웹기반 정보시스템의 성능 향상을 위한 HW 서버 할당 방법론

황성하, 박준형, 이강수
한남대학교 컴퓨터공학과

HW Server Allocation Methodology for Improve Performance of Web-based Information System

Sung-Ha Hwang, Jun-Hyoung Park, Gang-Soo Lee
Dept. of Computer Engineering, Hannam University
E-mail : hsh0408@se.hannam.ac.kr, junside@se.hannam.ac.kr, gslee@eve.hannam.ac.kr

요 약

최근 인터넷 사용의 증가로 인해 기존의 정보시스템들이 웹기반 정보시스템으로 이전하고 있다. 이러한 웹기반 정보시스템은 사용자에게 정보를 제공하는데 있어 빠른 처리 속도와 통신 지연의 최소화로 시스템에 대한 사용자의 신뢰성을 높이는데 있다. 본 논문에서는 사용자의 서비스 만족을 위해 웹기반 정보시스템 내 SW 서버간의 결합도(coupling)를 최소화 하고 응집도(cohesion)를 최대화 하는 개념인 소프트웨어 공학과 HW 서버간의 통신량을 분석하여 웹기반 정보시스템의 성능을 향상시키기 위한 HW 서버 할당 방법론을 제안한다.

1. 서 론

최근 인터넷 사용의 폭발적인 증가는 기존의 오프라인(off-line)을 기반으로 하는 정적 환경에서 실시간으로 정보 공유가 가능한 온라인(on-line)을 기반으로 한 동적 환경으로 변화하고 있다. 그러나, 실시간으로 다수 사용자의 서비스를 처리하다보니 병렬 시스템에서 문제점으로 대두된 서버의 과부하로 서비스 처리시간이 지연되는 문제점이 발생하였다. 이는 곧 웹기반 정보시스템에 대한 사용자의 신뢰도를 떨어뜨리는 결과를 낳았다. 예컨대, 전자 쇼핑몰 사이트는 물품의 선택에서 구입까지 특정 페이지에 대한 다수 사용자의 접속 과다로 인한 접속불능이나 무한대기는 웹 쇼핑몰에 큰 타격을 준다. 이를 위해, HW 서버(이하, HS)내 SW 서버(이하, SS)간의 결합도를 최소화하고 응집도을 최대화시키는 소프트웨어 공학 개념을 접목시켜 SS를 HS에 할당하는 알고리즘이 필요하다.

HS의 성능은 서버용량, 태스크 처리시간, 워크로드(단위시간당 처리율)와 관련이 깊다[1,2]. 특히, 웹 서버의 처리시간은 정확히 파악할 수 없다. 이는 웹이 가진 특성 때문이다[3]. 이러한 웹의 특성을 해결하기 위해 대량의 정보를 효율적으로 처리하고 사용자에게 신속한 서비스를 제공하기 위한 다양한 연구가 진행되고 있다. 그러나, SS간의 링크 수에 의한 복잡도와 이로 인한 처리시간 지연 및 무한대기를 해결하기 위한 연구는 미흡하다.

*) 본 연구는 한국과학재단 목적기초연구(과제번호: R05-2001-000-01492-0) 지원으로 수행된 결과의 일부임.

이러한 배경에서, 본 연구에서는 각 SS간의 결합도 및 응집도 결과를 가지고 SS를 실제의 HS에 할당하는 알고리즘과 HS간의 통신 대역폭을 측정 및 분석하여 통신 성능을 향상시키기에 적합한 통신망을 적용함으로써 웹기반 정보시스템의 성능을 향상시키기 위한 할당 방법론을 제시하고자 한다.

본 논문은 2장에서 웹기반 정보시스템을 향상시키기 위한 기존의 방법론에 관련된 연구동향을 검토하며, 3장에서는 웹기반 정보시스템의 성능을 향상시키기 위한 HS 할당 방법론을 제시하고, 4장에서는 HS 할당 알고리즘과 통신망을 적용하여 웹기반 정보시스템을 구현하는 사례연구를 기술한다. 마지막으로 5장에서는 결과를 정리하고 앞으로의 연구방향 제시를 끝으로 결론을 맺는다.

