

하이브리드 P2P를 위한 관심분야 기반 클러스터링

이이섭^{1†}

Interest Based Clustering Mechanism for Hybrid P2P

Lee-Sub Lee

ABSTRACT

P2P services occupy more than 50% of the internet traffics. A huge number of query packets are generated since pure P2P based models rely on message flooding for their query mechanisms. In this study, the numbers of query messages generated in the pure P2P and hybrid P2P model are analyzed. The results show that hybrid P2P models also could suffer from message flooding. To reduce the message flooding, this study proposes an interest based clustering mechanism for hybrid P2P services. By applying this clustering algorithm, it could reduce 99.998% of the message flooding. The proposed algorithm also reduces the cost of the joining operations by storing previous supernodes.

Key words : Peer to Peer, Hybrid P2P, Clustering, Data Mining, Algorithm

요 약

최근 P2P서비스는 인터넷 통신량의 50%를 넘게 차지하고 있다. 순수 P2P 기반 모델의 질의 메커니즘이 메시지 범람을 사용하기 때문에 대규모의 질의 패킷이 생성되기 때문이다. 본 연구에서는 순수 P2P 모델과 복합 P2P 모델에서 생성되는 질의 패킷의 수를 분석하였다. 그 결과 복합 P2P 모델도 메시지 범람을 발생시킨다는 것을 발견하였다. 이러한 메시지 범람을 감소시키기 위하여 본 연구에서는 복합 P2P 서비스에서의 클러스터링 메커니즘을 제안하였다. 이러한 클러스터링 알고리즘을 적용하게 되면, 약 99.998%의 메시지 범람을 감소시킬 수 있다. 제안된 알고리즘은 전에 사용되었던 슈퍼노드를 저장하므로써 조인 연산의 비용도 절감할 수 있게 하였다.

주요어 : P2P, 복합 P2P, 클러스터링, 데이터마이닝, 알고리즘

1. 서 론

최근에 P2P(Peer to Peer) 서비스의 사용이 기하급수적으로 증가하는 추세가 지속되어, 대다수의 ISP(Internet Service Provider)들은 P2P 관련 트래픽이 전체 인터넷 사용량의 50%가 넘는 것으로 보고하고 있다.^[1] 비록 네트워크의 대역폭이 기술의 발전에 따라 급속하게 증대되고 있으나, 전송되는 파일의 타입이 텍스트에서 이미지로 그리고 음악 파일과 심지어는 동영상 까지 발전하여 단위

데이터에 대한 용량이 수십KB에서 수백MB로 급격하게 증가하고 있으며, 인터넷의 보급에 따라 사용자의 숫자와 사용 빈도수도 매우 크게 증가하고 있다. 따라서 전체 사용량의 약 50%를 차지하는 P2P 서비스의 트래픽을 감소시키는 것은 인터넷 전송 서비스의 품질 향상과 비용 절감에 있어 가장 큰 이슈가 되고 있다.

P2P 서비스는 자원의 위치를 찾기 위한 질의와 찾은 자원을 피어들 간에 전송하는 2가지 단계로 진행된다. 본 연구에서는 2번째 단계인 피어들 간의 전송은 서비스를 위해서는 불가피한 단계이므로 논의에서 제외하고, 자원을 찾는 단계에서 발생하는 메시지들의 숫자를 감소시키는 방법에 대하여 연구하고자 한다.

P2P 서비스는 크게 <그림 1>과 같이 서버기반 P2P 모델(Server Based P2P Model), 순수 P2P 모델(Pure P2P Model), 그리고 하이브리드 P2P 모델(Hybrid P2P

* 본 연구는 금오공과대학교 학술연구비 의하여 연구된 논문임.

