

# 게임 서버 클러스터에서의 서버의 CPU 전력 소모 최소화를 위한 클라이언트-서버 배정 방법

김상철, 이성해

한국의국어대학교 일반대학원 컴퓨터및전자시스템공학과  
kimsa@hufs.ac.kr, liii1117@daum.net

A Method of Client-Server Assignment for Minimizing the CPU Power  
Consumption of Servers in a Game Server Cluster

Sangchul Kim, Sunghae Lee

Comp. Sci. & Eng. Major, Graduate School, Hankuk University of Foreign Studies

## 요 약

데이터 센터의 전력 소모가 크고, 그 중에 컴퓨터 서버의 전력 소모 비중이 높기 때문에, 최근 서버의 전력 절약을 위한 연구가 다양한 방향에서 진행되고 있다. 서버 컴퓨터의 구성 유닛들 중에서 특히 CPU는 주된 전력 소모 원인이다. 본 논문에서는 게임 서버 클러스터 환경에서 서버 CPU의 전력 소모를 최소화하기 위한 신규 클라이언트들을 서버에 배정하는 방법을 제안한다. 우리는 클라이언트-서버 배정 문제를 최적화 문제로 모델링하고, 시뮬레이티드 어닐링 기반 방법으로 그 해를 구한다. 우리의 방법의 특징들 중 하나는 CPU의 동작 주파수를 부하에 따라 적절히 선택하는 것으로서, 저 부하시에는 낮은 주파수를 선택해 전력 소모를 줄이게 된다. 우리의 조사에 따르면, 온라인 게임 서버의 전력 소모를 고려한 클라이언트-서버 배정에 관한 연구는 거의 없었다.

**Keywords :** 클라이언트-서버 배정, 전력 소모, CPU 동작 주파수

## ABSTRACT

Since the power consumption of data centers is large and computer servers take a large portion of it, there have been much research on the power saving of servers in various ways recently. Among the units of servers CPU is one of major power consuming units. In this paper, a method of client-server assignment for minimizing the CPU power consumption of servers in a game server cluster is proposed. We model the client-server assignment problem as an optimization problem, and find a solution to the problem using a simulated annealing-based technique. One of major features of our method is to select a proper operating frequency according to the amount of load on a server. The selection of a lower frequency in case of low load will result in reducing power consumption. To our survey, little research on client-server assignment in consideration of power consumption has been carried out.

**Keywords :** Client-Server Assignment, Power Consumption, CPU Operating Frequency

Received: Jul. 18, 2017

Revised: Aug. 14, 2017

Accepted: Aug. 20, 2017

Corresponding Author: Sangchul Kim(Hankuk University of Foreign Studies)

E-mail: kimsa@hufs.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서 론

많은 플레이어들이 동시 접속하는 온라인 게임의 경우에는 다중 서버를 운영한다. 다중 서버를 지원하는 온라인 게임에서는 게임 플레이를 요구하는 새 플레이어들을 서버에 배정하는 작업이 필요하다. 본 논문에서는 플레이어를 ‘클라이언트’라고도 부른다. 그룹(즉, 팀) 단위로 진행되는 게임인 경우, 새로운 클라이언트는 먼저 로비 서버에 접속해 팀이 구성될 때까지 일정시간 대기하게 된다. 그 후, 결성된 그룹들은 서버 클러스터내 한 서버에 자동으로 배정되어 게임을 진행하게 된다. 이와 같이 신규 클라이언트들에 대해 적절한 서버를 찾아 배정하는 문제를 CSA(Client-Server Assignment) 문제라 부른다.

클라이언트-서버 배정에서는 여러 요소들을 고려하는데, 그 중 주요한 것들로 게임 서버간의 부하 균등과 클라이언트-서버 거리 균등을 들 수 있다[1,2,3,4,5]. 전자를 통해 일부 서버들에게 지나친 부담을 주는 것을 피할 수 있다. 후자는 동일한 그룹내 플레이어들간의 클라이언트-서버 거리 균등을 달성하는 것이다. 이를 통해 클라이언트들간의 사용자 반응시간을 균일하게 하여 공정한 게임 플레이를 제공할 수 있다.

최근에는 IT 기기나 서비스에서 에너지 절약을 위한 연구가 활발하다. 데이터 센터가 많은 전력을 소모하는 것으로 알려져 있고, 그 중 컴퓨터 서버들 소모하는 전력의 비중이 높다[6]. 따라서 데이터 센터내 서버 운영에서의 전력 소모 최소화화를 위한 연구가 다양한 측면에서 이루어지고 있다.

최근에는 컴퓨터 전력 절약을 위해 기본 동작 주파수 외에 그보다 작은 주파수들(성능 상태 주파수)을 동시에 지원하는 CPU가 출현했고, 이를 장착한 서버가 늘고 있다. CPU의 전력소모는 동작 속도에 크게 영향을 받는데, 속도가 느려지면 전력소모도 줄어든다[7]. 이런 CPU의 경우, 서버의 부하가 낮으면 적절히 낮은 동작수를 선택해 전력소모를 줄일 수 있을 것이다. CPU는 컴퓨터 소모

전력의 약 30% 정도를 차지하는 전력 소모 유닛이기 때문에[8], 에너지 효율적인 CPU 설계나 이용을 위한 연구가 많이 진행되고 있다.

본 논문에서는 서버 클러스터 환경에서 운영되는 온라인 게임을 대상으로, 서버들의 CPU 전력소모를 최소화하는 CSA 방법을 제안한다. 우리는 CSA 문제를 NLP(NonLinear Programming)으로 모델링하고, 그 해를 구하는 시뮬레이티드 어닐링(SA: Simulated Annealing) 기반의 방법을 제안한다. 시뮬레이티드 어닐링[9]는 유전자 알고리즘등과 함께 최적화 문제의 해를 구하는 대표적인 기술이다. 우리의 방법에는 세 개의 클라이언트-서버 배정 알고리즘이 포함된다. 하나는 전력 소모 최소화만을 목적으로 배정을 구하고, 다른 두 개는 앞에서 구한 배정에 대해 서버간 부하 균등이나 클라이언트-서버간 거리 균등을 다시 개선하게 된다. 우리는 서버들이 여러 동작 주파수를 지원하는 CPU를 사용한다고 가정한다.

