JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 12 : 2981~2986 Dec. 2015

마감시간을 가진 요청들에 대한 브로드캐스트 스케줄링의 자원추가 분석

김재훈*

Resource Augmentation Analysis on Broadcast Scheduling for Requests with Deadlines

Jae-hoon Kim*

Department of Computer Engineering, Busan University of Foreign Studies, Busan 609-780, Korea

요 약

본 논문에서는 브로드캐스트를 수행할 수 있는 m 개의 서버가 존재하는 경우에 마감시간이 있는 요청들을 만족시키는 스케줄링 문제를 다룰 것이다. 서버가 어떤 페이지를 브로드캐스트하면 이 페이지를 요구한 모든 요청들은 만족된다. 스케줄링 알고리즘은 매 시간에 서버에서 브로드캐스트 할 페이지를 결정한다. 알고리즘의 목표는 마감시간 안에 만족된 요청들의 가중치 합을 최대로 하는 것이다. 온라인 알고리즘의 성능은 입력을 미리 다 알고 결정을 내리는 최적 오프라인 알고리즘의 성능과 비교된다. 일반적으로 최적 오프라인 알고리즘의 성능이 월등히 뛰어 나기 때문에 온라인 알고리즘이 보다 많은 자원을 이용할 수 있는 자원추가 분석 방법을 사용한다. 본 논문에서는 온라인 알고리즘이 보다 많은 서버를 사용하는 경우를 다룰 것이다.

ABSTRACT

In this paper, there are m servers to carry out broadcasts and the scheduling problem to serve the requests with deadlines is studied. If a server broadcasts a page, then all the requests which require the page are satisfied. A scheduling algorithm shall determine which pages are broadcasted on servers at a time. Its goal is to maximize the sum of weights of requests satisfied within their deadlines. The performance of an on-line algorithm is compared with that of the optimal off-line algorithm which can see all the inputs in advance. In general, the off-line algorithms outperform the on-line algorithms. So we will use the resource augmentation analysis in which the on-line algorithms can utilize more resources. We consider the case that the on-line algorithms can use more servers in this paper.

키워드 : 스케줄링, 온라인 알고리즘, 오프라인 알고리즘, 브로드캐스트, 자원추가 분석

Key word : scheduling, on-line algorithm, off-line algorithm, broadcast, resource augmentation analysis

Received 28 August 2015, **Revised** 16 September 2015, **Accepted** 30 September 2015

*** Corresponding Author** Jae-hoon Kim(E-mail:jhoon@bufs.ac.kr, Tel:+82-51-509-6226)
Department of Computer Engineering, Busan University of Foreign Studies, Busan 609-780, Korea

Open Access **http://dx.doi.org/10.6109/jkiice.2015.19.12.2981** print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

m개의 서버가 존재하고 각 서버는 매 시간에 n개의 페이지 중 하나를 브로드캐스트한다. 여기서 페이지를 브로드캐스트한다는 것은 이 페이지를 서비스 받고자 하는 모든 요청들을 동시에 만족시킬 수 있음을 의미한다. 이것이 일반적인 스케줄링 모델과의 차이점이다. 일반적인 스케줄링 모델에서는 어떤 순간에는 하나의 서버에 하나의 작업만이 스케줄되어 서비스된다. 다시 말해서, 어떤 순간 서버는 하나의 작업만을 만족시킬 수 있다. 하지만 본 논문에서 다루는 모델에서는 페이지를 요구하는 모든 요청들은 브로드캐스트를 통해서 동시에 만족될 수 있다. 이런 브로드캐스트 기술은 높은 대역폭을 요구하는 케이블 TV, 인공위성, 무선 네트워크에서 사용되고 있다. 예를 들어, DirecPC 시스템에서 클라이언트들은 전화로 요청을 하고 서버는 위성을 통한 브로드캐스트로 요청들을 만족시킨다.