2006년 2월 28일 접수, 2006년 3월 13일 채택

¹⁾ 금오공과대학교 컴퓨터공학부

주 저 자 : 이이섭

교신저자 : 이이섭

E mail; eesub@kumoh.ac.kr

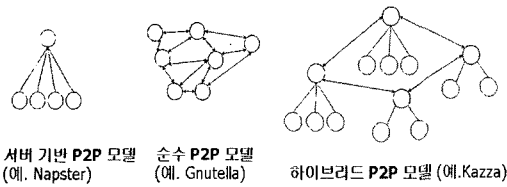


그림 1. P2P 서비스의 분류

Model)로 분류할 수 있다.^[2] 서버 기반 모델에서 각 피어는 자원의 위치를 서버에게 질의를 하고 결과를 얻게 되므로 2개의 메시지가 발생되어 질의 메시지의 발생이 최적화 되어 있다. 그러나 순수 모델에서는 중앙 서버가 존재하지 않기 때문에 질의가 인접 피어들 간의 메시지 범람(Message Flooding)으로 처리되므로 가입자의 숫자에 따라서 기하급수적으로 증가하게 된다. 예를 들어서 “grateful dead live”라는 검색어로 질의를 하게 되면 패킷의 크기는 83 bytes가 되고 각 피어들 간에 인접 피어의 개수가 8개이고 TTL(Time to live)이 8이라고 가정한다면 15,662,651개의 메시지가 생성되고, 이는 총 1,275,942,400 bytes 즉 약 1.3GB 크기 메시지가 발생된다. 이 정도의 데이터는 일반적으로 가장 큰 영화 동영상 파일에 해당이 되는 것이다.^[3]

이를 해결하기 위하여 대안으로 하이브리드 P2P 모델(Hybrid P2P Model)이 제작되어 사용되고 있다. 하이브리드 모델은 지역적으로는 서버 기반 모델을 따르며, 지역 서버 즉 슈퍼노드 간에는 순수 P2P 모델을 따르게 된다. 이 경우 슈퍼노드가 질의를 처리 할 수 있는 경우에는 서버 기반 모델과 같은 숫자의 메시지를 생성 시키지만, 그렇지 못한 경우에는 질의 처리를 위하여 슈퍼노드간의 메시지 범람을 사용해야하는 문제를 갖게 된다. 이에 대한 해결책으로서 본 연구에서는 데이터 마이닝에서 사용하는 클러스터링 기법을 P2P 서비스에서 메시지 범람을 감소시키는 방법을 제안하였다.

2장에서는 지금까지 제안되고 사용되고 있는 다양한 P2P 서비스와 모델들의 특징을 요약 비교 분석하였고, 2장에서는 메시지 범람 현상을 정량적으로 분석하여 클러스터링 방법의 타당성에 대한 근거를 제시하였다. 4장에서는 P2P 서비스에 대한 클러스터링 기법을 제안하고, 5장에서는 적용 결과에 대한 효율성을 검증하였다.

2. 관련 연구

<그림 2>은 지금까지 나타난 주요 P2P 기술을 분류한

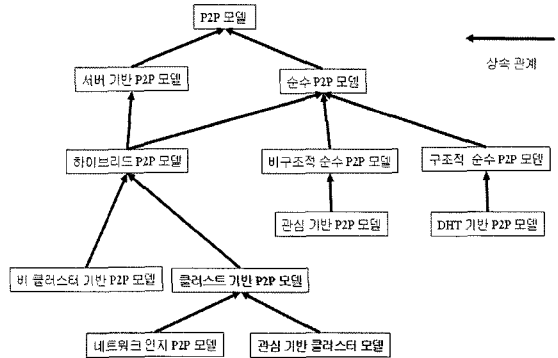


그림 2. P2P 기법의 분류

것이다. 누텔라^[3]와 같은 구조를 순수 P2P라 할 수 있다. 이미 언급한 바와 같이 순수 P2P 모델은 메시지 범람으로 인한 확장성 제한 문제가 매우 심각하다. 이 문제를 해결하기 위하여 제안된 모델이 관심 기반 P2P 모델^[6]이다. 이 모델은 가장 최근에 접근했던 자원을 보유하고 있던 피어들에게 먼저 질의를 보냄으로써 메시지 발생량을 평균적으로 5배까지 감소시키는 결과를 실험 데이터를 사용하여 확인하였다. 그러나 이러한 방법은 다음과 같은 문제가 있다.

- P2P 서비스의 피어들은 항상 접속해 있는 것이 아니라, 조인과 떠남을 반복하고 있기 때문에 시뮬레이션의 결과와는 달리 최근 접속 리스트에 의한 접속이 현실적인가에 대한 문제가 있다.
- 초기에 또는 수행 중에 지름길 목록(Shortcut List)을 찾기 위해서 발생하는 모든 피어들의 부하가 매우 크다는 문제가 있다.
- 이 모델은 순수 P2P 모델과 복합 P2P 모델 모두에게 적용 가능하지만, 순수 P2P 모델은 점차 복합 P2P 모델로 대체되고 있으며, 이 모델은 복합 P2P 모델의 장점을 활용하지 못하고 있다.

