 12는 그림 11과 같은 mesh torus 환경이며, CCR = 3으로 설정하였다. 노드 당 에지 개수 변화에 따라 이전의 시뮬레이션과 같이 총 5,000번을 수행하였고 수행한 결과 값을 이용하여 평균 SLR을 계산하였

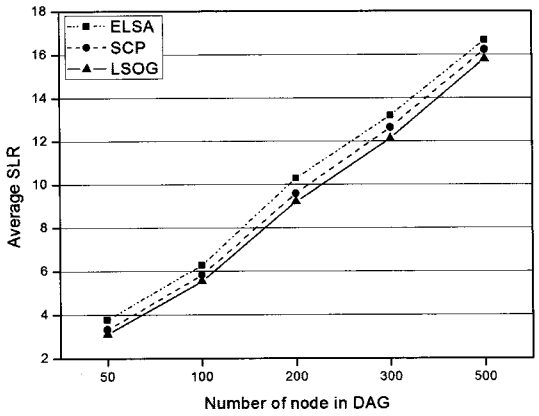

그림 13. 노드 개수 변화에 따른 평균 SLR
Fig. 13. Average SLR for varying number of node.

다. 그림 12와 같이 노드 당 에지의 개수가 증가할수록 평균 SLR의 성능 향상 폭이 3.54%, 4.85%, 5.77%, 6.69%, 8.34%의 비율로 늘어나는 것을 확인할 수 있다. 즉, 노드 당 에지의 개수가 증가하면 통신비용 스케줄링을 통해 기대할 수 있는 성능 향상의 폭이 늘어나게 된다.

그림 13은 CCR = 3으로 설정하고, 노드 당 평균 10개의 에지를 갖는 DAG 500개를 임의로 생성하고 이질성 계수의 변화에 따라 총 5,000번의 시뮬레이션을 수행하였다. 그림 13의 시뮬레이션에서 노드의 개수를 늘려간다면 그만큼 노드 당 에지의 수가 감소한다. 따라서 통신비용 스케줄링을 통해 기대할 수 있는 성능 향상의 폭이 6%, 4.8%, 4.06%, 3.57%, 2.65%의 비율로 줄어들게 된다. 이것은 LSOG가 중점을 두는 네트워크 리소스 활용의 폭이 좁아지는 것을 의미한다.

IV. 결 론

본 논문은 옵티컬 그리드 환경에서 태스크 스케줄링을 위해 우선순위 결정 방법에서 계층화 선택을 하는 LSOG 알고리즘을 제안하고 시뮬레이션을 통해 기존에 제안된 ELSA 및 SCP 알고리즘과 성능을 비교하였다. 입력 그래프는 DAG를 사용하였으며 여기에 각각의 노드와 그리드 리소스에 수행 타입이라는 요소를 추가함으로써 리소스에서 수행될 수 있는 노드의 자격 요건을 설정하고 옵티컬 그리드 환경에서 노드 간 통신 링크 스케줄링의 필요성을 중점적으로 고려하였다.

옵티컬 그리드 환경의 특성 상 노드의 계산비용보다는 노드 간 통신비용이 전체 스케줄링에서 차지하는 비

중이 더 크며 LSOG는 이를 반영하기 위해 bottom-level 우선순위 결정 방법 대신 계층화를 통한 우선순위 결정 방법을 사용하였다. 이로 인해 노드와 통신 링크의 할당이 계층 간 우선순위에 따라 DAG 상에서 순차적으로 이루어진다. 따라서 통신 링크 할당에 뒤따르는 라우팅 과정을 효율적으로 고려할 수 있다. 한편, 동일 계층에서의 노드 간 우선순위 결정 과정에서 통신비용을 중요한 파라미터로 사용할 수 있고 이를 통해 lightpath를 구성할 때 다양한 경로를 선택할 수 있는 기회를 마련하였다. 또한 다익스트라 알고리즘을 사용한 라우팅 경로의 최소화 과정으로 네트워크 리소스를 최적화시켰다. 이러한 시도는 기존 ELSA 및 SCP 알고리즘에 비해 전체 수행 시간과 평균 SLR 값이 각각 최대 6.17%, 4.71%까지 줄어들었으며, 동일한 LSOG 알고리즘을 적용하였을 때 옵티컬 그리드의 통신 환경에 따라 그 폭이 달라지는 것을 확인하였다.

참 고 문 헌

[1] I. Foster and C. Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure," 1st edition, Morgan Kaufmann Publishers. Aug. 1998.

[2] A. Sahara and T. Ono and J. Yamawaku and A. Takada and S. Aisawa and M. Koga, "Congestion-Controlled Optical Burst Switching Network With Connection Guarantee: Design and Demonstration," IEEE Journal of Light Technology, vol. 26, no. 14, pp. 2075-2086, Jul 2008.