2. 관련연구

기존의 정보시스템의 성능 향상을 위한 연구는 1970년대 큐잉이론에 기반을 둔 시스템 성능평가 방법[4]들이 연구되었으며, 최근 인터넷과 웹을 기반으로 기존의 정보시스템이 웹기반 정보시스템으로 이전함에 따른 새로운 웹기반 정보시스템에 대한 워크로드(workload) 분석, 성능평가와 프로세서에 부과되는 로드 밸런스(load balance)를 통해 서버에 프로세서를 할당하는 연구가 활발히 진행되고 있다[5~7]. 이와 관련된 연구내용을 간략히 살펴보면 다음과 같다.

2.1 분산 시스템의 성능 향상을 위한 태스크 할당 방법

분산 시스템의 성능 향상을 위해 단위 시간에 대한 처리량과 이에 대한 응답시간, 실행비용, 자원에 할당되어지는 프로세서 수를 통해 자원 사용률을 높이고, 최소의 프로세서 수, 최소 비용으로 높은 성능을 갖는 웹 서버의 구현모델을 제시하였다.[8~10]. 다음은 분산 시스템에서 중요시되는 로드 밸런스를 위해 다음과 같은 3가지 태스크 할당 방법이 제안되었다.

· 정적 태스크 할당 방법

정적 태스크 할당은 프로세서 실행전 미리 한꺼번에 각 프로세서에 태스크들을 할당하는 방법으로 각 프로세서의 균등한 태스크 분배를 그 목적으로 한다.

· 동적 태스크 할당 방법

정적 태스크 할당 방법의 단점인 전체 실행시간을 최소화하기 위해 도입된 방법으로, 전체의 프로세서 할당을 지휘하는 마스터가 존재하여 실행을 종료한 프로세서를 태스크에 할당하여 전체 실행시간을 단축시켜 웹 서버의 처리율을 높이는 방법이다.

· 혼합 태스크 할당 방법

혼합 태스크 할당은 정적 태스크 할당과 동적 태스크 할당을 혼합한 방법으로 인접한 태스크에 정적으로 프로세서를 할당하고, 동적 태스크 할당에 의한 전체 실행시간의 최소화로 로드 밸런스를 용이하게 하는 방법론이다.

2.2 웹기반 정보시스템의 재구성 휴리스틱 방법

정적 및 동적 태스크 할당을 통해 웹 서버 구조와 병목부분을 식별하여 시스템을 재구성하는 방법을 제안하였다. 시스템 재구성을 위한 방법으로 병렬처리, 병합, 분할, 격리 등과 같은 휴리스틱 방법을 제안하였다[11].

· 병렬처리(parallel processing): 사용빈도가 높은 노드는 병렬처리하거나 멀티쓰래딩함.

· 병합(merge): 사용빈도가 낮은 기능은 하나의 기능으로 통합함

· 분할(partition): 복합적인 기능을 세부기능으로 분할

· 격리(separation): 고도의 보안성을 요구하는 기능은 방화벽을 설치함.

기존의 정보시스템 성능 향상을 위한 서버 용량계획과 재할당 방법[1~5,11]은 서버의 특성 중 성능요소만을 고려하였을 뿐, 서버의 성능을 향상시키기 위한 SW 컴포넌트(또는, 모듈)간의 클러스터링과 이를 HW서버로 할당하는 체계적인 방법이 부족하다. 또한 HW 서버간의 통신량을 통신 대역폭 분석을 통한 통신망 구축을 위한 체계적인 방법을 제안하고 있지 않다.