우리의 CSA 방법의 주요 특징들 중 하나는 서버의 부하의 정도에 따라 적절한 CPU 동작 주파수를 선택하는 것이다. 온라인 게임에서 CPU 부하는 동시에 게임을 진행하는 플레이어들의 수에 따라 좌우된다[10]. 실험에 따르면, 우리의 방법은 CPU를 최대 주파수로 고정했을 때보다 현저히 전력소모를 줄임을 보였다. 또한 소비 전력 최소화만을 고려해 구한 클라이언트-서버 배정에 대해, 앞에서 언급한 2-3번째 알고리즘은 서버간 부하 균등이나 클라이언트-서버간 거리 균등을 개선함을 보여주었다.

서버 클러스터 환경에서의 클라이언트-서버 배정에 관한 기존 연구가 다수 발표되었다. 대부분의 연구[1,2,3,4,11]에서는 클라이언트를 서버에 자동으로 배정하면서 특정 목표 지표를 최대화 또는 최소화하도록 노력한다. 대표적인 목표 지표로서, 클라이언트-서버간의 평균 지연 시간[3,4], 주어진 클라이언트 그룹을 배정할 서버의 수[11], 서버간 부하 균등[1,2] 등이 있다. 온라인 게임에서는 특히 서버간 부하 균등에 관한 연구가 많이 발표되었다.

[1]에서는 클라이언트들을 존간에 이동시켜 또는 존의 크기를 조절해서 게임 존간의 부하 균등을 달성한다. [2]에서는 과부하된 서버가 주변 서버들에 부하의 일부를 이전시키는 분산형 방법을 제안했다. 여러 목표지수들은 서로 독립적이며, 이들 중 한 개를 최적화 대상으로 정하고, 필요한 다른 지표들은 제약 사항들(constraints)로 고려하게 된다[5].

데이터 센터의 전력 소모는 세계 에너지의 1.4%를 차지하고, 데이터 센터의 총 전력 소모의 50% 정도가 서버 등의 IT 장비가 차지한다[6]. 따라서 전력 에너지 절약을 위한 서버 운영에 관한 연구가 활발한데, 크게 세 가지 방향으로 진행되어 왔다. 하나는 작업량이 작은 때에는 CPU나 하드디스크 동작 속도를 낮추어 전력을 절약하는 것이다 [12,13]. 다른 하나는 작업들을 서버에 배정시 전력 효율이 높도록 서버 부하를 조절하는 것이다[14]. [14]에서는 서버 매핑 문제를 빈 패킹 등의 최적화 문제로 모델링한 후, 에너지 비용이 낮은 서버에 우선 배정하는 알고리즘을 제안했다. 또 다른 하나는 [15,16]과 같이 위의 두 가지를 동시에 고려하는 것이다. [15]에서는 서버의 입력 큐의 길이가 평균보다 작으면 CPU 동작 주파수를 낮추고, 동적으로 휴식상태에 빠진 서버에 작업으로 동적으로 옮겨(migration) 전력 소모 효율적으로 부하를 조절하는 방법을 제안했다.

우리가 조사한 바, 모바일 게임이나 온라인 게임을 대상으로 전력 소모 절감에 대한 연구도 진행되지만, 아직 시작 단계로 볼 수 있다. [7]에서는 모바일 게임의 프레임들의 렌더링에 소요되는 작업량을 추적해 작업량이 작으면 CPU 동작 주파수를 낮추는 방법으로 모바일 기기의 전력 절약하는 방법을 제안했다. [17]에서는 모바일 게임과 원격 서버간 패킷 전송량을 조절해 모바일 장치의 작업량을 줄여 전력 절약하는 기술을 제시했다. 모바일 게임이 실행되는 모바일 기기의 전력 절감을 위해 오프로딩(offloading) 기술이 제안되었는데, 이것은 응용 코드에서 계산량이 많은 부분은 원격지 서버

로 이전하는 것이다[18].

온라인 게임의 전력 소모에 대한 기초 연구도 발표되었다[10, 19]. [19]에서는 온라인 게임 서버의 소모 전력에 대한 이론적 모델과 이를 기반으로 최소 한계치 (low bound)를 제시했다. [10]에서는 다수사용자용 온라인 게임의 소모 전력 특성을 포함한 제반 특성을 분석했다. 이 연구에 따르면, 동접자 수에 따라 프로세서의 활용도(utilization)는 증가하는데, 그 형태가 컨벡스(convex:볼록) 함수 형태를 띤다. [20]에 따르면, CPU 활용도와 전력 소모는 선형적으로 비례한다. 따라서 온라인 게임 서버의 CPU 소모 전력은 동접자 수의 컨벡스 함수 형태로 표현될 수 있을 것이다. 실제 CPU의 소모 전력은 CPU의 전력선에서의 전압과 전류를 측정함으로써 가능하다.

기존 연구에서 보듯이, 온라인 게임 환경에서 서버의 전력 소모를 고려한 클라이언트-서버 배정 방법에 대한 발표는 거의 없었다.

## 2. 기본 개념 및 문제 모델링

### 2.1 기본 개념

본 논문에서 서버 개수는  $N$ 으로서, 서버 리스트는  $S = \{s_1, s_2, \dots, s_N\}$ 로 정의한다.

게임 서버의 부하는 접속된 클라이언트 수에 따라 결정되는데, 클라이언트 수가 증가할수록 게임 객체 관리와 클라이언트간의 동기화를 위한 작업이 늘어나게 된다. 따라서 서버가 특정 부하를 처리하기 위해 필요한 소모 전력은 해당 부하의 크기에 영향을 받는다. 부하가 커지면 서버 자원의 활용도가 늘어나서 소모 전력도 커지게 된다. 앞에서 언급한 대로, 부하가 증가하면 서버의 소모 전력은 컨벡스 함수 형태로 증가하게 된다[10].

여러 동작 주파수를 지원하는 CPU의 경우에는, 현 동작 주파수에 따라 에너지 소모가 달라진다. 동작 속도가 느려지면, CPU의 소모 전력은 줄

어든다[7]. 본 논문에서  $Watt_j^f(L)$ 은, 서버  $s_j$ 의 동작 주파수가  $f$ 일 때 부하  $L$ 을 처리한  $s_j$ 의 소모 전력을 나타낸다. 부하의 크기에 따라서 기본 동작 주파수가 아닌 보다 작은 적절한 동작 주파수를 선택하게 되면, 대부분의 경우 사용자 반응시간은 늘어난다고 CPU의 소모 전력을 줄일 수 있게 된다.