DHT(Distributed Hash Table)과 같은 종류의 자원 접근 방법을 사용하는 모델을 구조적 P2P 모델(Structured P2P Model)이라고 한다.^[6] 대표적인 예는 CAN,^[7] Pastry,^[8] Tapestry^[9] 등이 있다. 구조적 P2P는 성능이나 이론적인 완성도 면에서는 가장 우수하지만, 주로 해싱 기법을 사용하여 주소를 도출하기 때문에, 부분 검색 즉 자원의 ID를 정확하게 알지 못한다면 검색을 수행할 수 없다. 따라서 실제적으로 구축되어 사용되는 서비스가 없다.^[10]

하이브리드 P2P 모델의 대표적인 예는 Kazza^[11]나

e-Donkey이며, 가장 최근에 많이 사용하는 구조이다. 이 모델 또한 슈퍼노드간의 메시지 범람의 문제를 갖고 있다. 네트워크 인지 P2P^[12]는 물리적 네트워크 토폴로지를 기반으로 슈퍼노드를 선정하기 때문에 전통적인 기업 환경 즉 같은 지역에 존재하는 같은 부서의 사용자들 간에는 효과가 있으나, 인터넷과 같이 물리적인 위치가 사용자들의 관심 분야와 독립적인 환경에서는 의미가 없게 된다.

3. P2P모델의 질의 메시지 발생 분석

이 장에서는 순수 P2P 모델과 하이브리드 P2P 모델의 질의 메시지 발생량을 정량적으로 분석하였다.

3.1 순수 P2P 모델의 질의 메시지 발생량

순수 P2P 모델은 양방향 그래프 형태의 오버레이 네트워크를 구성한다. 따라서 하나의 질의는 그 그래프에 대한 신장 트리(Spanning Tree)를 생성하게 된다.^[3] 즉 각 피어에 대하여 하나의 메시지가 전달되므로 하나의 질의에 대한 메시지의 발생량을 계산하는 함수 f 는 사상수(Cardinality) c 와 TTL t 로 계산된다. 즉

$$f(c,t) = \sum_{k=1}^t (c-1)^{k-1} * c \tag{식(1)}$$

<그림 3>는 사상수가 3이고 t 가 4인 경우 발생하는 메시지의 개수를 보여준다.

이를 식 (1) 에 대입하면,

$$f(3,4) = \sum_{k=1}^4 (3-1)^{k-1} * 3 = 2^0 * 3 + 2^1 * 3 + 2^2 * 3 + 2^3 * 3 = 3 + 6 + 12 + 24 = 45$$

식을 정리해 보면,

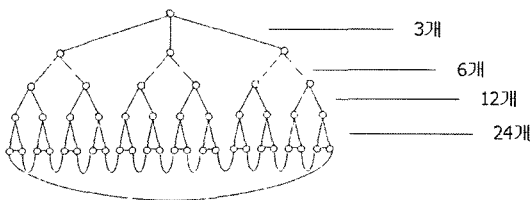


그림 3. 메시지 발생 예

$$O(f(c,t)) = O\left(\sum_{k=1}^t (c-1)^{k-1} * c\right) = O\left(\sum_{k=1}^t c^k\right)$$

이는 첫 항이 c 이고 등비가 c 인 등비수열의 합이므로

$$\left(\frac{c(c^t - 1)}{c - 1}\right) = O(c^t) \tag{식(2)}$$

식(2)는 두개의 인자를 사용하고 있으므로 각 인자를 고정시켜는 경우 전체 메시지의 발생량을 고려해보자. 우선 c 를 상수라고 가정하고 t 가 증가하는 경우 메시지의 증가량은 상수 c 에 대하여 지수적으로(Exponential) 증가하며, t 를 상수라고 가정하고 c 가 증가하게 되는 경우에는 다항식적으로(Polynomial) 증가하게 된다. 일반적으로 사상수가 고정되는 경우가 많으므로 메시지의 증가 속도는 지수적으로 증가하는 경향이 많기 때문에 더욱 문제가 된다. <표 1>에서는 이를 시각적으로 보여주기 위하여 계산 결과를 테이블로 작성하였고, <그림 4>에서 그래프로 표현하였다. 테이블을 보게 되면, $t=8$ 이고 $c=8$ 인 경우에 한 번의 질의가 700만개 이상의 메시지를 발송하게 된다.