3. HS 구현을 위한 할당 방법론

최적의 HS 구현을 위한 할당 방법론이란 유사한 기능객체들(예, 검색엔진)을 가진 기능들을 클러스터링 한 결과를 가지고 각 SS 간의 성능값과 통신량, 소프트웨어공학의 결합도와 응집도를 제약조건으로 하여 최적의 성능을 갖는 웹기반 정보시스템의 HS를 구축하기 위한 방법이다. <그림 1>은 최적의 HS를 구축하기 위한 할당 방법론을 나타낸다.

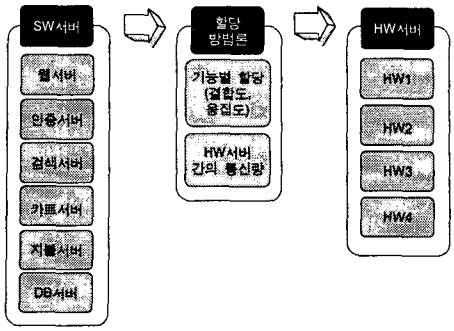

그림 1. HS 할당 방법론

SS를 HS에 할당하기 위해 소프트웨어공학의 결합도와 응집도 개념을 적용하였다. 즉, 다양한 기능을 가진 모듈들에 대해서 각기 다른 역할을 하는 모듈들의 독립성을 위해 모듈간의 결합도를 최소화하고 유사한 기능을 가진 모듈들의 응집도를 최대화하였다. 다음은 HS 할당을 위한 기본 가정이다.

[가정 1] 하나의 "HS"의 최대 처리용량은 100

[가정 2] 하나의 "HS"에 1개 이상의 SS 할당 가능

[가정 3] 결합도("스탬프 결합도")의 최소화와 응집도("순차적 응집도")의 최대화를 제약조건으로 한다.

가정 3에서 결합도를 최소화한다는 것은 기능별로 묶어 하나의 HS에 할당함을 의미하고, 응집도를 최대화하는 것은 HS의 최대 처리용량(또는, 최대 응집도)을 높이기 위해 순차적으로 입·출력이 이루어지는 기능을 하나의 서버로 할당하는 것을 의미한다. <그림 2>는 SS의 링크 구성을 나타낸다.

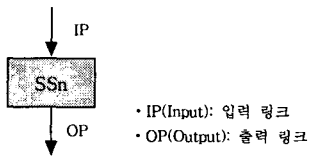

그림 2. SS의 입·출력 링크 구성

3.1 기능별 HS로 할당

<그림 3>은 SS간의 입·출력을 통한 복잡도와 처리용량을 포함하는 SS 네비게이션 다이어그램(Software Server Navigation Diagram: SSND)이다. <표 1>은 각 SS별 속성(SS 수, SS 명, 입·출력 링크 수, 처리용량, 기능)을 포함한다.

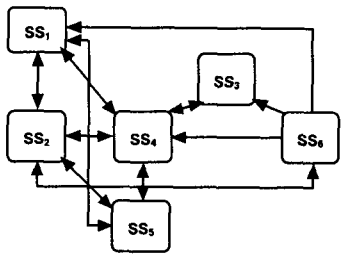

그림 3. SS 네비게이션 다이어그램(SSND)

표 1. SS별 속성 메트릭스

SS_{ID}	SS_{NAME}	IP 수	OP 수	처리용량	기능
SS_1	웹서버	4	3	95	웹
SS_2	검색서버	3	3	40	DB
SS_3	지불서버	1	0	50	인증
SS_4	인증서버	5	3	40	DB
SS_5	DB 서버	3	3	50	DB
SS_6	카트서버	1	3	90	응용

<그림 4>는 SS의 기능별로 할당한 다이어그램이다. 기능별로 할당한 HS의 결과는 다음과 같다.