$Pow_j(L, \Delta, t)$ 는 시점  $t$ 에서부터  $\Delta$  시간 동안, 서버  $s_j$ 가 현 부하  $lj(t)$ 와 함께 새로운 부하  $L$ 을 처리하기 위해 필요한 예상 전력을 나타낸다. 앞으로 우리는  $\Delta$ 의 값으로서 클라이언트의 평균 게임 플레이 시간을 사용하게 된다.  $Pow_j(L, \Delta, t)$ 는 다음과 같이 계산할 수 있다.

$$Pow_j(L, \Delta, t) = \sum_{u=t}^{t+\Delta} Watt_j^{fn_j(l_j(u)+L)}(l_j(u) + L) \quad (\text{식 1})$$

위 식에서  $lj(u)$ 는 시점  $u$ 에 처리할  $s_j$ 의 부하를 나타낸다. 따라서  $lj(u) + L$ 는 시점  $u$ 에서 서버  $s_j$ 에  $L$ 을 추가로 배정한 후, 시점  $u$ 에서부터 게임을 진행하고 있을 예상 플레이어 수를 나타낸다.  $Watt$  함수의 위첨자인  $fn_j(L)$ 은 부하  $L$ 에 따라서 미리 지정된 동작 주파수를 나타낸다.

$fn_j(L)$ 는 서버  $s_j$ 와 게임의 특성에 따라 결정하면 될 것이다. 먼저 각 CPU 동작 주파수  $f$ 에 대해, 사용자 반응시간이 미리 설정된 상한값을 넘지 않는 최대 부하인  $\mathcal{L}_f$ 를 실험적으로 찾는다. 그러면,  $fn_j(L)$ 는  $L$ 보다 큰  $\mathcal{L}_f$ 들 중 가장 것의 동작 주파수로 결정하면 된다. 즉,  $fn_j(L)$ 는 다음과 같이 구할 수 있다:

$$fn_j(L) = \arg \min_{f \in FS, \mathcal{L}_f \geq L} \mathcal{L}_f \quad (\text{식 2})$$

위의 식에서  $FS$ 는  $s_j$ 가 지원하 모든 CPU 동작 주파수들의 집합을 나타낸다.

$Pow_j(L, \Delta, t)$ 에 서버  $s_j$ 가 설치된 데이터 센터의 단위 전력당 가격을 곱하면,  $s_j$ 의 소모 전력의 비용을 산출할 수 있다.

## 2.2 문제 모델링

앞에서 언급한대로, 서버 리스트는  $S = \{s_1, s_2, \dots, s_N\}$ 이다. 시간 인터벌  $T$  동안 새로 도착한 게임 클라이언트 수는  $L$ 이고, 이들을 서버에 배정하게 된다.  $U$ 는 클라이언트 그룹의 크기, 즉 구성원 수를 나타내는데, 플레이어  $U$ 명이 한 그룹이 되어, 서로 경쟁하면서 게임 플레이를 진행한다. 만약 바둑 게임인 경우,  $U=2$ 가 될 것이다. 클라이언트 그룹의 수는  $M=L/U$ 이고, 클라이언트 그룹 리스트는  $G=\{g_1, s_2, \dots, g_M\}$ 로 정의한다.

클라이언트 그룹  $g_i$ 가 서버  $s_j$ 에 배정되었다면  $x_{i,j} = 1$ , 아니면  $x_{i,j} = 0$ 이다. 플레이어  $c$ 와 서버  $s_j$  간의 거리를  $dist(c, s_j)$ 로 표현한다.

---

Input Parameters:  $L, t$

Find  $x_{i,j}$ 's,  $1 \leq i \leq M, 1 \leq j \leq N$  at time  $t$  such that

Minimize  $PC =$

$$\sum_{j=1}^N Pow_j(L_j, \Delta, t)$$

Subject to

$$(C1) L = \sum_{j=1}^N L_j$$

$$(C2) L_j = U * \sum_{i=1}^M x_{i,j}$$

$$(C3) DD_i = \sum_{j=1}^N \sum_{c \in g_i}$$

$$x_{i,j} |dist(c, s_j) - d_i^{avg}|/U < DD_{th}, 1 \leq i \leq M$$

$$(C4) LD = \sum_{j=1}^N$$

$$|(L_j(t) + U * \sum_{i=1}^M x_{i,j}) - L^{avg}(t)|/N < LD_{th}$$

$$(C5) \sum_{j=1}^N x_{i,j} = 1, 1 \leq i \leq M$$

$$(C6) \ x_{i,j}=1 \text{ or } 0, 1 \leq i \leq M, 1 \leq j \leq N$$

[Fig. 1] Problem Modeling in Nonlinear Programming Form

목적 함수 (objective function)에 나오는 PC는 주어진 부하 L을 N개 서버들에 배정한 후, 이들이 게임 플레이를 마칠 때까지 예상되는 총 CPU 전력 소모를 나타낸다. [Fig. 1]에 정의된 문제는 이런 전력 소모를 최소화하는 클라이언트-서버 배정을 구하는 것이다.

목적함수와 C1 및 C2에 나오는  $L_j$ 은 서버  $s_j$ 에 배정되는 클라이언트의 수이다. 제약조건 C3은 그룹  $g_i$ 에 속한 클라이언트들이 거리 편차가 큰 서버에 배정되는 것을 막기 위한 조건이다.  $DD_i$ 는 그룹  $g_i$ 에 속한 클라이언트들과 배정된 서버간의 평균 거리 편차이다. 따라서 C3은 각 그룹  $g_i$ 내 구성원들간 거리 편차가 임계값  $DD_{th}$ 보다 작아야 함을 나타낸다.  $L_i^{avg}$ 는 그룹  $g_i$ 에 속한 클라이언트들과 서버  $s_j$ 간의 평균 거리이다.

제약 조건 C4는 서버간의 부하 불균형을 피하기 위한 조건이다. 표현식  $L_j(t) + U * \sum_{i=1}^M x_{i,j}$  는  $s_j$ 의 현 부하와 새로 배정된 부하 성분의 합계를 나타낸다.  $L^{avg}(t)$ 는 서버들의 이런 합계의 평균을 나타낸다. 따라서 LD는 서버 배정 후의 서버간 평균 부하 편차로서, C4는 LD가 임계값 LD보다 작아야 함을 나타낸다.  $DD_i$ 와 LD는 응용에 따라서 적절히 정의하면 될 것이다.

제약 조건 C5는 각 그룹  $g_i$ 는 서버 한 개에만 배정되어야 됨을 나타낸다.

### 3. 시뮬레이티드 어닐링 기반의 서버 배정 알고리즘

#### 3.1 전력 최소화를 위한 서버 배정

#### Procedure *PowMin\_Alloc*

Input Parameters:  $L, t$

Step 1)  $temp = temp_{ini}$

$nv = \langle 0, \dots, 0 \rangle; ss = \{1, \dots, M\};$

$av = DoGreedyAlloc(nv, ss, t);$

$best\_v = last\_v = av;$

Step 2)

**for**  $LOOP = 1$  **to**  $LOOP\_THR$  {

Step 2-1)

choose randomly a set of  $RN$  different numbers among 1 through  $M$ , and let  $rs$  be the set.

