

유전자 알고리즘을 활용한 P2P 기반 동적 작업할당 관리자 설계

이승하*, 방세중*, 김양우*

*동국대학교 정보통신공학과

e-mail: lesh915@dongguk.edu, neovega@dongguk.edu, ywkim@dongguk.edu

A Design of Dynamic Job Allocation Manager based on P2P using Genetic Algorithms

Seungha Lee*, Sechung Pang*, Yangwoo Kim*

*Dept of Information and Communication, Dongguk University

요 약

P2P(Peer-to-Peer) 기반에 분산 작업 할당의 경우 정해진 수의 모든 피어에 작업을 분배하는 방식으로 처리하기 때문에 한 피어에 작업량이 증가하는 경우나 피어에 문제가 발생하는 경우 이를 사전에 예측하지 못해 시스템에 문제가 발생하게 된다. 또한 P2P 형태의 특성 상 피어의 그룹 참여는 유동적이며 그룹에 참여하는 피어의 사양도 다양하게 존재하게 된다. 이와 같은 상황에서는 작업을 할당 하더라도 작업 진행이 이루어지지 않을 확률이 높아지고 전반적으로 시스템의 성능을 저하시키는 문제점을 가지게 된다. 본 논문에서는 이러한 문제점을 해결하기 위해서 인공지능 알고리즘의 하나인 유전자 알고리즘(Genetic Algorithms: GA)를 활용하여 피어의 상태를 사전에 예측하여 작업 분배 시 동적으로 작업할 피어를 선택하여 처리하는 P2P기반 동적 작업할당 관리자를 설계하였다.

1. 서론

클러스터 기반의 작업 할당 시스템은 한정된 자원과 강결합적인 특성 때문에 작업 사용량의 증가에 따라 시스템을 능동적으로 확장하는데 어려움을 가지고 있다. 이러한 문제점을 해결하기 위해 P2P(Peer-to-Peer) 방식을 통한 분산 작업 할당 시스템이 등장하게 되었다. 하지만 이러한 P2P 기반에 분산 작업 할당[1]의 경우 정해진 수의 모든 피어에 작업을 분배하는 방식으로 처리한다. 이때, 한 피어에 작업량이 증가하면 이를 사전에 예측하지 못해 시스템 부하로 인해 작업 처리가 지연되는 문제가 발생하게 된다. 또한, P2P 형태[2-4]의 특성 상 피어의 그룹 참여는 유동적이며 그룹에 참여하는 피어의 사양도 다양하게 존재하게 된다.

시스템 관리자는 피어의 상태를 파악하지 않고 작업을 할당하여 피어의 부하를 증가시키며 최악의 경우에는 시스템이 멈추는 상황을 초래하기도 한다. 결과적으로 작업을 할당하더라도 작업 진행이 이루어지지 않을 확률이 높아지고 전반적인 시스템의 성능을 저하시키는 문제점을 가지게 된다.

본 논문에서는 이러한 문제점을 해결하기 위해서 인공지능 알고리즘의 하나인 유전자 알고리즘(Genetic Algorithms: GA)[8,9]를 활용하여 피어의 상태를 사전에 예측하여 작업 분배 시 동적으로 작업할 피어를 선택하여 처리하는 P2P기반 동적 작업할당 관리자를 제안하고자

한다.

본 논문에서 제안하고 있는 시스템은 P2P 환경을 기반으로 하고 있으며 이를 위해 SUN사에서 오픈 소스 프로젝트로 개발하고 있는 P2P 개발 플랫폼인 JXTA[5-7]를 이용하였다. 이러한 환경 하에서 작업할당 관리자가 동적으로 작업을 할당하기 위해 유전자 알고리즘을 활용하여 시스템의 상태를 파악하여 작업을 할당하도록 설계하였다.

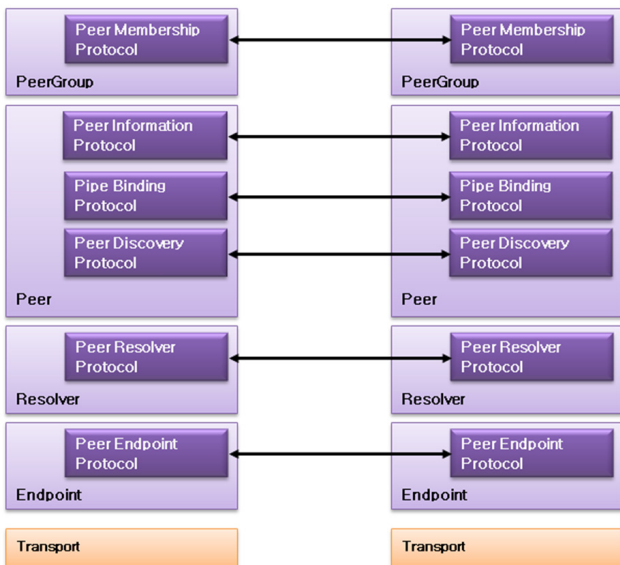
본 논문은 다음과 같이 구성된다. 2절에서 본 논문의 시스템의 기반 기술인 JXTA 플랫폼에 관해서 설명하고 3절에서 유전자 알고리즘에 관해 살펴본다. 4절에서는 본 논문에서 제안하고자 하는 동적 작업할당 관리자에 대한 설계를 살펴보고 마지막으로 5절에서 결론을 맺는다.