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
c=1	1	1	1	1	1	1	1	1
c=2	2	4	6	8	10	12	14	16
c=3	3	9	21	45	93	189	381	765
c=4	4	16	52	160	484	1453	4372	13,120
c=5	5	25	105	425	1,705	6,825	27,305	109,225
c=6	6	36	186	924	4,626	22,458	117,167	555,936
c=7	7	49	301	1,813	10,885	65,317	391,908	2,351,461
c=8	8	64	456	3,200	22,408	153,864	1,098,096	7,688,400

표 1. 순수 P2P 모델의 메시지 발생량

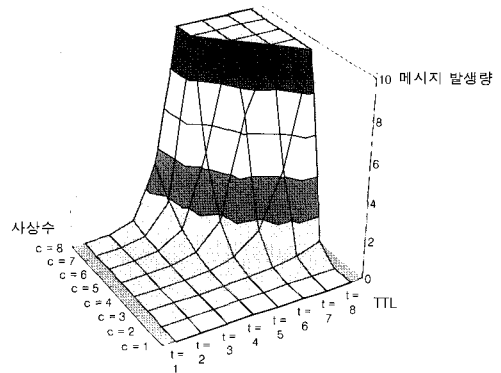


그림 4. 순수 P2P 모델의 메시지 발생량

일반적으로 사용자들은 질의를 수행시키기 위하여 P2P에 접속하게 된다. 따라서 연결되어 있는 사용자는 질의를 수행시킬 확률이 매우 높다. 만약 각 사용자가 1시간에 1회 정도의 질의 수행 시킨다면, 700만 명이 사용하는 시스템의 경우 각 피어는 시간당 700만 번의 질의를 처리해야 하는 부하를 갖게 된다. 더구나 누텔라의 경우 단순히 네트워크의 연결 상태를 점검하는 Ping과 Pong 메시지가 질의 메시지보다 평균 2배 정도 많다는 사실을 고려한다면, 각 피어가 처리해야 하는 메시지의 수는 “전체 연결된 피어의 개수” * 3 * “단위 시간당 평균 발생 질의 개수”가 된다. 따라서 순수 P2P 모델의 확장성 제한이 심각함을 알 수 있다. 따라서 순수 P2P 모델에서는 전체 검색이 현실적으로 불가능하여 TTL을 4-5로 제한하여 사용하고 있다.^[4]

3.2 하이브리드P2P 모델 메시지 발생량

하이브리드 P2P 모델의 메시지의 처리량을 분석해 보자. <그림 5>에서 보는 바와 같이, 하이브리드 P2P의 경우에는 경우에 따라서 발생하는 메시지의 개수가 매우 다양하게 나타나게 된다. 하이브리드 P2P 모델에서 각 피어는 자신의 슈퍼노드에게 질의를 요청한다. 만약 슈퍼노드가 요청된 자원의 위치를 아는 경우 즉 슈퍼노드가 관리하는 피어에 자원이 존재하는 경우에는 단지 요청과 응답의 2개의 메시지가 발생되어, 최적의 상태를 갖게 된다. 그러나 그렇지 못할 경우에는 주변 슈퍼노드들에게 메시지 범람기법을 사용하여 질의를 하게 된다. 최악의 경우 즉 모든 질의가 해당 슈퍼노드에서 처리되지 못하게 되면, 모든 피어의 질의는 슈퍼노드들 간의 오버레이 네트워크에 메시지 범람을 야기하게 된다.

슈퍼노드 간에만 순수 P2P 방식으로 연결되기 때문에 슈퍼노드들 간의 메시지 범람에서 발생하는 메시지의 개수는 같은 규모의 순수 P2P에서 발생하는 메시지의 개수의 그룹의 평균 구성원의 개수만큼 감소됨을 알 수 있

다. 이 경우 다음과 같은 문제가 발생된다.

- (가) 평균 구성원의 개수는 상수이고 메시지의 개수는 지수적으로 증가하여, 전체 규모가 커지는 경우 효과가 급격하게 상쇄된다.
- (나) 슈퍼노드들 간의 오버레이 네트워크의 메시지 발생량은 감소하지만, 각 슈퍼노드는 동일 그룹의 모든 피어들에 대한 메시지를 처리해야 하므로, 부하가 급격하게 증가하게 된다.