Step 2-1)

$next\_v = Greedy\_Alloc(last\_v, rs, t);$

Step 2-3)

**if** ( $CalPow(next\_v, t) < CalPow(best\_v, t)$ )

$best\_v = next\_v;$

**if** ( $e^{(CalPow(last\_v,t)-CalPow(next\_v,t))/Temp} > random(0,1)$ )

$last\_v = next\_v;$

Step 2-4)  $Temp = CF * Temp;$

}

Step 3) output  $best\_v, f_1, \dots, f_N$

[Fig. 2] Procedure *PowMin\_Alloc*

M개 그룹으로 이루어진 신규 클라이언트들에 대해, 우리의 소모 전력을 최소화를 위한 클라이언트-서버 배정 프로시저는 [Fig. 2]에 기술되어 있다.  $nv, av, best\_v, next\_v$ 와  $last\_v$ 는 특정 클라이언트-서버 배정을 나타내는 벡터형 변수로서, 각 원소  $x_{i,j}$ 는  $g_i$ 가  $s_j$ 에 배정된 지 여부를 나타낸다.

단계 1에서는  $nv$ 의 모든 원소가 0인데, 이것은 모든 그룹에 대해 서버 배정이 이루어지지 않음을 나타낸다. *Greedy\_Alloc* 프로시저를 이용해서 초기 클라이언트-서버 배정을 구해  $av$  변수에 저장한다.

Greedy\_Alloc 프로시저는 [Fig. 3]에 기술되어 있다.

단계 2는 시뮬레이티드 어닐링 기반의 일반적인 알고리즘 형태를 가지고 있다. 단계 2-1에서 RN 개의 그룹을 선택한 후 이들을 rs 변수에 저장한다. RN 값은 실험적으로 정하면 되는데, 우리의 경우, M/2로 했다.

단계 2-2에서는 Greedy\_Alloc 프로시저를 이용해 last\_v에 대해 rs 내 그룹들의 서버 배정을 다시 수행하고, 그 결과를 next\_v 변수에 저장한다.

단계 2-3에서 CalPow(av, t) 함수는 시뮬레이티드 어닐링 알고리즘에서의 비용 함수(cost function)으로서, 시점 t에서 av에 명시된 대로 플레이어들을 서버에 배정 후 예상되는 전체 전력 소모를 계산한다. 이 함수는 다음과 같이 정의되는데, 각  $x_{i,j}$ 는 클라이언트-서버 배정 av의 원소이다:

$$CalPow(av, t) = \sum_{j=1}^N Pow_j(L_j, \Delta, t),$$

where  $L_j = U * \sum_{i=1}^M x_{i,j}$  (식 3)

단계 2-3에서는 다음 반복을 위해, 온도 파라미터인 Temp에 쿨링 계수 CF를 곱해 Temp를 갱신한다.

최종 배정인 best\_v에 따라, 서버들의 처리할 부하가 달라진다. 단계 3에서 f1, ..., fN은 '식 2'로 구한 각 서버의 CPU 동작 주파수를 나타낸다.

still allocated to  $s_i$  in AV;

$$W_j = c * U; \}$$

Step 2)

**for** idx = 1 **to** |GS| {

Step 2-1) let  $i$  be the idx-th element of GS;

Step 2-2) sort all the servers in an increasing order of  $Watt_j^{fn_j(l_j(t) + W_j)} (l_j(t) + W_j)$ . let  $os = \langle S_{o(1)}, S_{o(2)}, \dots, S_{o(N)} \rangle$  be the sorted list of the servers.

Step 2-3) find the smallest number  $j$  such that C3 and C4 of Fig 1 are satisfied after allocating group  $g_i$  to  $S_{o(j)}$ ;

Step 2-4) set  $AV_{i,o(j)} = 1$ ; set  $AV_{i,k} = 0$  for each  $k. 1 \leq k \leq o(j), o(j)+1 \leq k \leq N$ ;

Step 2-5) increment  $W_j$  by  $U$ ;

}

Step 3) output the final AV;

[Fig. 3] Procedure Greedy\_Alloc

[Fig. 3]의 Greedy\_Alloc 프로시저에서는 클라이언트-서버 배정 AV에 대해, GS에 포함된 각 그룹의 서버 배정을 새로 수행하게 된다. 단계 1에서는 먼저 GS내 그룹들의 기존 서버 배정을 취소한 후, 각 서버  $s_j$ 에 배정되어 있는 플레이어 수인  $W_j$ 를 구한다. 단계 2에서는 GS내 각 그룹  $g_i$ 에 대해,  $g_i$ 를 [Fig. 1]의 조건 C2와 C3을 만족하면서 가장 전력 소모가 작은 서버에 배정한다.

Procedure Greedy\_Alloc

Input Parameters: AV, GS, t

Step 1)

**for** each  $i \in GS$  {

undo the server allocation of group  $g_i$  in AV;

}

**for**  $j = 1$  to  $N$  {

get the number  $c$  of the groups that are

### 3.2 거리 또는 부하 편차 개선

Procedure PowLdMin\_Alloc

Input Parameters: L, t

Step 1) obtain a client-server allocation av by using PowMin\_Alloc first;

Step 2) improve av in terms of LD by

using an SA-based algorithm;

Step 3) Output the final  $av$ ;

[Fig. 4] Procedure *PowLdMin\_Alloc*

[Fig. 4]에 기술된 *PowLdMin\_Alloc* 프로시저는 *PowMin\_Alloc* 프로시저로 먼저 클라이언트-서버 배정을 구한 후, 추가로 클라이언트와 서버간의 평균 거리 편차인 LD를 개선한다.

단계 2는 [Fig. 2]의 단계 2와 비교해서 다음 두 가지 외에는 동일하다. 하나는 비용 함수인데, 본 프로시저에서는 주어진 클라이언트-서버 배정에 대해 LD를 계산하는 함수를 사용한다. 다른 하나는 Greedy\_Alloc 함수의 정의인데, 본 프로시저에서는 파라미터로 주어진 GS내의 각 그룹  $g$ 에 대해, 그룹  $g$ 를 전력 소모를 증가하지 않는 다른 서버들 중 하나에 배정한다.