2 JXTA™ Platform

JXTA는 SUN Microsystems사에서 오픈소스 프로젝트의 일환으로 개발을 시작한 P2P(Peer-to-Peer) 플랫폼으로 P2P를 비롯한 분산 컴퓨팅을 위한 기반 서비스를 제공하며 P2P 컴퓨팅 환경을 위한 공개 네트워크 컴퓨팅 플랫폼[5-7]이다.

JXTA 프로토콜의 구성은 그림 1과 같이 Peer Discovery, Peer Resolver, Rendezvous, Peer Information, Pipe Binding, Endpoint Routing 등 총 6개의 프로토콜로 구성되어 있다.

- Peer Discovery Protocol : 피어로 하여금 네트워크상의 피어 서비스를 탐색할 수 있도록 다른 피어에게 광고(Advertisement)를 요청하고, 자신의 광고에 대한 다른 피어의 요청에 응답하기 위한 프로토콜
- Peer Resolver Protocol : JXTA 네트워크상에서 말단의 피어가 메시지 전파할 수 있도록 하는 프로토콜
- Rendezvous Protocol : 피어 간의 메시지 전파를 통해 다른 피어 핸들러에게 일반형의 질의를 보내고, 질의에 대한 일반형의 응답을 처리하는 프로토콜
- Peer Information Protocol : 피어에게 네트워크상의 다른 피어의 상태 정보를 입수하는 방법을 제공하는 프로토콜
- Pipe Binding Protocol : 피어 엔드포인트(Peer Endpoint)에 대하여 가상의 통신 채널을 연동하는 메커니즘을 제공하는 프로토콜. 가상의 통신채널을 파이프(Pipe)라고 함.
- Endpoint Routing Protocol : 출발지 피어에서 목적지 피어에 대한 메시지 라우팅이 가능하도록 사용되는 메시지의 집합을 제공하는 프로토콜



(그림 1) JXTA 네트워크 프로토콜

3. 유전자 알고리즘 (Genetic Algorithms:GA)

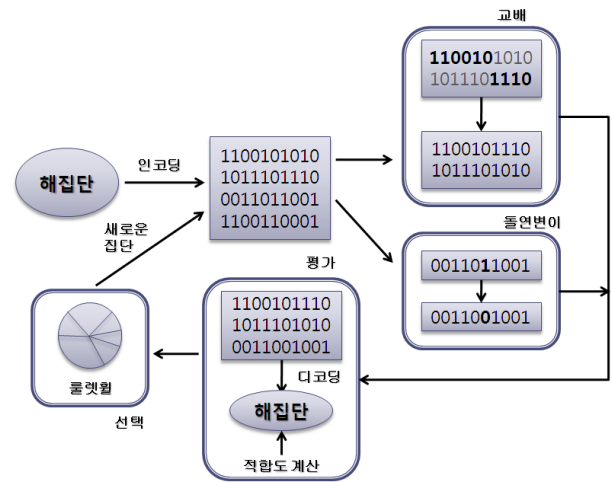
유전 알고리즘[8,9]은 진화의 원리를 문제 해결 능력에 사용하는 대표적인 알고리즘이다. 생명체의 자연 진화 법칙을 일반 최적화 문제에 적용한 방법이다. 자연 진화의 법칙에 의하면 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률을 크게 가진다는 개념을 바탕으로 두고 있다.

설계영역에 다수의 개체를 분포시켜 목적함수 값과 구속조건의 위반 정도에 따라 각 개체에 적합도(fitness)를

부여하여 적합성이 좋은 점들을 계속 활용한다. 교배와 변종을 시켜 나가면서 최적점(global optimum)을 찾아가는 것을 통해 범용성과 유용성을 높게 된다.

여러 개체들이 집단(population)을 이루어 설계디자인 영역에 대한 탐색을 행함으로써 보다 넓은 설계영역에 대한 정보를 활용할 수 있다. 전체 최적점에 수렴할 확률이 기존의 방법에 비해 상대적으로 크기 때문에 보다 최적의 값을 찾을 수 있게 된다.

유전자 알고리즘의 구성은 초기화된 집단(population)을 설정하고 선택(Select), 교배(Crossover), 돌연변이(Mutation) 과정을 진행하게 된다. 다음은 일반적인 유전자 알고리즘의 동작 순서이다.