페이지들은 모두 동일한 크기를 가진다. 다시 말해서, 서버가 페이지들을 브로드캐스트하는 시간은 모두 같다. 편의상 모든 페이지의 크기를 단위 시간이라고 가정한다. 이산시간(discrete time) 모델의 경우에는 시간을 단위 시간 구간들로 나눠서 단위 구간의 끝점들을 0, 1, ..., T로 표현한다. 그러면 시간 t (= 0, ..., $T-1$)에, 다시 말해서, 각 구간의 시작에, 페이지에 대한 요청들이 도착한다. 그리고 각 서버에 스케줄된 페이지들은 이 단위 시간 구간동안 브로드캐스트된다. 연속시간(continuous time) 모델의 경우에는 연속된 시간상의 어떤 지점에서도 요청들이 도착할 수 있고 임의의 시간에 페이지를 스케줄해서 서비스할 수 있다.

각 요청은 중요도를 나타내는 가중치 값을 가지고 마감시간을 가진다. 요청은 마감시간 이전에 서비스 받을 것을 요구한다. 따라서 스케줄링 알고리즘의 성능은 마감시간 안에 서비스 받은 요청들의 가중치 합으로 분석한다.

본 논문에서는 온라인(on-line) 스케줄링 알고리즘을 다룰 것이다. 온라인 알고리즘은 미래에 도착할 입력에 대한 정보를 미리 알지 못하고 현재 알려진 정보만을 가지고 결정을 내려야 한다. 이것은 모든 입력 정보를 알고 결정을 내리는 일반적 오프라인(off-line) 알고리즘과 대비된다. 그러면 이러한 입력 정보의 부재 속에서 결정을 내려야 하는 온라인 알고리즘의 성능은 최적 오프라인 알고리즘의 성능과 비교된다. 임의의 입력 집합 I에 대해서, $A(I)$를 온라인 알고리즘 A의 성능이라고 하자. 본 논문에서 $A(I)$는 마감시간 안에 서비스 받은 요청들의 가중치 합이 된다. 그리고 $OPT(I)$를 최적 오프라인 알고리즘의 성능이라고 하자. 다음 식을 만족하면 온라인 알고리즘 A는 c-competitive 라고 한다:

모든 입력 집합 I에 대해서,

$$OPT(I) \leq cA(T).$$

그러면 온라인 알고리즘 A의 경쟁비(competitive ratio) σ는 다음과 같이 정의된다:

$$\sigma = \max_I \frac{OPT(I)}{A(I)}.$$

경쟁비가 1에 가깝게 작을수록 좋은 온라인 알고리즘이라고 생각할 수 있다. 하지만 일반적으로 입력 정보의 부재로 인해서 많은 온라인 알고리즘들의 경쟁비은 굉장히 크다. 따라서, 오프라인 알고리즘에 비해서 보다 많은 자원(resource), 예를 들어, 보다 빠른 속도의 서버 또는 보다 많은 서버 등을 온라인 알고리즘에 제공하는 모델이 연구되었다. 본 논문에서는 온라인 스케줄링 알고리즘이 보다 많은 수의 서버를 사용할 수 있는 경우를 다룬다.

본 논문에서는 그리디(greedy) 형태의 온라인 알고리즘들의 성능을 분석할 것이다. 이 알고리즘들이 m개의 서버를 사용할 수 있는 경우에 1개의 서버를 사용하는 최적 오프라인 알고리즘과의 경쟁비를 분석할 것이다. 이산시간 모델의 경우에 m개의 서버를 사용하는 경쟁비가 $1+\frac{1}{m}$인 온라인 알고리즘을 제안할 것이다. 또한 연속시간 모델의 경우에 2개의 서버를 사용하는 2-competitive 인 알고리즘을 제안할 것이다.