우리는 *PowMin\_Alloc* 프로시저로 구한 해에 대해, 추가로 클라이언트와 서버간의 평균 거리 편차인 DD를 개선하는 *PowDdMin\_Alloc* 프로시저도 제안한다. DD는 모든 DDi들의 평균으로 정의된다. 이 프로시저의 코드는 *PowLdMin\_Alloc* 프로시저와 비슷하지만, 비용 함수로 DD를 계산하는 것을 사용하는 것이 다르다.

#### 4. 실험

우리는 CPU 전력 소모 최소화를 위한 클라이언트-서버 배정 방법의 유용성을 실험을 통해 분석했다. 실험에서 서버의 수는  $N=10$ 으로 설정했는데, 그 중 5개 서버는 ‘AMD Opteron’ CPU를, 나머지 5개 서버는 ‘Intel Pentium M’ CPU를 갖는다고 가정했다. 같은 CPU를 갖는 서버들의 성능은 동일하다고 가정했다.

전자 CPU에 대해, 부하(즉, 동시 접속자 수)에 따라 동작 주파수는 2.6, 2.4, 2.2, 2.0, 1.8GHz 중 하나로 설정되고, 각 주파수별 최대 동시 접속자

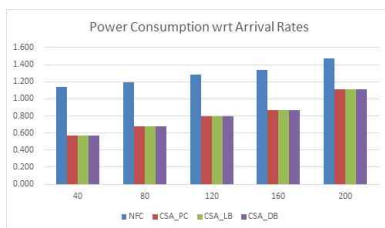
수는 동작 속도를 감안해 2000, 1850, 1700, 1550, 750명으로 정했다. 후자 CPU에 대해, 부하에 따라 1.6, 1.4, 1.2, 1.0, 0.8GHz 중 하나가 동작 주파수가 되고, 동작 주파수별 최대 동시 접속자 수는 동작 속도를 감안해 1550, 1400, 1250, 1100, 750, 600명으로 정했다. CPU 동작 주파수별로 최대 소모 전력은 [15]를 참조하면 된다.

부하  $L$ 에 대해 동작 주파수가  $f$ 로 설정될 때, CPU의 소모 전력  $Watt_j^f(L)$ 는 다음같이 정의했다[14]:  $Watt_j^f(L) = P_{idle} + \alpha_f L^2$ .  $P_{idle}$ 는 휴식 상태의 소모 전력이고,  $\tilde{L}$ 이 주파수  $f$ 에서 처리 가능한 최대 부하라면  $Watt_j^f(\tilde{L})$ 이 최대 소모 전력 값이 되도록  $\alpha_f$ 를 정의했다. 참고로 ‘AMD Opteron’과 ‘Intel Pentium M’의  $P_{idle}$ 은 각각 15 와트와 5와트이다.

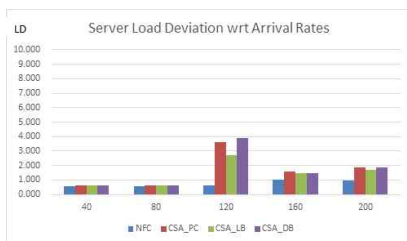
실험에 사용한 파라미터들은 게임 플레이를 원하는 사용자 도착률(user arrival rate)  $\lambda$ , 클라이언트 그룹의 크기  $U$ , 서버간 부하 편차의 한계치  $LD_{th}$ , 클라이언트-서버간 거리 편차의 한계치  $DD_{th}$ , 쿨링 계수  $CF$ , 클라이언트의 평균 플레이 타임  $\Delta$ 이다. 사용자 도착률은 단위 시간 동안에 발생하는 게임 플레이를 원하는 플레이어들의 평균 수를 나타내는데, 본 실험에서는 단위시간을 30초로 설정했다. 우리는 신규 사용자 수가 포아송 분포를 따른다고 가정했다. 편의상 클라이언트와 서버간의 거리는 1 ~ 10 사이 값을 갖고, 평균값이 5인 균일 분포(uniform distribution)를 갖는 것으로 가정했다. 우리의 실험에서는 서버의 CPU 타입별로 최대 부하를 달리 정했기에, 서버간 부하 편차 LD는 같은 CPU를 가진 서버간의 부하 편차를 먼저 구한 후, 이들의 평균으로 정의한다.

이들 파라미터에 다양한 값을 설정한 상황에서, 4시간 동안 발생하는 신규 사용자들을 서버들에 배정하는 시뮬레이션을 수행했다. 그 결과에 대해 서버들의 CPU가 소모하는 총 전력을 위주로 우리의 방법의 성능을 파악했다. 실험결과는 게임의 특성 및 부하에 대한 CPU 전력소모 함수에 따라 달라질 수 있다.

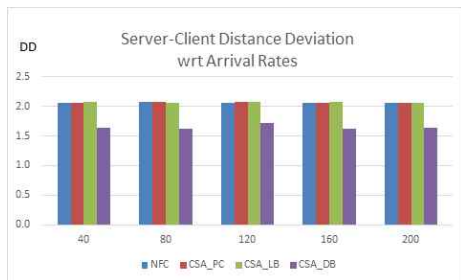
본 장의 그림에서는, CSA\_PC, CSA\_LB, CSA\_DB와 NFC라는 네 경우들의 성능을 비교해 보여준다. 앞 세 경우들은 각각 우리의 방법을 이루는 PowMin\_Alloc, PowLdMin\_Alloc와 PowDdMin\_Alloc의 결과이다. NFC는 PowMin\_Alloc 프로시저를 구한 결과이지만, 우리의 방법과는 달리 부하에 따라 서버의 CPU 주파수를 변경하지 않고 기본 동작 주파수만 고정시켰다.



(a) Power Consumption Comparison



(b) Server Load Deviation Comparison



(c) Client-Server Distance Deviation Comparison

[Fig. 5] Comparisons w.r.t. User Arrival Rates ( $U=4$ ,  $\Delta=30$ ,  $DD_{th}=4$ ,  $LD_{th}=0.5*\lambda$ ,  $CF=0.9$ )

[Fig. 5]는 파라미터  $U=4$ (명),  $\Delta=30$ (분),  $DD_{th}=4$ ,  $LD_{th}=0.5*\lambda$ ,  $CF=0.9$ 로 설정한 상태에서,

서로 다른  $\lambda$ 들에 대한 결과를 보여준다. (a)에서 보듯이, 각 경우에서  $\lambda$ 가 증가하면 소모 전력도 늘어난다. NFC와 비교해서 다른 세 경우들의 소모 전력이 현저히 낮은데, 각  $\lambda$ 별로 전력 소모 비율은 NFC 대비 약 50%, 57%, 62%, 65%, 76%이다. 이들 세 경우들의 전력 소모가 낮은 이유는 부하에 따라 적절히 낮은 CPU 동작 주파수를 선택해 소모 전력을 최소화하기 때문이다. 각 경우별로  $\lambda$ 가 증가에 따라 전력 소모 비율이 증가하는 이유는, 서버의 부하가 증가하면 CPU 동작 주파수를 높게 되어 결과적으로 소모 전력이 늘기 때문이다.