$HS_1=\{SS_1\}$, $HS_2=\{SS_3\}$
$HS_3=\{SS_6\}$, $HS_4=\{SS_2, SS_4, SS_5\}$

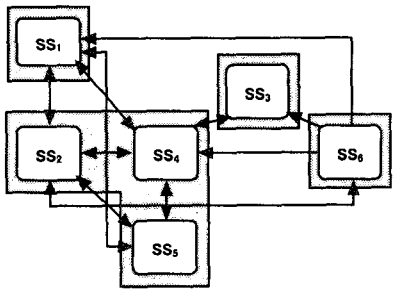

그림 4. 기능별 HS 할당

SS를 HS에 기능별로 할당하다 보니, HS의 최대 허용 처리용량(100<130)을 넘게된다. 이는 HS의 처리량의 과다로 오버헤드(overhead)가 발생하게 된다. 따라서, 이를 개선하기 위해 복잡도를 줄이고, HS의 처리량을 높이기 위해 <그림 5>와 같이 할당한다.

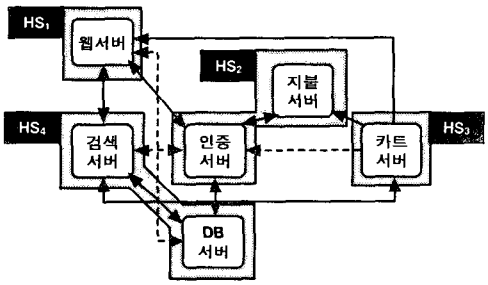

그림 5. 개선된 HS 네비게이션 다이어그램(HSND)

<그림 5>는 <그림 4>가 가지는 문제점인 HS 처리용량과 복잡도를 개선한 HS 네비게이션 다이어그램(Hardware Server Navigation Diagram: HSND)이다. 예컨대, HS_1(웹서버)와 HS_4(DB서버)간의 링크, HS_2(인증서버)와 HS_4(검색서버)간의 링크, HS_3(카트서버)에서 HS_2(인증서버)로의 링크와 같은 HS간 반복 링크를 단절함으로써, 복잡도를 줄였다.

또한, HS 응집도(SW공학의 "순차 응집도")를 최대화하기 위해 SS간의 입력이 다른 SW의 출력이 되는 SS를 한 HS에 할당하는데 단, HS의 최대 허용 처리용량을 넘지 않는 한에서 할당한다.

<표 2>는 <표 1>에 비해 SS간의 입·출력 링크 수가 줄고, 적절한 처리용량을 갖는 HS별 속성 메트릭스를 나타낸다.

표 2. 개선된 HS별 속성 메트릭스

HW_{ID}	IP 수	OP 수	처리용량	기능
HS_1	2	3	95	웹
HS_2	3	2	90	인증
HS_3	1	2	90	응용
HS_4	3	3	90	DB

3.2 HS 간의 성능 향상을 위한 통신망 구축

HS간의 최적의 성능을 갖는 통신망을 위해 HS 대화 다이어그램(Hardware Server Interaction Diagram: HSID) 모델을 이용하였다. SID 모델이란 사용자가 웹기반 정보시스템이 제공하는 서비스를 수행한 시나리오를 의미한다[3]. 즉, 시스템에 포함된 하위 개념인 기능객체에 대한 네비게이션은 표시할 수 없지만, 기능객체를 포함한 상위 개념인 HW 서버간의 네비게이션은 나타낼 수 있다.

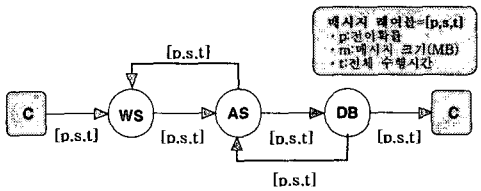

그림 6. HS 대화 다이어그램(HSID) 구조

<그림 2>는 SID의 구조를 나타낸다. HSID에서 각 서버에 전달될 메시지 레이블은 웹 브라우저가 활성화될 전이확률 p, 사용자가 입력한 메시지 크기를 나타내는 m, 마지막으로 브라우저가 활성화되어 디스플레이 할 때까지의 시간을 나타내는 전체 수행시간 t로 정의된다.

3.3 HSAA: Hardware Server Allocation Algorithm

<표 3>은 본 논문에서 제안한 HS 할당 방법을 정형화한 알고리즘을 나타낸다.