Ⅱ. 관련 연구

일반적으로 스케줄링 알고리즘은 기계(machine)상에 작업을 스케줄할 때, 한 순간에 한 기계에 하나의 작업만을 수행하도록 하는 모델을 생각한다. 하지만 브로

드캐스트 스케줄링은 한 순간에 여러 개의 작업을 수행할 수 있는 모델이다. 이 브로드캐스트 스케줄링에 대한 연구는 주로 플로타임(flow time)에 대한 문제에 집중되었다. 요청의 플로타임이란 요청의 도착시간과 요청의 서비스 완료시간 사이를 말한다. 다시 말해서, 요청이 도착해서 서비스 받을 때까지 얼마나 오래 걸렸는지를 나타낸다. 특별히 요청들의 플로타임들의 총 합을 최소화하는 문제가 연구되었다[1-4]. [1]에서 저자들은 어떤 결정적 온라인 알고리즘도 $O(1)$-competitive 일 수 없음을 증명하였다. 오프라인 알고리즘의 경우에는 이 문제가 NP-hard임이 보여졌다[2]. 또한 최대 플로타임을 최소화하는 문제도 연구되었다[5, 6].

온라인 스케줄링 연구에서 마감시간을 가진 작업들에 대한 많은 연구들이 있었다[7-9]. [7]에서는 선점(preemptive) 스케줄링의 경우에 임의의 온라인 알고리즘이 4-competitive 보다 좋을 수 없다는 것을 보였다. [8]에서는 비선점(nonpreemptive) 스케줄링의 경우를 연구하였다. 브로드캐스트 스케줄의 경우에도 마감시간을 가진 요청들의 스케줄링 대한 연구들이 있었다[5, 10]. [5]에서 모든 요청들을 마감시간 안에 만족시킬 수 있는 브로드캐스트 스케줄이 존재하는 지 여부를 결정하는 오프라인 알고리즘에 대한 연구가 있었다. [10]에서는 만족된 요청들의 퍼센트를 최대화하는 문제를 연구하였다. 하지만 그들은 요청들의 확률분포를 가정하고 분석하는 방법을 사용하였다. 본 연구에서는 마감시간을 가진 요청들의 브로드캐스트 스케줄링 문제에서 온라인 알고리즘에 대한 연구를 시행할 것이다.

자원추가 분석 모델은 [11]에서 처음으로 제안되었다. 저자들은 작업 스케줄링 문제에 대해서 온라인 알고리즘이 최적 오프라인 알고리즘보다 더 빠른 속도의 기계를 가질 수 있는 경우에 경쟁비를 분석하였다. 이 후에 이 모델은 그 밖의 여러 다른 온라인 스케줄링 문제에 적용되었다[12, 13].

Ⅲ. 알고리즘 및 분석

우선 브로드캐스트 스케줄링의 이산시간(discrete time) 모델을 다룰 것이다 다시 말해서, 매 시간 t에서, 새로운 요청들이 도착하고, 스케줄러는 서비스할 페이지들을 선택한다. 서버들은 선택된 페이지들을 브로드캐스트함으로서 이 페이지들에 대한 모든 요청들을 단위 시간 구간 $[t, t+1)$안에 모두 만족시킨다. 우리는 이산 시간 모델에 대해서 m개의 서버를 가진 브로드캐스트 스케줄링 온라인 알고리즘 G의 성능을 분석할 것이다. 알고리즘 G는 다음과 같이 동작 한다:

G : 매 시간 t에서, 페이지의 요청들의 가중치의 합이 큰 순서대로 m개의 페이지들 $p_1, p_2, ..., p_m$를 선택해서 각 서버에서 한 페이지씩 브로드캐스트한다.

다음 정리 3.1에서는 알고리즘 G가 1개 서버를 가진 최적 알고리즘에 대해서 $(1+\frac{1}{m})$-competitive 임을 보일 것이다.

정리 3. 1 m개 서버를 가진 알고리즘 G는 $(1+\frac{1}{m})$-competitive 이다.

증명. A를 알고리즘 G가 만족시키는 요청들의 집합이라 하고, OPT를 최적 알고리즘이 만족시키는 요청들의 집합이라고 하자. 알고리즘 G의 수행 특성에 의해서 연속된 구간 T_i, $i = 1, ..., h$, 가 존재해서 각 T_i에서 알고리즘 G는 모든 서버에 페이지를 스케줄하고 이 페이지들의 요청들을 만족시킨다.