CSA\_PC 비교해서 CSA\_LB와 CSA\_DB의 전력 소모는 거의 비슷하다. 이것은 CSA\_LB와 CSA\_DB에서만 수행된 LD 또는 DD 개선 과정에서 전력 소모가 증가하게 되는 그룹 재배정은 고려하지 않기 때문이다.

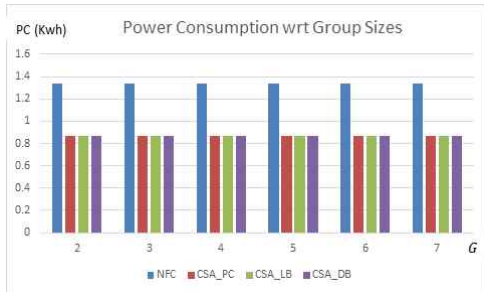
(b)에서 보듯이, 각 경우별로  $\lambda$ 가 증가함에 따라 서버간 부하 편차인 LD가 증가하는 경향을 보인다.  $\lambda < 120$ 일 때, CSA\_PC와 CSA\_LB의 LD는 거의 비슷하고,  $\lambda \geq 120$ 일 때, CSA\_LB의 LD는 CSA\_PC에 비해 약 5% 정도만 개선되었다. 이것은 CSA\_PC에서 전력 소모를 최소화하려는 노력이 결과적으로 서버간 부하 균등도 달성하게 됨을 의미한다.

또한,  $\lambda \geq 120$ 일 때, CSA\_PC, CSA\_LB 및 CSA\_DB가 NFC에 비해 LD가 크다. 즉, 이들 세 경우에는 서버간 부하 균등을 희생하는 대신 소모 전력을 줄이게 되는 특성을 보인다. 특히,  $\lambda = 120$ 일 때, 다른 도착률에 비해 LD가 큰 것을 볼 수 있다.  $\lambda$ 가 120 부근까지는 대부분 부하가 에너지 효율이 높은 서버들(즉, Intel Pentium M 기반의 서버들)에 배정된다.  $\lambda$ 가 120 부근에 가까워질수록 이런 서버의 용량에 비해 전체 부하가 커져, 전력 소모 최소화를 위한 배정을 찾는 과정에서 서버간 부하 균등이 희생되기 때문이다.  $\lambda$ 가 120을 넘어서면, 부하가 전체 서버들에 상대적으로 골고루 나누어져 LD가 줄어드는 것으로 판단된다.

(c)에서는 CSA\_PC나 CSA\_DB에서는 클라이

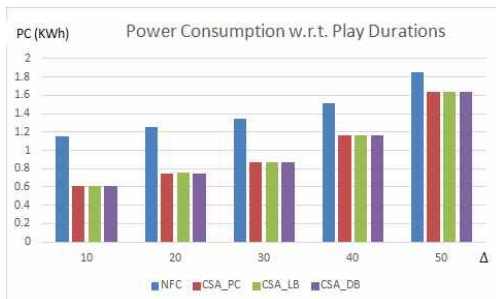


언트-서버간 거리 편차인  $DD$ 는  $\lambda$ 에 무관함을 보인다. 또한  $CSA\_DB$ 가 다른 경우들에 비해  $DD$ 가 20%~30%정도 개선되는 효과를 보였다.



[Fig. 6] Power Consumption w.r.t. Group Sizes ( $\lambda=160, \Delta=30, DDth=4, LDth=0.5*\lambda, CF=0.9,$ )

[Fig. 6]는  $CSA\_PC$ ,  $CSA\_LB$ 와  $CSA\_DB$ 의 각 경우에서, 에너지 소모가 클라이언트 그룹의 크기  $G$ 에 무관하게 거의 동일함을 보여준다. 따라서 우리의 클라이언트-서버 배정 방법이 바둑과 같이 2명이 경쟁하는 전통적인 보드 게임에서부터 보다 큰 그룹으로 플레이되는 MMOG(Massive Multi-Player Online Game)에까지 성능면에서 큰 차이를 보이지 않고 적용 가능성을 알 수 있다. 또한 모든  $G$ 에서,  $CSA\_PC$ ,  $CSA\_LB$ 와  $CSA\_DB$  간의 전력 소모가 비슷한데, 이것은  $CSA\_LB$ 와  $CSA\_DB$ 에서는 전력 소모가 증가될 수 있는 그룹 재배정을 배제하기 때문이다.

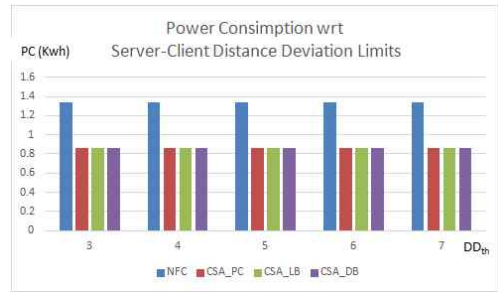


[Fig. 7] Power Consumption w.r.t. Play Durations ( $\lambda=160, U=4, DDth=4, LDth=0.5*\lambda, CF=0.9$ )

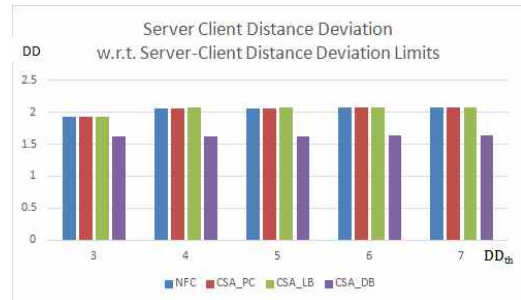
[Fig. 7]은 클라이언트들의 평균 플레이 시간인  $\Delta$

가 달라짐에 따른 전력 소모의 변화를 보여준다. 각 경우에서는  $\Delta$ 가 10분씩 커질 때마다 약 10%~14% 사이의 전력 소모 증가율을 보인다.  $\Delta$ 의 커짐은 동시에 접속하는 클라이언트의 수가 늘어나는 결과를 초래하고, 이것은 전체 부하 증가를 의미한다.

또한 그림에서 보다시피,  $\Delta$ 가 커질수록 전력 소모 증가율도 조금씩 커지는데, 이것은 [Fig. 4]에서 처럼 부하가 커지면 이를 처리하기 위해 CPU의 동작 주파수를 보다 큰 값으로 설정해 그 만큼 전력 소모가 늘기 때문이다.