표 3. HSAA

```
//SS의 기능별 HS로 할당
MAX_HS=100          //HS의 최대 처리용량
MAX_SA=*            //최대 할당 가능한 SS 수

For i=1 to n do, For j=1 to m do//SS를 HS 할당
  Aj.SF = NULL                  //기능(SF)과 처리용량(HS) 초기화
  Aj.HS = 0
  Aj.SA = 0

  If(Aj.SF = NULL) Then         //HS에 할당
    Aj = Aj + SAi
    Aj.SF = SAi.SF
    Aj.HS = Aj.HS + SAi.HS
    Aj.SA = Aj.SA + 1; i=i+1
  Else If(Aj.HS<MAX_HS) Then//HS에 추가
    Aj = Aj + SAi
    Aj.HS = Aj.HS + FOi.HS
    Aj.SA = Aj.SA + 1; i=i+1
  Else j = j+1
  End_If End_do End_do
```

4. 사례 연구

본 논문에서는 웹기반 정보시스템 중에서도 일반적으로 대중화된 웹 쇼핑몰을 연구대상으로 했다. 위에서 언급한 HS 할당 알고

리즘을 적용한다면, 웹 사이트에 대한 근본적인 문제인 접속불능이나 처리시간 지연으로 인한 문제를 해결할 수 있으므로, 모든 종류의 웹기반 정보 시스템 튜닝시에 이 연구결과를 그대로 활용할 수 있다.

4.1 SS를 HS에 할당

유사한 기능을 수행하는 SS는 하나의 HS에 할당하여 결합도를 최소화하고 단, 최대 허용 처리용량을 넘지 않은 하에 응집도를 최대화하였다. <그림 5>는 이에 대한 결과이다.

4.2 HS간의 통신망 구축

HS간의 적합한 통신망을 구축하기 위해서는 각 서버에 접근한 총 Hit수와 접근확률, 서비스 전달될 데이터의 메시지 크기(MB) 및 전체 수행시간이 필요하다. 예컨대, 도서구입 쇼핑몰은 1시간(3,600sec)동안에 각 서버에 평균 1,000회의 Hit수를 가진다. 네트워크 통신 대역폭을 계산하기 위한 식은 다음과 같다.

$$\text{네트워크 통신 대역폭} = \frac{\text{전체 Hit수(h)} \times \text{전이확률(p)} \times \text{메시지 크기(m)}}{\text{전체 수행시간(t)}}$$

<그림 3>은 각 서버간의 네비게이션 Hit 확률, 전송될 메시지 크기 및 전체 수행시간을 나타내는 HS 대화 다이어그램(HSID)을 나타낸다. 각 서버에서 전이될 확률은 또 다른 서버로 분기되거나 사이트를 종료(Exit)하는 두 가지 전이 유형으로 구분된다. 예컨대, WS에서 AS로 전이되는 확률 0.9와 나머지 0.1의 확률은 다른 서버로 분기되거나 종료하는 확률을 의미한다.

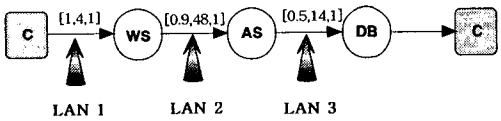

그림 3. 도서 쇼핑몰의 HSID 구조

각 HW 서버간의 통신 대역폭은 SID의 구성요소 정의에서 언급한 네트워크 통신 대역폭 공식을 통해 구할 수 있으며, 기존에 네트워크 통신망에 대한 대역폭은 이미 알려져 있으므로, 이 결과와 비교한다면 HW 서버간의 통신망을 할당하는데 그대로 활용할 수 있다. <표 4>는 HW 서버간의 통신 대역폭 결과를 통해 적용될 통신망을 나타낸다.