우리는 우선 요청 $r \in OPT - A$을 고려한다. 다시 말해서, 요청 r은 최적 알고리즘에 의해서 만족되지만 알고리즘 G에 의해서는 거절된다. 그러면 요청 r은 어떤 구간 T_j에서 만족됨을 알 수 있다. 다시 말해서, 만약 요청 r이 구간 T_i들 밖의 시간 t에서 스케줄 되었다고 가정하자. 시간 t에서 알고리즘 G의 m개의 서버 중 적어도 하나는 서비스하는 페이지가 없어야 한다. 그러면 그 서버에서 요청 r을 서비스할 수 있기 때문에 요청 r은 알고리즘 G에 의해 만족되었어야 한다. 따라서 요청 $r \in OPT - A$은 적어도 어떤 구간 T_j 안에서 만족되어야 한다. 요청 $r \in OPT - A$이 어떤 구간 T_j 안의 시간 t에서 최적 알고리즘에 의한 페이지 p의 브로드캐스트에 의해 만족되어진다고 가정하자. 이 때, 만족되어진 $OPT - A$안의 모든 요청들의 가중치의 합을 w라고 하자. 이 요청들은 시간 t까지 알고리즘 G에 의해 스케줄 되지 않았기 때문에 시간 t에서 G에 의해 고려된다. 하지만 알고리즘 G는 더 큰 가중치 합 $w_1, ..., w_m$을 가지는 p가 아닌 페이지들을 스케줄 하였다. 따라서

$$w \le w_i, i = 1, ..., m.$$

결과적으로 다음과 같은 부등식을 얻을 수 있다:

$$m \| OPT - A \| \le \| A \|,$$

여기서, 임의의 요청들의 집합 S에 대해서, $\| S \|$는 S에 속하는 요청들의 가중치 합을 나타낸다. 그러면 위 식으로부터 우리는 다음과 같은 결과를 얻을 수 있다:

$$\begin{aligned} \| OPT \| &= \| OPT \cap A \| + \| OPT - A \| \\ &\le \| A \| + \frac{1}{m} \| A \| \\ &= (1 + \frac{1}{m}) \| A \|. \end{aligned}$$

정리 3.1에서 온라인 알고리즘 G의 경쟁비의 상한을 증명하였다. 다음에서 우리는 G의 경쟁비의 하한을 보일 것이다. 결과적으로 $1 + \frac{1}{m}$가 알고리즘 G의 경쟁비임을 보일 것이다.

정리 3.2 m개 서버를 가진 알고리즘 G는 $\Omega(1 + \frac{1}{m})$-competitive 이다.

증명. m개의 페이지 $p_1, p_2, ..., p_m$이 존재한다고 가정하고 T를 길이가 1보다 큰 임의의 시간 구간이라고 하자. 시간 0, 1, ..., T-1에 각 페이지 p_i를 요청하는 m개의 요청들이 도착한다. 이 요청들은 가중치 1과 무한대의 마감시간을 가진다. 알고리즘 G는 매 시간 각 서버에 페이지 하나씩 스케줄해서 이 m개의 요청들을 만족시킨다. 또한 시간 0에 페이지 p_1을 요청하는 T개의 요청들이 도착한다. 이 요청들은 모두 가중치 $1 - \varepsilon$과 마감시간 T를 가진다. 여기서 ε은 충분히 작은 양의 실수이다. 이 요청들은 가중치 값이 이전 요청들보다 작기 때문에 알고리즘 G는 이 요청들을 거절하고 시간 T안에 서비스하지 못한다. 하지만 최적 오프라인 알고리즘은 시간 0, 1, ..., $T-1$에서 가중치 $1 - \varepsilon$인 요청들을 만족시키고 시간 T이후에 가중치 1인 요청들을 만족시킬 수 있다. 따라서 알고리즘 G가 만족시키는 요청들의 가중치 합은 mT이고 최적 알고리즘이 만족시키는 요청들의 가중치 합은 $(1-\varepsilon)T + mT$이다. 이것이 알고리즘 G의 경쟁비의 하한이 $1 + \frac{1}{m}$임을 보인다.