(그림 2) 유전자 알고리즘의 동작 순서

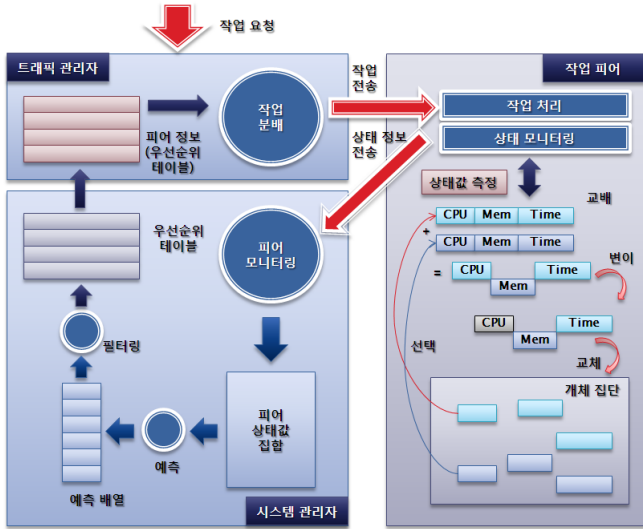
1. 선택 : 집단에서 좋은 개체를 뽑는 과정으로 일반적 방식으로는 룰렛 휠과 토너먼트 방식이 있다.
2. 교배 : 두 개체가 만나서 새로운 개체를 만드는 과정으로 두 부모 게놈을 적절히 조합하여 새로운 자손을 생성하는 연산이다. 임의의 교배점을 선택한 후, 이 위치를 기준으로 두 부모의 우측 유전자들을 교환한다.
3. 돌연변이 : 하나의 개체의 일부분을 변형하여 새로운 개체를 만드는 것으로 게놈 내의 유전자를 임의로 변경시키는 연산을 말한다.

4. 유전자 알고리즘을 활용한 P2P기반 동적 작업 할당 관리자 설계

4.1 동적 작업할당 관리자 구조

GA기반 동적 작업할당 관리자는 그림 3과 같이 크게 시스템 관리자, 트래픽 관리자, 작업 피어로 구성되어 있다. 시스템 관리자는 피어 모니터링을 통해 작업 피어의 상태 정보를 수집하게 된다. 이를 통해 시스템 관리자는 피어 상태값 집합을 구성하게 되며 이를 토대로 어떤 피

어의 상태값이 더 높은지를 필터링하여 우선순위 테이블을 구성하게 된다. 이와 같이 작성된 우선순위 테이블은 피어에 작업 분배를 위해 트래픽 관리자로 전송되며 트래픽 관리자는 작업요청이 있는 경우 우선순위 테이블의 정보에 따라 작업을 분배하게 된다.



(그림 3) 유전자 알고리즘을 활용한 P2P기반 동적 작업할당 관리자 구조도

작업 피어에서는 본 논문에서 제안하고자 하는 유전자 알고리즘을 활용한 피어 상태 예측의 기본 기능들을 수행한다. 작업 피어는 작업 처리를 수행하는 작업 처리 서비스와 함께 피어의 상태 정보를 주기적으로 모니터링하는 상태 모니터링 서비스로 구성된다.

상태 모니터링은 피어의 상태 정보를 일정한 시간에 따라 CPU 사용값과 메모리 사용량을 함께 저장하게 된다. 이 정보를 바탕으로 개체 집단을 구성하고 앞서 설명한 유전자 알고리즘의 선택, 교배, 돌연변이 연산을 수행하여 피어의 다음 상태값을 도출하게 된다. 도출된 상태값은 시스템 관리자에 전송되어 다음 작업요청 때 작업 할당의 기준이 되는 우선순위 테이블 업데이트 시 반영되게 된다.

4.2 동적 작업할당 관리자 구성 클래스

(1) SystemManger 클래스

시스템 관리자의 역할은 피어 그룹 생성, 참여, 그룹 정보 광고 발행, 작업 피어의 상태 정보 모니터링, 피어의 상태 정보에 따라 예측 판단을 수행하여 우선순위 테이블 작성 및 트래픽 관리자에 우선순위 테이블 정보 전송 등이다.

<표 1> SystemManger 메서드

메소드	내용
startJXTA()	JXTA 플랫폼 시작
startServer()	작업 피어로부터 정보수집을 위한 서버(서비스) 동작
receivePeerInfo()	작업 피어로부터 정보 수신 메서드
buildPriorityTable()	우선순위 테이블 구성
run()	작업 피어로부터 정보 수신을 위한 쓰레드 메서드
peerMonitor()	작업 피어의 상태 정보를 관리하는 메서드
calcEstimateList()	우선순위 테이블 작업 피어들의 상태값 비교 메서드

(2) TrafficManger 클래스

트래픽 관리자는 시스템 관리자로 부터 받은 우선순위 테이블 정보를 바탕으로 요청받은 작업을 큐에 저장하고 있다가 작업 피어의 상태에 따라 작업을 분배한다.

