클러스터링에서의 이중 구조 큐 순환을 이용한 부하 분산 기법에 관한 연구

구민오*, 민덕기*1
*건국대학교 컴퓨터공학부
e-mail:{happykus*,dkmin*1}@konkuk.ac.kr

A Study on Load Balancing Method using Dual Structure Queue Circulation on Clustering

Min Ku*, Dugki Min*1
*School of Computer Science and Engineering, Konkuk University

요 약

클러스터링 분야에서는 대규모 계산 및 처리, 저장 등의 작업을 하기 위해 다양한 정책과 효율적인 전략들이 기술적 사상으로써 적용되는 대표적인 분야 중 하나이다. 그 중에서도 대규모 처리(Load)시 처리자원으로의 효율적인 자원 분배(Balancing)를 하는 LoadBalancing에 관한 연구는 지속적으로 이루어지고 있다. 효율적인 로드밸런싱 기법에는 경제적인 계산 비용이 소요되는 서버풀내의 각 처리 서버들에 대한 연결 정보를 Queue에 넣고 FCFS 스케줄링 알고리즘등을 활용하는 방법들이 널리 쓰이고 있다. 그러나, 이 방법의 경우, 서버풀의 추가 및 삭제와 같은 변동상황 발생시 이를 처리하는데 많은 비용이 소요된다는 문제를 가지고 있다. 본 논문에서는 클러스터링 처리 환경에서 이중 구조 큐 순환을 이용한 서버풀 및 이에 속하는 각각의 처리서버에 대한에 부하 분산 기법을 소개하고자 한다.

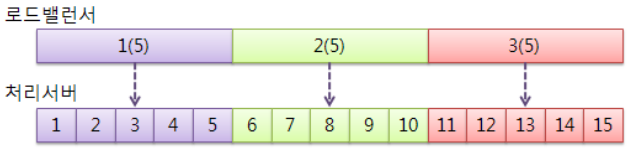
1. 서론

클러스터링은 컴퓨터의 능력을 확장 시키는 기술로써, 웹페이지 분석, 데이터 마이닝을 통한 데이터 추출, 로그 분석을 통한 다양한 경영적 자료 및 산술적 자료 산출, 대규모 연산이 필요한 생명공학분야등의 전분야에서 활용되고 있으며 시스템의 발전과 네트워크 기술의 발전과 확산은 더욱더 대규모화되는 데이터의 분석과 계산, 저장등의 목적의 과제로 인해 중요성은 더욱더 높아지고 있다. [1,2,3]

최근 클라우드 컴퓨팅 기술의 확산 영향으로 인해 대규모 처리 자원 활용에 대한 기술적, 방법적, 전략적, 비용적 중요성이 부각되면서 대규모 처리 자원을 활용하기 위한 방법으로 클러스터링 기술을 기반으로 한 국내외의 다양한 연구 및 실험, 활용등의 시도가 이루어지고 있다. [4,5,6]

대규모 클러스터를 형성할 경우, 이에 대한 고장 및 실패에 따른 많은 문제들이 야기되고 있으며, 이러한 문제들을 극복하고자 고가용성(High Availability)을 달성하고자 Heartbeat 메세지 체크 방식을 활용하거나 처리서버들 내에서 Election Algorithm[7]을 통해서 처리서버군들 중에서 고가용성 문제 해결을 시도한다.

(그림 1)은 Queue를 활용한 FCFS(First Come Fist Service) 스케줄링 기반의 시스템에서의 처리요청에 따른 자원 할당 순서에 관한 그림이다. (그림 1)에서와 같은 계산 비용이 저렴한 Queue를 활용한 FCFS(First Come Fist Service) 스케줄링 기반의 시스템에서 위와 같은 고장·실패에 따른 특성1) 서버풀내의 처리자원에 대한 정보를 제거하고자 할 경우, 제거에 대한 비용 역시 많은 양의 부하 분산을 해야하는 로드밸런서(LoadBalancer)는 주기적인 점검에 대한 자원의 할당 및 제거 및 추가에 따르는 자원 할당에 대한 비용을 지불해야 한다.[8,9]



(그림 1) FCFS 정책을 기반으로 한 Queue 방식의 자원할당순서

본 논문에서는 서버풀에 대한 정보를 개별적으로 담고 있는 서버풀 큐와 서버풀내의 처리서버의 연결정보를 담고 있는 다중 큐를 복합 적으로 활용하여 처리 요청에 맞게 각 서버풀 내의 처리서버에 작업 부하를 분산할 수 있는 부하 분산 기법을 소개한다.

---

[1] 교신저자: 민덕기 (dkmin@konkuk.ac.kr)

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음 (NIPA-2009-C1090-0902-0026)
* 본 연구는 지식경제부의 IT성장동력기술개발사업의 일환으로 수행하였음. [2008-S-007-01, 차량 전장용 통합제어 SW플랫폼 개발]

2. 이중 구조 큐 순환을 이용한 부하 분산 기법

본 논문에서 소개하는 “이중 구조 큐 순환을 이용한 부하 분산 기법”은 서버풀에 대한 큐와 서버풀 내의 처리 서버의 연결 정보에 대한 큐에 대한 이중 구조 다계층 큐(Dual Structure - Multi Layered Queue)를 복합적으로 사용함으로써, 요청된 처리규모에 따라 1차 큐와 2차 큐를 동시에 순환 시키며 처리 부하를 분산하는 기법이다.