정리 3.1과 3.2를 통해서 우리는 온라인 알고리즘 G의 경쟁비가 $1 + \frac{1}{m}$임을 보였다. 다음으로 우리는 브로드캐스트 스케줄링의 연속시간(continuous time) 모델을 다룰 것이다. 이 모델에서는 요청들이 연속된 시간상의 어떠한 지점에서도 도착할 수 있다. 다시 말해서, 어떤 요청 r_1이 구간 $[t, t+1)$에서 서비스 받고 있다고 하자. 그러면 새로운 요청 r_2가 구간 $[t, t+1)$안에서 도착 할 수 있다. 그러면 알고리즘은 임의의 시간에 페이지를 스케줄해서 서버에서 서비스를 시작할 수 있다. 또한 위의 경우처럼 r_2가 구간 $[t, t+1)$안의 시간 s에서 도착할 때, 시간 s에서 서비스 받고 있는 페이지 r_1을 중간에 멈추고 r_2가 요청한 페이지를 브로드캐스트 할 수도 있다. 이것을 재시작(restart)이라고 한다. 이 때 알고리즘의 성능은 끝까지 서비스 받은 요청들의 가중치 합으로 한다.

여기서 2개의 서버 s_1과 s_2를 사용하는 온라인 알고리즘 H를 생각한다. 시간을 단위시간들의 구간 $T_i = [i, i+1)$으로 나눈다. 그러면 서버 s_1은 짝수 구간 $T_0, T_2, ...$을 책임지고 서버 s_2는 홀수 구간 $T_1, T_3, ...$을 책임진다.

서버 s_1은 임의의 짝수 구간 T_i안에 도착하는 요청들에 대해서 같은 페이지를 요구하는 요청들의 가중치 합이 가장 큰 페이지를 브로드캐스트 한다. 이것은 다음과 같이 구현할 수 있다. 구간 T_i의 시간 t에서 새로운 요청 r이 도착하고 이것이 요구하는 페이지가 p라고 하자. 현재 시간 t에서 서비스되는 페이지가 p'이고 이 서비스를 받고 있는 요청들의 가중치 합을 w'이라고 하면, 시간 t에 도착해 있고 페이지 p를 요구하고 마감시간이 $t+1$이상인 요청들을 생각한다. 이 요청들의 가중치 합이 w라 할 때, $w > w'$이면 현재 서비스를 중지하고 새롭게 페이지 p를 브로드캐스트한다. 구간 T_i의 끝 시간 $i+1$까지 이 규칙을 적용하면 시간 $i+1$에는 구간 T_i안에 도착한 요청들에 대해서, 요구하는 요청들의 가중치 합이 가장 큰 페이지가 서비스된

다. 그러면 다음 구간 T_{i+1}에는 어떤 페이지도 브로드캐스트하지 않는다. 서버 s_2는 임의의 홀수 구간에 대해서 위와 같이 동작한다.

정리 3. 3 2개 서버를 가진 알고리즘 H는 2-competitive 이다.

증명. B를 알고리즘 H가 만족시키는 요청들의 집합이라 하고, OPT를 최적 알고리즘이 만족시키는 요청들의 집합이라고 하자. 요청 $r \in OPT - B$를 생각해보자. 요청 r이 최적 알고리즘에 의해서 서비스 받는 시간 t는 알고리즘 H에서 두 서버 중 하나가 서비스하는 구간 안에 있어야 한다. 왜냐하면 이러한 구간 밖에 있다면 알고리즘 H에서 두 서버가 모두 서비스 하지 않는 경우이고 요청 r을 서비스할 수 있기 때문이다.