<표 2> TrafficManger 메서드

메소드	내용
TrafficManager()	트래픽 관리자 생성자 - 사용할 스레드(최대개수) 초기화
sendQueryEvent(Vector logEvent)	피어로 부터 작업 요청 쿼리의 정보를 받는다.
run()	트래픽 관리자를 스레드로 동작하기 위한 실행 메서드
sendJobEvent()	요청 받은 작업을 작업 피어에 전송하는 메서드
setPeerGroupConfigure(SystemManager mySysManager)	피어 그룹 설정 메서드

(3) WorkPeer 클래스

작업 피어는 크게 두 가지의 기능을 수행한다. 요청받은 작업을 처리하고 피어의 상태를 계속 모니터링하여 상태 정보를 시스템 관리자에게 전송하는 일을 수행한다. 피어의 상태정보는 유전자알고리즘을 이용하여 개체 생성, 선택, 교배 및 돌연변이 연산을 수행한다.

<표 3> WorkPeer 메서드

메소드	내용
startJXTA()	JXTA 플랫폼 시작
run()	작업 피어 작업 처리를 위한 쓰레드 메서드
runClient()	Pipe 정보 설정 및 Client 기능 시작
sendAndReceiveData (PipeAdvertisement pipeAdv)	요청받은 작업(Job)을 처리하기 위한 메서드
calcGA()	유전자 알고리즘 계산 메서드
selectionState()	개체 집단에서 적합한 개체를 선택하는 메서드
crossoverState()	선택된 두 개체를 일점 기준으로 교배시키는 메서드
mutationState()	하나의 개체를 임의로 값을 변경하는 메서드

5. 결론 및 향후 연구

JXTA 플랫폼은 P2P 서비스를 위한 기반 기술이고 유전자 알고리즘은 최적값을 찾으려고 하는 진화론에 기반을 둔 알고리즘이다. 일반적인 작업 할당의 경우 처음 피어의 사양을 파악하여 작업을 할당하거나 피어의 현재 상태 정보를 파악하여 작업을 할당한다. 하지만 이는 일정한 패턴을 가지는 피어의 상태를 사전에 미리 예측하는 것이 아니기 때문에 시스템에 문제가 발생하는 확률을 줄이지 못하는 문제점을 가진다.

본 논문은 위 두 기술을 사용하여 일반적인 작업 할당 방식이 아닌 피어의 상태 정보를 히스토리를 중심으로 유전자 알고리즘을 적용하여 피어의 일정 패턴을 미리 예측해 피어에 작업을 할당하는 P2P기반 동적 작업할당 관리자를 설계하였다.

본 논문에서 제안한 시스템의 경우 피어의 상태 정보를 보다 많이 확보하여 정확한 상태 패턴을 파악하는 것이 중요할 것으로 판단된다. 향후 연구로는 시스템의 상태 정보를 더 추가하고 동적 작업할당 관리자와 기존의 작업할당 관리자의 성능을 비교, 분석하여 본 시스템이 보다 안정적인 성능을 유지할 수 있도록 테스트를 진행 할 예정이다.

참고문헌

- [1] 이승하, 방세중, 김양우, 이필우, "JXTA 플랫폼 기반 분산 정보 검색 시스템 설계 및 구현에 관한 연구", 한국정보처리학회 춘계학술대회, 2008, 5.16-17
- [2] G. Flammia, "Peer-to-peer is not for everyone", IEEE Intelligent Systems [see also IEEE Expert], Volume: 16 Issue: 3, May-June 2001 Page(s): 78 -79

[3] G. Fox, "Peer-to-peer networks", Computing in Science & Engineering, Volume: 3 Issue: 3, May-June 2001 Page(s): 75 -77

[4] S. Waterhouse, D.M. Doolin, G. Kan, A. Faybishenko, "Distributed search in P2P networks", IEEE Internet Computing, Volume: 6 Issue: 1, Jan.-Feb. 2002 Page(s): 68 -72

[5] Li Gong, "JXTA: a network programming environment", IEEE Internet Computing, Volume: 5 Issue: 3, May-June 2001 Page(s): 88-95

[6] Project JXTA web site, <http://www.jxta.org>, Sun Microsystems. 2001.

[7] Project JXTA, "JXTA v2.0 Protocols Specification", Sun Microsystems 2003.

[8] J.Holland, "Adaptation in Natural and Artificial Systems", MIT Press, 1992.

[9] Z.Michalewicz, "Genetic Algorithms + Data Structures = Evolutionary Programs, 3rd edition", Springer, 1999.