따라서 이를 방지하는 방법은 동일 그룹의 슈퍼노드가 처리할 수 있게 하는 확률 즉 히트율(Hit Ratio)을 높여야 한다. 히트율을 높이기 위해서는 본 연구에서는 동일한 분야의 자원에 관심을 갖는 구성원들을 모아 클러스터링시키는 방법을 연구해 보고자 한다

4. 하이브리드 P2P 모델의 클러스터링

4.1 클러스터링

데이터의 클러스터링은 사람의 도움이 없이 자동적으로 대규모 데이터를 읽어 들여 유사한 데이터들을 모은 여러 개의 클러스터를 만드는 기법이다. 이때 모아진 데이터 객체들을 클러스터라고 하는데, 동일한 클러스터 내에 존재하는 객체들은 유사성이 높고, 다른 클러스터에 존재하는 객체들과는 유사성이 낮게 된다. 이를 위해서는 클러스터링 될 데이터들의 집합이 있어야 하고, 각 데이터들은 속성 집합을 갖고 있어야 한다. 그리고 데이터 사이의 유사성을 정량적으로 측정하여 하나의 클래스 내에서는 높은 유사성을 갖게 하고, 클래스 간에는 낮은 유사성을 갖게 해야 한다.

클러스터링을 위해서는 다음과 같은 사전 작업이 필요하다. 유사성 측정을 위해서 기본적으로 각 데이터 별로 속성 값들이 저장된 테이블인 데이터 매트릭스가 있어야 한다. 그리고 데이터 매트릭스에 저장된 속성 값을 사용하여 두개의 객체 즉 객체 i 와 객체 j 간의 비유사성을 수치화한 거리함수(distance function) $d(i, j)$ 를 정의해야 한다. 거리함수의 정의는 응용분야에 따라 적합하게 작성되어야 한다. 모든 객체의 유사성을 비교하기 위해서는 모든 객체들 간의 거리함수를 계산한 비유사성 매트릭스(dissimilarity matrix)를 작성한다.

비유사성 매트릭스가 만들어지면 유사한 것들끼리 구조적으로 그룹화시키는 작업을 수행한다. 이 작업은 <그림 6>에서와 초기에 전체를 하나의 클래스로 분류하고 유사하지 않는 서브 클래스로 나누는 분할(Division) 방법과 초기에 모든 객체들을 별도의 클래스로 분류한 후 유

질의 메시지의 개수

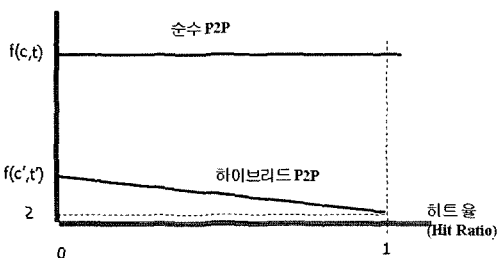


그림 5. 히트율에 따른 메시지 발생량

분할 방법 (Divisive) ← → 병합 방법 (Agglomerative)

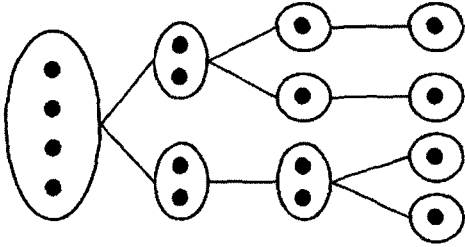


그림 6. 구조적 클러스터링 기법

사한 서버 클래스를 병합하여 상위 클래스를 형성하는 병합(Agglomeration) 방법이 사용된다.^[4]

4.2 하이브리드 P2P 모델을 위한 거리 함수

거리함수는 응용 분야에 따라 다양하게 정의된다. 본 연구에서는 각 피어간의 유사성의 판단으로 현재 피어가 가지고 있는 파일들의 중복성으로 평가하였다. 예를 들어 사용자가 비틀스의 “Yesterday”를 가지고 있다면, 다른 비틀스의 음악을 찾을 확률이 높으며, 또한 다른 비틀스의 음악을 가질 확률이 높다고 가정하였다. 이는 관심 기반 P2P의 경우와 같은 가정이다. 따라서 데이터 매트릭스는 각 피어가 가지고 있는 파일들의 이름이 된다. 피어가 가지고 있는 파일들이 매우 많은 경우에는 거리함수의 계산 시간이 길어 질 수 있으므로, 가장 최근에 저장한 파일들 중 상수 C개의 이름을 사용하게 된다.