시간 t에서 알고리즘 H는 두 서버 중 하나에서 페이지 p를 브로드캐스트하고 서비스 받는 요청들의 가중치 합을 w라고 하자. 요청 r과 같은 페이지를 요구하고 $OPT - B$에 속하면서 시간 t에 최적 알고리즘에 의해 서비스 받는 요청들은 알고리즘 H에 의해 고려된다. 이 요청들의 가중치 합을 w'라 할 때, 알고리즘 H는 이 요청들을 거절하기 때문에 $w > w'$이다. 따라서 다음 식을 얻는다:

$$\| OPT - B \| \leq \| B \| .$$

위 부등식으로부터 다음과 같은 결과를 얻을 수 있다:

$$\| OPT \| = \| OPT \cap B \| + \| OPT - B \|$$
$$\leq 2 \| B \| .$$

Ⅳ. 결 론

본 논문에서는 브로드캐스트 스케줄링 문제 중에서 요청들이 마감시간을 가지는 문제를 다루었다. 우선 요청들이 단위시간 구간의 시작점에서만 도착하고 서버는 이 구간 동안 서비스하는 이산시간 모델을 생각하였다. 이 모델에 대해서 m개의 서버를 가지고 경쟁비가 $1 + \frac{1}{m}$인 온라인 알고리즘을 제안하였다. 또한 요청들이 임의의 시간에 도착할 수 있고 서비스도 임의의 시간에 이루어지는 연속시간 모델도 다루었다. 이 모델에 대해서는 2개의 서버를 가진 2-competitive 알고리즘을 제안하였다. 향후에 연속시간 모델에 대한 m개 서버를 가지는 알고리즘의 연구와 요청들의 서비스 시간이 단위시간이 아닌 임의시간인 경우의 연구가 필요할 것이다.

ACKNOWLEDGMENTS

This work was supported by the research grant of the Busan University of Foreign Studies in 2015

REFERENCES

[1] B. Kalyanasundaram, K. Pruhs, M. Velauthapillai, “Scheduling broadcasts in wireless networks," in *Proceeding of the 8th European Symposium on Algorithms*, pp. 290-301, 2000.

[2] T. Erlebach, A. Hall, “NP-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow," in *Proceeding of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 194-202, 2002.

[3] J. Edmonds, K. Pruhs, “Broadcast scheduling: when fairness is fine," in *Proceeding of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 421-430, 2002.

[4] R. Gandhi, S. Khuller, Y. A. Kim, Y. C. Wan, “Algorithms for minimizing response time in broadcast scheduling," in *Proceeding of the 9th International Symposium on Integer Programming and Combinatorial Optimization*, pp. 425-438, 2002.

[5] Y. Bartal, S. Muthukrishnan, “Minimizing maximum response time in scheduling broadcasts," in *Proceeding of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 558-559, 2000.

[6] J. Chang, T. Erlebach, R. Gailis, S. Khuller, “Broadcast scheduling: Algorithms and Complexity," in *Proceeding of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 473-482, 2008.

[7] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, F. Wand, “On the competitiveness of

on-line task real-time task scheduling," *Journal of Real-Time Systems*, vol. 4, pp. 124-144, Apr. 1992.

[8] R. Lipton, A. Tomkins, "Online interval scheduling," in *Proceeding of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 302-311, 1994.

[9] M. H. Goldwasser, "Patience is a virtue; The effect of slack on competitiveness for admission control," in *Proceeding of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 396-405, 1999.

[10] S. Jiang, N. Vaidya, "Scheduling data broadcasts to impatient users," in *Proceeding of ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 52-59, 1999.

[11] B. Kalyanasundaram, K. Pruhs, "Speed is as powerful as clairvoyance," *Journal of ACM*, vol. 47, pp. 617-643, Sep. 2000.

[12] M. Brehob, E. Torng, P. Uthaisombut, "Applying extra-resource analysis to load balancing," in *Proceeding of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 560-561, 2000.

[13] C. A. Phillips, C. Stein, E. Torng, J. Wein, "Optimal time-critical scheduling via resource augmentation," in *Proceeding of the 29th ACM Symposium on Thoery of Computing*, pp. 140-149, 1997.

김재훈(Jae-Hoon Kim)

1994 서강대학교 수학과 이학사
1996 KAIST 수학과 이학석사
2003 KAIST 전산과 공학박사
2003~ 부산외국어대학교 컴퓨터공학과 교수
※관심분야 : 알고리즘, 최적화, 스케줄링