(a) Power Consumption Comparison



(b) Client-Server Distance Deviation Comparison

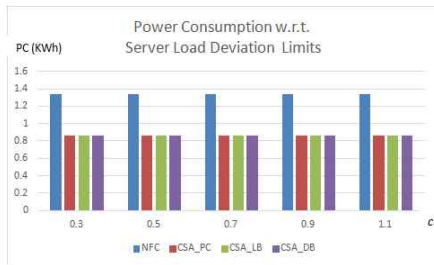
[Fig. 8] Comparisons w.r.t. Client-Server Distance Deviation Limits( $\lambda=160, U=4, \Delta=30, LDth=0.5*\lambda, CF=0.9$ )

[Fig. 8]은 클라이언트-서버 거리 편차의 한계치  $DDth$ 의 차이에 따른 전력 소모를 서로 비교한 결과이다. 여기에서는 나타나지 않지만,  $DDth \leq 2$ 와 같이  $DDth$ 가 너무 작으면, 이를 만족하는 클라이언트-서버 배정은 존재하지 않았다.

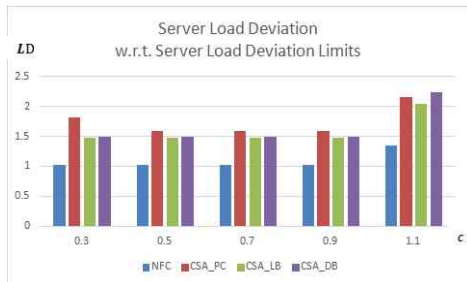
$DDth$ 가 커짐은 클라이언트-서버 배정 과정에서

선택할 수 있는 서버의 개수가 많아짐을 말하고, 결과적으로 전력 소모를 줄인 배정을 찾을 가능성이 높아짐을 의미한다. 하지만  $DD_{th} \geq 3$ 일 때 각 경우별로  $DD_{th}$ 가 커짐에 따라 전력 소모는 눈에 띌 만큼 줄지 않았다. 이것은 (b)에서 보듯이, 각  $DD_{th}$ 에서  $DD$ 가 2에 가까워,  $DD_{th}$ 에 영향을 거의 영향을 받지 않았기 때문이다. 따라서  $DD_{th}$ 가 일정한 값 (본 실험에서는 3) 이상만 되면,  $DD_{th}$ 의 증가가 전력 소모의 큰 감소를 의미하지 않음을 알 수 있다.

(b)는  $CSA\_DB$ 가 다른 방법에 비해  $DD$ 를 약 20% 정도를 개선함을 보여주는데, 이것은  $PowDdMin\_Alloc$  프로시저의 특성에 따른 당연한 결과이다.



(a) Power Consumption Comparison



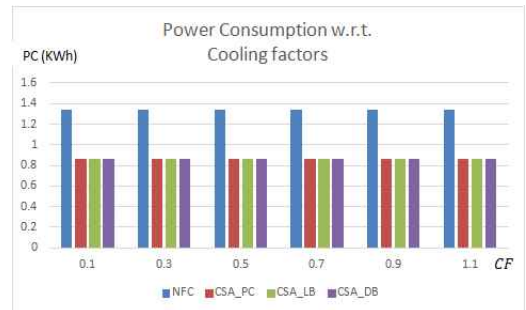
(b) Server Load Deviation Comparison

[Fig. 9] Comparisons w.r.t. Server Load Deviation Limits ( $\lambda=160$ ,  $U=4$ ,  $\Delta=30$ ,  $DD_{th}=4$ ,  $LD_{th}=c*\lambda$ ,  $CF=0.9$ )

[Fig. 9]은 서버간 부하 편차의 임계치인  $LD_{th}$ 의 차이에 따른 전력 소모를 비교한 결과이다. 각

경우별로  $LD_{th}$ 가 일정한 값 (본 실험에서는  $0.3*\lambda$ ) 이상이면 서버의  $LD_{th}$ 가 증가하더라도 전력 소모의 감소는 눈에 띌 만큼 크지 않았다.

(b)에서는 우리의 방법인  $CSA\_PC$ ,  $CSA\_DB$ 와  $CSA\_LB$ 가  $NFC$ 보다 서버간 부하 편차  $LD$ 가 큼을 보여준다. 이들이 전력 소모를 감소하기 위해  $LD$ 를 희생함을 나타낸다. 각  $LD_{th}$ 에서,  $CSA\_LB$ 가  $CSA\_PC$ 에 비해  $LD$ 를 평균 10% 정도를 개선함을 보여 준다.  $NFC$ 에 비해  $CSA\_LB$ 나  $CSA\_PC$ 의 경우,  $LD_{th}=0.3*\lambda \sim 0.9*\lambda$ 까지는  $LD$ 가 거의 비슷하지만,  $1.1*\lambda$ 에서 현저히 증가한다. 이것은  $LD_{th}$ 가 일정한 값 이상인 경우에서는(본 실험에서는  $1.1*\lambda$ ) 클라이언트-서버 배정의 결과가  $LD$ 를 희생하지만 그보다 작은  $LD_{th}$ 보다 전력 소모를 감소시키지는 못함을 알 수 있었다.



[Fig. 10] Power Consumption w.r.t. Cooling Factors ( $\lambda=160$ ,  $U=4$ ,  $\Delta=30$ ,  $DD_{th}=4$ ,  $LD_{th}=0.5*\lambda$ )

[Fig. 10]는 SA 알고리즘의 주요 파라미터인 쿨링 계수  $CF$ 가 전력 소모에 미치는 영향을 보여준다. [Fig. 2]의 단계 2에서 여러 회차에 걸쳐 반복적으로 해를 개선해 나갈 때,  $CF$ 가 작아지면 현 회차에서 구한 해가 지난 회차의 해보다 못하지만 그 해를 받아들이는 확률이 줄어든다. 즉, 최적 해를 구하기 위한 탐색 범위를 빨리 좁힘을 의미한다. 그림에서 보듯이,  $CF$ 의 변화는 우리 방법의 결과에는 큰 영향을 주지 않음을 알 수 있다.

## 6. 결 론

최근 IT 기기의 에너지 소모를 줄이려는 연구가 여러 분야에서 진행되고 있다. CPU는 서버 컴퓨터의 구성 유닛들 중에서 특히 전력 소모가 많은 유닛이다. 게임 서버 클러스터 환경에서 운영되는 온라인 게임에서는 로비 서버에 도착한 신규 클라이언트들을 서버에 적절히 배정하는 작업이 필요하다.