표 4. HW 서버간 통신 대역폭

구분 / 통신망	h	h	p	m	대역폭 (Mbps)	적 용 통신망
LAN 1 (C→WS)	1,000	3,600	1	4	8.9	Ethernet 10Mbps
LAN 2 (WS→AS)	1,000	3,600	0.9	48	96	FDDI Ring 100Mbps
LAN 3 (AS→DS)	1,000	3,600	0.5	14	15.6	Token Ring 16Mbps

결국, 각 SS의 HS로의 할당(또는 매핑) 방법과 HS간의 통신망 결과를 통해 <그림 4>와 같은 웹기반 정보시스템을 구현한다.

5. 결 론

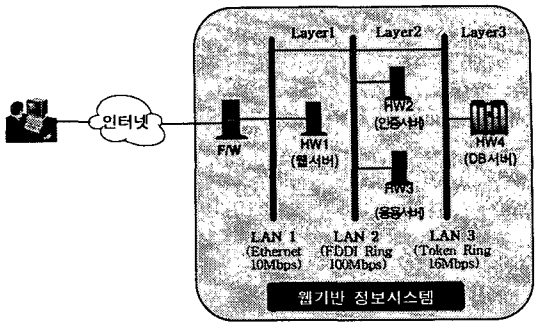

그림 4. 웹기반 정보시스템 구조

본 논문에서는 인터넷의 활성화로 인한 웹기반 정보시스템으로 변화하므로 인해 SW 기능을 포함한 SS간의 복잡도는 소프트웨어 공학의 결합도와 응집도로 해결할 수 있으며, HS간의 통신 지연으로 인한 서버 성능 저하는 통신 대역폭 측정 및 기존 통신망과의 비교를 통해 웹공학이 당면한 문제를 다소 극복할 수 있다. 특히, 웹기반 정보시스템을 구현할 때, HS 할당과 통신망 문제는 웹기반 정보시스템에 대한 사용자가 가질 수 있는 신뢰성과 밀접한 관계를 가진다. 웹기반 정보시스템은 각 SS의 결합도와 응집도를 바탕으로 웹기반 정보시스템은 각 SS에 대한 결합도와 응집도를 제약조건으로 하여 할당에 대한 타당성을 제시할 수 있고, HS간 통신성능은 HSID을 통해 분석하여 사용자 서비스 요청에 대한 신속한 처리와 시스템에 대한 신뢰성을 높일 수 있으므로, 웹기반 정보시스템에 그대로 활용할 수 있다. 최적의 웹기반 정보시스템 구현을 위한 보안성에 관한 문제는 향후 연구과제로 남긴다.

참고문헌

[1] P. Killela, *Web performance tuinging*, O'reiley, 1998.

[2] D. Menasce, *Capacity Planning for Web Performance - metric, moelis, & method*, Prentice-Hall, 1998.

[3] D. Menasce, *Scaling for E-business: Technologies, Models, Performance, and Capacity Planning*, Prentice-Hall, 2000.

[4] L. Kleinrock, *Queuing Systems*, Weley, 1975.

[5] N. Nissanke, *Realtime Systems*, Prentice-Hall, 1997.

[6] F. Arlitt, et al., "Internet Web Servers : Workload Characterization and Performance Implications," *IEEE/ ACM trans. Networking*, Vol.5, No.5, 1997.

[7] T. C. K. Chou, J. A. Abraham, "Load Balancing in Distributed Systems," *IEEE Trans. S.E*, vol. SE-8, July 1982.

[8] W. W. Chu, et al., "Task Allocation in Distributed Real-Time Systems," *Computer*, pp.57-69, Nov 1980.

[9] W. W. Chu, L. M. Lan, "Task Allocation and Precedence Relations for Distributed Real-Time Systems," *IEEE Trans. on Computers*, vol. C-36, No 6, June 1987.

[10] C. E. Houstis, "Module Allocation of Real-Time Applications to Distributed Systems," *IEEE Trans. on S.E*, Vol.16, No.7, July 1990.

[11] 박학수, 황성하, 이강수, "웹기반 정보시스템의 재구성을 위한 항해구조 및 사용자 행동 모델링", 한국멀티미디어학회 논문지(12월 게재예정), 2002.