2.1 서버풀 등록에 대한 처리

로드밸런서에 서버풀이 등록되는 시점에서 서버풀큐에 해당 서버풀에 대한 처리서버규모 등의 정보를 정보를 삽입하며, 동시에 해당 서버풀내의 각 처리 서버에 대한 연결정보를 삽입한 서버큐를 추가한다. 서버큐에 등록되는 시점에서 복수개 이상의 서버풀이 등록되는 상황이라면 다중 계층 큐 구조(Multi Layered Queue)가 형성된다.

2.2 부하분산 처리

부하 분산 서버(Load Balancer)에 처리 작업이 요청되면, 서버풀 큐에서 항목을 Pop을 하여 서버풀 정보를 획득 한 뒤, 획득된 서버풀 정보를 다시 서버풀 큐에 넣음과 동시에 서버풀 정보를 기반으로 서버큐를 선택하고 해당 서버큐에서 부하를 분산할 처리서버를 다시 획득한다.

서버풀 정보를 획득하는 단계에서 요청된 작업에서 처리서버 규모가 선택된 서버큐내의 가용 서버 규모보다 클 경우, 요청된 작업에서 요구하는 규모 만큼을 수용할 수 있는 횟수로 서버풀 큐에서 Pop과 Push를 반복적으로 수행한다.

(그림 2) 본 논문 제안 부하 분산 기법 예시

(그림 2)은 본 논문에서 소개한 이중 구조 큐 순환을 이용한 부하 분산 기법을 적용하여 처리서버를 2대 요구하였다고 가정을 하고 실행횟수를 3회 실시한 것에 대한 그림이다.

서버풀 서버에서 처리서버에 대한 처리 방식 및 관리 방식에 따라 서버풀 서버들의 연결정보가 담긴 Queue 항목들은 소멸될 수 도 있으며 유지될 수도 있다는 점은 Queue를 활용할 경우, 가장 경제적으로 활용할 수 있는 활용 방안 중 하나가 될 수 있다.

뿐만 아니라, 서버풀에 대한 연결정보를 바탕으로 작업 할당을 위해 통신시 문제가 발생할 경우 정책에 따라서 해당 서버풀 자체에 대한 배제 비용 역시 경제적으로 이루어질 수 있는 장점을 가지게 된다.

3. 결론 및 향후 계획

본 논문에서 제시한 순환형 부하 분산 알고리즘을 통해서는 대규모 처리작업시 각 서버들에게 동일한 작업을 Idle 상태 혹은 처리작업 가능 상태시 배분하는 구조로 되어 있으므로 처리서버에 일방적으로 작업을 할당하여 부하분산을 이루는 기존의 방식과는 다른 점을 보이고 있다.

특히, 일의 처리량이 동일하다고 가정을 할 경우 본 논문에서 제시한 순환형 부하 분산 알고리즘은 서버 성능 평가 자료에 필요한 비용을 지불하지 않으면서 큐에 다시 돌아오는 순서에 따라 작업을 할당하는 구조를 가지게 되므로 위와 같은 문제를 자연스럽게 해결할 수 있는 강점을 가지고 있다..

본 논문에서 제안하는 부하분산 기법은 처리작업의 단위가 일정하며 처리작업의 양에 따라 부하분산을 고려하고 있다. 향후 지속적인 연구과정을 통해 보다 범용적으로 활용될 수 있도록 처리작업의 단위의 가변성 역시 고려할 수 있도록 할 예정이다.

참고문헌

[1] O. Zamir and O. Etzioni, "Web document clustering: a feasibility demonstration", Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 98), pp 46-54, 1998.
[2] Inderjit S. Dhillon and Dharmendra S. Modha, "A Data-Clustering Algorithm On Distributed Memory Multiprocessors", Large-Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence, pages 245-260, 2000
[3] LVS(Linux Vitual Server), http://www.linuxvirtualserver.org.
[4] 김세회, 금태훈, 김건우, 이원주, 전창호, "클라우드 컴퓨팅을 위한 실험적 클러스터 구축 및 성능 분석", 한국정보과학회 학술발표 논문집, pp. 545-548, 2009
[5] 민영수, 김홍연, 김영균, "클라우드 컴퓨팅을 위한 분산 파일 시스템 기술", 정보과학회지, pp. 86~94, 2009
[6] 한국소프트웨어진흥원, "클라우드 컴퓨팅의 현재와 미래, 그리고 시장전략", 2008.
[7] HECTOR GARCIA-MOLINA, "Elections in a Distributed Computing System", IEEE TRANSACTIONS ON COMPUTERS, 1982
[8] Nitin H. Vaidya, "On Checkpoint Latency", Proceedings of the Pacific Rim International Symposium on Fault-Tolerant Systems, pp.60-65, 1995
[9] James S. Plank, "An Overview of Checkpointing in Uniprocessor and Distributed Systems, Focusing on Implementation and Performance", Technical report UT-CS-97-372, 1997