본 논문에서는 게임 서버 클러스터 환경에서 서버 CPU의 전력 소모를 최소화하기 위한 클라이언트-서버 배정 방법을 제안했다. 우리는 먼저 클라이언트-서버 배정을 NLP 문제로 모델링하고, 그 해를 구하는 시뮬레이티드 어닐링 기반의 알고리즘을 제안했다. 우리의 NLP 모델에는 서버 부하 균등과 클라이언트-서버간 거리 균등에 관한 제약조건들을 포함해서, 서버 배정시 소모 전력뿐만이 아니라 이들도 함께 고려하였다. 또한 전력 소모의 최소화를 목적으로 구한 클라이언트-서버 배정에 대해, 서버간 부하 편차나 클라이언트-서버간 거리 편차를 개선하는 알고리즘들도 제안했다.

우리는 서버들의 CPU가 여러 동작 주파수를 지원한다고 가정한다. 이런 CPU는 동작 주파수가 낮으면 소모 전력도 그 만큼 낮아지는 특성을 갖는다. 제안된 클라이언트-서버 배정 방법의 주된 특징은 서버 CPU의 동작 주파수를 부하에 따라 적절히 선택하는 것이다. 즉, 저 부하시에는 허용 가능한 제일 낮은 동작 주파수를 선택해 전력 소모를 줄이게 된다.

우리는 전력 소모가 두 가지 타입의 CPU를 정해, 이 중 한 CPU를 가진 서버들이 혼재된 서버 클러스터를 대상으로 제안한 방법을 실험해 보았다. 실험 결과, CPU의 동작 속도를 기본 주파수인 최대 주파수로 고정시킨 경우와 비교해, 우리의 방법이 전력소모를 현저히 줄임을 보여 주었다. 서버 소모 전력 중 CPU가 차지하는 비중이 약 30%인 점을 감안하면[8], CPU 소비 전력의 감소는 전체 서버 전력의 절약에 크게 기여한다고 볼 수 있다.

또한 전력 소모 최소화를 목적으로 구한 클라이언트-서버 배정을, 전력 소모를 늘이지 않으면서 서버간 부하 편차나 클라이언트-서버간 거리 편차면에서 개선하는 효과도 보였다.

우리가 조사한 바, 온라인 게임에서 전력 소모를 고려한 클라이언트-서버 배정에 관한 연구는 거의 없었다.

## ACKNOWLEDGMENTS

This work was supported by Hankuk University of Foreign Studies Research Fund of 2016.

## REFERENCES

- [1] C. Eduardo B. Bezerra, et. al, "Adaptive Load-balancing for MMOG Servers Using KD-trees", ACM Computers in Entertainment, Vol. 10, No. 3, Article 5, 2012, pp. 1-16.
- [2] R. W.H. Lau, "Hybrid Load Balancing for Online Games", Proc. of ACM International Conference on Multimedia, 2010, pp.1231-1234.
- [3] Y. Chen, et al, "Server Selection with Delay Constraints for Online Games." Proceedings of GLOBECOM Workshops, 2010, pp.882-887.
- [4] M. Kohana, et. al, "Dynamic ReAssignment Rules on Multi-Server Web-based MORPG System", International Journal of Grid and Utility Computing, vol. 3, no. 2/3, pp. 136-144, 2012
- [5] Sunghae Lee, Sangchul Kim, "A Method for Assigning Clients to Servers for the Minimization of Client-Server Distance Deviation", Journal of Korea Game Society, 16(3), 2016, pp.97-108.
- [6] A. Dhesikan, "Data Center Energy Efficiency: Power vs. Performance", <https://www.scribd.com/document/228651809/303>
- [7] Yan Gu, Samarjit C., "Control Theory-based DVS for Interactive 3D Games", Proc. of 45th

- DAC, 2008. pp.740-745
- [8] X. Chen, et. al, "How is energy consumed in smartphone display applications?" Proceedings of Workshop on Mobile Computing Systems and Applications, HotMobile '13. 2013, pp.3:1 - 3:6.
- [9] [https://en.wikipedia.org/wiki/Simulated\\_annealing](https://en.wikipedia.org/wiki/Simulated_annealing)
- [10] A. Abdelkhalik, et. al, "Behavior and Performance of Interactive Multi-player Game Servers", IEEE International Symposium on Performance Analysis of Systems and Software. 2001.pp.137-146
- [11] K. Lee, B. Ko, S. Calo. "Adaptive Server Selection for Large Scale Interactive Online Games." Computer Networks, Vol. 49, Issue 1, 2005, pp.84-102.
- [12] M. Weiser, B. Welch, A. Demers, and S. Shenker. "Scheduling for reduced CPU energy. Proc. of OSDI, 1994, Article No. 2
- [13] Alger, D.. "Grow a greener data center", Indianapolis, IN: Cisco Press, 2010
- [14] Dimitris Hatzopoulos, et al., "Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources", Proc. of IEEE ICC, 2013, pp.4217 - 4221
- [15] George Terzopoulos; Helen Karatza, "Power-aware load balancing in heterogeneous clusters", 2013 Proc. of SPECTS, 2013, pp.148 - 154
- [16] Yuan Yao, et al., "Data Centers Power Reduction: A two Time Scale Approach for Delay Tolerant Workloads", Proc. of IEEE INFOCOM, 2012, pp.1431-1439.
- [17] Bhojan Anand, et el., "Game action based power management for multiplayer online game", Proc. of MobiHeld, 2009, pp55-60
- [18] K. Kumar, et al., "A Survey of Computation Offloading for Mobile Systems", Mobile Netw Appl, 18, 2013, pp.129-140
- [19] Lam Ngok, "A simple bound on the game server computing power", [https://www.cse.ust.hk/~lamngok/pdf/Internet\\_gaming.pdf](https://www.cse.ust.hk/~lamngok/pdf/Internet_gaming.pdf)
- [20] Lauri Minas and Brad Ellison, "The Problem of Power Consumption in Servers", <http://www.drdoobs.com/the-problem-of-power-consumption-in-serv/215800830>, 2009.



김 상 철 (Kim, Sangchul)

1994 미시간주립대학교 컴퓨터공학과 박사  
1983-1994 ETRI 연구원  
1994 한국외국어대학교  
컴퓨터및전자시스템공학부 교수

관심분야 : 기능성게임, 게임 AI, 멀티미디어시스템

---



이 성 해 (Lee, Sunghae)

2014- 한국외국대학교 컴퓨터공학전공 석사과정  
2014- 라움 대표

관심분야: 기능성게임, 멀티미디어시스템

---