거리 함수를 정의하기 위하여 다음과 같은 데이터 타입과 연산자를 정의하였다.

- $\langle i \rangle$ 는 객체 i의 자원들의 리스트를 생성하는 연산자이다.
- $|L|$ 은 리스트 L의 원소의 수를 계산하는 연산자이다.
- $+$ 는 리스트를 병합하는 연산자이다.
- L_1 과 L_2 가 리스트 인 경우 거리 연산자 Δ 는 다음과 같이 정의한다.

$$L_1 \Delta L_2 = \{p \mid \text{for each } p \text{ not in } L_1, p \text{ in } L_2\} + \{p \mid \text{for each } p \text{ not in } L_2, p \text{ in } L_1\}$$

객체 i와 객체 j의 거리 함수 d는 다음과 같이 정의된다. $d(i, j) = \langle i \rangle \Delta \langle j \rangle / (|\langle i \rangle| + |\langle j \rangle|)$ 예를 들어 객체 i가 파일 a, b, a, c를 저장하고 있고 객체 j가 b, b, a, d를 저장하는 경우, $\langle i \rangle = [a, b, a, c]$, $\langle j \rangle = [b, b, a, d]$ 가 된다. $\langle i \rangle \Delta \langle j \rangle = [c] + [d] = [c, d]$ 가 된다. 또한 $\langle i \rangle +$

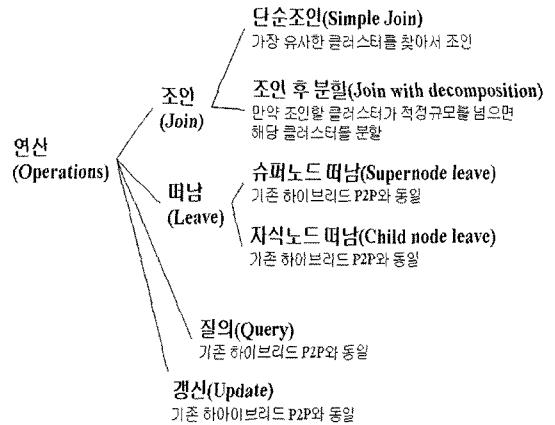


그림 7. 연산별 클러스터링의 적용

$\langle j \rangle = [a, b, a, c, b, b, a, d]$ 따라서, $s(i, j) = 2 / 8$ 가 된다. 어떤 경우에도 $|i \Delta j| > (|i| + |j|)$ 되지 않으므로 s의 값은 0에서 1의 값을 갖게 된다. 또한 한 번의 계산에 필요한 시간은 $O(|\langle i \rangle| * |\langle j \rangle|)$ 가 된다.

4.3 하이브리드 P2P에 대한 클러스터링 적용

<그림 7>은 하이브리드 P2P 방식의 연산에 대하여 클러스터링 기법 적용 방법을 요약한 것이다. 기존의 하이브리드 방식과 다른 점은 조인 연산뿐이다. 따라서 조인 연산에 대하여 자세하게 살펴보자.

<알고리즘 1>은 단순 조인 연산을 보여준다. 가장 유사한 클러스터를 찾아내는 연산은 조인 당 한 번의 메시지 범람을 발생시키므로, 기존에 소속되었던 클러스터에 우선 조인하는 것이 효율적이다. 라인 1부터 6까지는 이 과정을 기술하였다. 만약 그전에 소속되었던 슈퍼노드가 떠난 상태이면, 새로운 유사 클러스터를 찾아내야 한다. 이 과정은 라인 7부터 10까지에 기술하였다. 찾아낸 클러스터가 용량을 초과하게 되면 해당 클러스터를 유사도에 따라 분할한다. 이 과정은 라인 11과 12에서 호출된다. 저장된 슈퍼노드가 다시 조인을 할 때까지 연결 상태에 있는 경우에 이 알고리즘은 매우 효율적이다. 특히 계속 연결 상태에 있는 피어가 슈퍼노드가 되므로 이 알고리즘은 일반적으로 효율적이라고 볼 수 있다. 그러나 드문 경우이지만 만약 슈퍼노드가 떠난 상태이면, 가장 유사한 클러스터를 찾기 위하여, 모든 슈퍼노드가 유사성 계산을 수행해야 하므로 상대적으로 큰 부하를 줄 수 있다.