[3] O. Sinnen and L. Sousa, "List scheduling: extension for contention awareness and evaluation of node priorites for heterogeneous cluster architectures," IEEE Parallel Computing, vol. 30, pp. 81-101, 2004.

[4] D. Simeonidou and R. Nejabati and G. Zervas and D. Klonidis and A. Tzanakaki and M. J. Mahony, "Dynamic optical-network architectures and technologies for existing and emerging grid services," Journal of Lightwave Technology, Journal of Lightwave Technology, vol. 23, issue. 10, pp. 3347-3357, Oct 2005.

[5] G. C. Sih and E. A. Lee, "A Compile-Time Scheduling Heuristic for Interconnection - Constrained Heterogeneous Processor Architectures," IEEE transaction on parallel and distributed system, vol. 4, no 2, pp. 175-187, Feb 1993.

[6] H. Topcuoglu and S. Hariri and M. Wu, "Performance - Effective and Low - complexity Task Scheduling for Heterogeneous Computing," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 2, pp. 1273-1284, Mar 2002.

[7] I. Ahmad and Y. K. Kwok, and M. Y. Wu, "Amalysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors," Parallel Architectures, Algorithms, and Networks, vol. 12, pp. 207-213, June 1996.

[8] Y. Wang and Y. Jin and W. Guo and W. Sun and W. Hu, "Joint scheduling for optical grid applications, " Journal of Optical Networking, Vol. 6, Issue 3, pp. 304-318, Mar 2007.

[9] Z. Sun and W. Guo and Z. Wang and Y. Jin and W. Sun, "Scheduling Algorithm for Workflow-Based Applications in Optical Grid," IEEE Journal of Lightwave Technology, vol. 26, no. 17, pp. 3011-3020, Sep 2008.

[10] H. Zang and J. P. Jue and B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks," IEEE Optical Network Magazine, vol. 23, issue 9, Jan 2000.

[11] A. Sahara and T. Ono and J. Yamawaku and A. Takada and S. Aisawa and M. Koga, "Congestion-Controlled Optical Burst Switching Network With Connection Guarantee : Design and Demonstration," IEEE Journal of Light Technology, vol. 26, no. 14, pp. 2075-2086, Jul 2008.

[12] Y. C. Lee and R. Subrata and A. Y. Zomaya, "On the Performance of a Dual-Objective Optimization Model for Workflow Applications on Grid Platforms," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 9, Sep 2009.

[13] T. H. Cormen and C. E. Lesiserson and R. L. Rivest and C. Stein, "Introduction to algorithms," The MIT Press, second edition, pp. 394-434, 2008.

[14] X. Liang and X. Lin and M. Li, "Adaptive Task Scheduling on Optical Grid," IEEE Asia-Pacific Conference, pp. 486-491, Dec 2006.

[15] D. Cavendish and A. Kolarov and B. Sengupta, "Routing and Wavelength Assignment in WDM Mesh Networks," IEEE Communications Society Globecom, pp. 1016-1023, Aug 2004.

저 자 소 개

윤 완 오(학생회원)
2000년 경기대학교 전자공학과 학사 졸업.
2002년 인하대학교 전자공학과 석사 졸업.
2002년~현재 인하대학교 전자공학과 박사과정.
<주관심분야 : 분산 처리 시스템, 병렬프로그램밍, 컴퓨터 구조, 무선 통신, 컴퓨터 네트워크>

송 인 성(학생회원)
2009년 인하대학교 전자공학과 학사 졸업.
2009년~현재 인하대학교 전자공학과 석사과정.
<주관심분야 : 분산 처리 시스템, 병렬프로그램밍, 컴퓨터 구조>

최 상 방(평생회원)
1981년 한양대학교 전자공학과 학사 졸업.
1981년~1986년 LG 정보통신(주).
1988년 University of washinton 석사 졸업.
1990년 University of washinton 박사 졸업.
1991년~현재 인하대학교 전자공학과 교수.
<주관심분야 : 컴퓨터 구조, 컴퓨터 네트워크, 무선 통신, 병렬 및 분산 처리 시스템>

임 현 수(학생회원)
2009년 인하대학교 전자공학과 학사 졸업.
2009년~현재 인하대학교 전자공학과 석사과정.
<주관심분야 : 분산 처리 시스템, 병렬프로그램밍, 컴퓨터 구조, 컴퓨터 네트워크>

김 지 원(학생회원)
2007년 건양대학교 전자정보 공학과 학사 졸업.
2009년 인하대학교 전자공학과 석사 졸업.
2010년~현재 인하대학교 전자공학과 박사과정.
<주관심분야 : 멀티미디어 통신, 무선 통신, 센서 네트워크, 컴퓨터 네트워크>