<알고리즘 2>는 분할 과정을 나타낸 것이다. 먼저 모든 객체를 각 독립적인 클러스터로 변환 시킨다. 총 클러

```

1. Join()
2. s = saved_supernode_of_cluster();
3. if s.online() {
4.     s.add(this);
5.     return;
6. }
7. tempSuperNode = getAnySuperNode()
8. r = getRecentSearchedResources()
9. s = tempSuperNode.findSimilarCluster(r)
10. s.updateMetadata(this)
11. if s.isExceedCapacity():
12.     s.Decomposition()
    
```

<알고리즘 1> 조인 알고리즘

```

1. Decomposition() {
2.     for each member in this cluster
3.         convert it to singular cluster;
4.     while (the number of clusters is not 2) {
5.         find a pair (ci, cj) which is highest similarity;
6.         merge(ci, cj);
7.     }
8.     elect_superNode(ci);
9.     elect_superNode(cj);
10. }
    
```

<알고리즘 2> 분할 알고리즘

스터의 개수가 2가 될 때까지 가장 유사성이 높은 클러스터를 찾아서 병합시킨다. 2개로 나누어진 각각의 클러스터에 슈퍼노드를 선정한다. 이 알고리즘의 복잡도는 클러스터의 구성원의 수를 n 이라고 하였을 때, $O(n^2)$ 가 된다. 하나의 클러스터내의 평균 구성원 수는 상수가 되고, 이러한 과정이 지역적으로 수행되므로 시스템 전체에 미치는 영향이 미미하다. 단지 조인의 빈도가 크면 부하가 증가되나, 통상적으로 조인의 빈도 보다는 질의의 빈도가 높으므로 전반적으로 메시지 범람이 크게 감소된다. 또한 이전에 소속되었던 클러스터를 저장하여, 조인하는 경우 적합한 클러스터를 찾는데 필요한 비용을 절감하였다. 따라서 메시지의 범람이 발생하는 경우는 이전에 소속되었던 슈퍼노드가 비연결 상태인 경우로 한정된다.

4.4 클러스터링 효과 분석

본 연구의 목적은 히트율을 높여 메시지 범람을 최소화 하는데 있다. 유사성 함수의 히트율은 0에서 1까지의 값을 가지게 된다. 이러한 히트율은 유사성 함수의 품질에 의하여 결정된다. 본 클러스터링 기법의 장점은 비교

표 2. 메시지 발생 증가량 표

	h= 0.01	h= 0.03	h= 0.05	h= 0.07	h= 0.09
k = 1	0.01	0.03	0.05	0.07	0.09
k = 31	0.267697	0.611023	0.796093	0.894568	0.946262
k = 61	0.458315	0.844018	0.956234	0.988047	0.996827
k = 91	0.599315	0.93745	0.990606	0.998645	0.999813
k = 121	0.703613	0.974917	0.997984	0.999846	0.999989
k = 151	0.780763	0.989942	0.999567	0.999983	0.999999
k = 181	0.83783	0.995966	0.999907	0.999998	1
k = 211	0.880043	0.998383	0.99998	1	1
k = 241	0.911268	0.999351	0.999996	1	1
k = 271	0.934365	0.999974	0.999999	1	1
k = 301	0.95145	0.999896	1	1	1

적 낮은 품질의 유사성 함수를 사용하더라도 매우 높은 히트율을 기대할 수 있다는 것이다. 하나의 피어가 유사한 피어에서 원하는 자원을 찾을 수 있는 히트율을 h 라고 하고, k 를 클러스터 내의 피어들의 평균 개수라고 한다면, 동일한 클러스터 내에서 원하는 자원을 찾을 수 있는 히트율의 기대 값 H 는 다음과 같다.

$$H = h*(1-h)^0 + h*(1-h)^1 + h*(1-h)^2 + h*((i-h)^{k-1})$$

$$= \sum_{i=0}^{k-1} h*(1-h)^i \tag{3}$$

식 (3)에 대하여 h 를 각각 1%, 3%, 5%, 7%, 그리고 9%로 가정하여 계산하고 테이블로 표현해 보면 표 2와 같다. 일반적으로 클러스터의 규모인 k 가 200에서 1000 사이이므로 $k=211$ 이라고 가정하면, h 가 0.05만 되어도 0.99998의 기대값을 기대할 수가 있다. 관심기반 P2P에서 실 데이터를 분석해본 결과 h 가 0.1이 되므로 k 가 40만 되어도 0.99998 기대값을 기대할 수 있다. 따라서 실제적으로 클러스터링 기법을 사용하게 되면 기대값이 거의 1에 가깝게 된다.

5. 결 론

본 연구에서는 순수 P2P 모델 뿐만 아니라 하이브리드 모델에서도 메시지 범람이 가장 큰 문제가 될 수 있다는 사실을 정량적으로 분석하였다. 이를 해결하기 위한 방안으로서 공통의 관심을 갖는 피어들 간의 클러스터링 기법을 제안하였다. 클러스터링을 위해서 필요한 거리함수와

주요 연산인 조인, 떠남, 질의, 갱신의 적용 방안에 대하여도 제안하였다. 또한 본 방법을 적용하였을 때, 조인에서 클러스터링에 필요한 부하는 슈퍼노드에게 $O(n^2)$ 이 된다. n 은 평균 그룹의 규모가 되어 상수가 되므로 크지 않으며, 메시지의 범람이 발생하는 경우는 이전에 소속되었던 슈퍼노드가 비연결 상태인 경우로 한정된다. 관심 기반 지름길 방법의 결과를 기준으로 클러스터링 기법을 사용할 때, 조인 빈도를 고려하지않는 경우 약 99.998%의 메시지 범람을 감소시킬 수 있다는 사실을 밝혀냈다. 향후 연구로 실제 운영 결과 데이터 기반으로 본 클러스터링 기법을 적용하였을 때 실제 메시지 범람의 감소효과를 확인할 계획이다.

참 고 문 헌

1. A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, "P2P The Gorilla in the Cable", National Cable & Telecommunications Association (NCTA) 2003 National Show, Chicago, IL, June 8-11, 2003.
2. R. Kurmanowytch, E. Kirda, C. Kerer and S. Dustdar, "OMNIX: A topology independent P2P middleware", UMICS 2003 Ubiquitous Mobile Information and Collaboration Systems Klagenfurt/Velden, Austria, June 2003
3. J. Ritter, "Why Gnutella Can't Scale, No, Really," Available from <http://www.darkridge.com/~jpr5/doc/gnutella.html>.
4. Z. Yazti, A. Folias, "A Quantitative Analysis of the Gnutella Network Traffic", 2002, <http://www.cs.ucr.edu/courses/cs204/project/gnuDC.pdf>.
5. T. Graepel, "Statistical physics of clustering algorithms", Master thesis, Technical University of Berlin, 1998.
6. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in Peer-to-Peer systems. In Proc. of the IEEE INFOCOM03), San Francisco, CA USA, 2003.
7. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network." In SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, ACM Press, 2001.
8. A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems", In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), 2001.
9. B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapstry: An infrastructure for fault-resilient wide-area location and routing," Technical Report UCB//CSD-01-1141, U.C.Berkeley.
10. C. Wang and B. Li, "Peer-to-peer overlay networks: A survey," Technical Report, Department of Computer Science, HKUST, Feb. 2003.
11. B. Thau Loo, R. Huebsch, I. Stoica, and J. M. Hellerstein, "The Case for a Hybrid P2P Search Infrastructure." IPTPS 2004: 141-150.
12. B. Krishnamurthy and J.Wang. "On Network-Aware Clustering of Web Clients." In Proceedings of ACM Sigcomm, August 2000.



이 이 섭 (eesub@kumoh.ac.kr)

1998 서강대학교 이공대학 수학과 학사
 1990 서강대학교 공과대학 전산학과 석사
 2004 고려대학교 정보통신대학 컴퓨터과 박사
 1990~1993 삼성종합기술원 주임연구원
 1994~2004 삼성SDS 정보기술연구소 수석연구원
 2004~현재 금오공과대학교 컴퓨터공학부 교수

관심분야 : Software Engineering, P2P, Workflow.