전처리 검색 기반의 P2P 그룹 검색 알고리즘

김분희

P2P Group Search Algorithm based on Preprocessing Search

Boon-hee Kim

요 약

네트워크로 연결된 환경에서 클라이언트는 서버가 보유한 파일을 요청함으로써 자원을 공유하였으나 서버의 부하로 P2P 시스템이 그 기능을 대신하고 있다. 이러한 P2P 시스템에 관한 성능 향상을 목표로 많은 연구가 진행되고 있다. 본 연구에서는 P2P 검색 시스템을 이용하는 사용자의 사용성 개선 효과를 기반으로 P2P 검색 시스템의 전처리 검색 기반 그룹 검색 알고리즘을 제안한다. 이는 사용자가 검색 용어를 검색 리스트 기반으로 전처리 과정을 거쳐 검색 속도를 높이고자 한다.

ABSTRACT

In the connected environment by network, clients shared resources as that requested the file that a server had, but P2P system is alternative because of the overload of servers. To improve the performance of this P2P system, we are studying about that. In this study, we can improve a usability for users using P2P search system, and suggest a group search algorithm based on a preprocessing search.

키워드

P2P(Peer-to-Peer), 사용성(Usability), 전처리(Preprocess), 그룹(Group)

Ⅰ. 서 론

일반적으로 네트워크 기반의 구조로 클라이언트 서버 시스템을 언급하게 된다. 그러나 서버 자체의 부하가 집중되어 서버의 성능이 무엇보다도 중요한 형태의 시스템이다. 이러한 중앙 집중식의 쏠림 현상을 극복하고 개인용 컴퓨터의 성능이 예전의 워크스테이션에 버금가는 상황과 네트워크 자체의 견고한 성장을 통해 개인이 보유한 자원을 네트워크로 연결된 많은 개인 컴퓨터의 동참을 통해 개인적으로 공유할 수 있는 P2P 시스템이 주목받게 되었다. 중앙 집중식 클라이언트 서버 시스템에 비해 P2P 시스템은 클라이언트의 유휴 자원을 이용한다는 측면에서도 전체적으로 소요된 비용 측면에서 우수하다. P2P 참여 컴퓨터의 일반적인 목적으로의 사용중에 미치는 영향은 거의 없으므로 해당 컴퓨터의 사용성 기반 성능에서 또한 장점을 가지고 있다. 자원 검색 및 다운로드 과정에서의 성능 측면에서도 기존 시스템과 유사한 특징을 가지고 있다.[1][2][3] P2P는 저가의 컴퓨터 기반 피어를 바탕으로 고가의 서버 기능을 대체할 수 있고, 유휴 시간이 많은 특징을 가지고 있는 네트워크 기반 컴퓨터들을 활용한다는 측면에서 그 유용성이 증명된 바

* 동명대학교 미디어공학과(m7515101@nate.com)
접수일자 : 2010. 09. 03 심사(수정)일자 : 2010. 09. 28 게재확정일자 : 2010. 10. 15

있다.

P2P 시스템은 피어간의 연계 방법에 따라 다양하게 분류할 수 있고, 각 모델에 따른 성능향상 방법이 다양하게 연구되고 있다. 또한 P2P 시스템에서 자원 검색, 자원 전송, 자원 전송 과정의 오류 등으로 인한 자원 재전송 등으로 높은 트래픽이 유발될 수 있어 이에 관한 성능 향상에 기존 연구의 기대치가 높았다 [4][5][6][7]. 이러한 성능향상의 다양한 방향성 가운데 사용자 입장에서의 사용성 개선 측면을 고려하고, P2P 검색 시스템의 전처리 검색 기반 그룹 검색 알고리즘을 제안한다. 이는 사용자가 검색 용어를 기반으로 시스템을 동작 시키는데 있어 검색 리스트 기반으로 전처리 과정을 거쳐 검색 속도 자체를 높여 그 사용성을 높이고자 한다.

```
Total_RSS {
Initialization : Local Variables
Definition : RSS Variables from RSS
Pre_Processor : P2P Search System Initialization
Start Search Case(Each State)
  kind = getActionCommands();
  switch (kind) {
  case search:
    Term_Generating();
    Automated_RSS();
  case peer:
    Term_Generating();
    Group_Algorithm();
    Mapping_Algorithm();
  }
End Search
Start Automated_RSS
  add_Group_Table();
  RSS_by_group();
End Automated_RSS
```

그림 1. 검색 알고리즘[4]
Fig. 1 Search algorithm[4]

II. 관련연구

본 연구의 기반 연구이면서 이전 연구로써 사용성 개선에 기여한 그룹 검색 알고리즘은 다음과 같다.“ 그림 1은 시스템을 구성하는 용어 기반 검색 알고리즘으로 RSS 기반의 검색 알고리즘으로 자원 검색 과정을 실행하고자 하는 피어의 절차와 일반 자원을 제공해주는 피어로써 P2P 검색 시스템에 참여하는 피어의 절차를 나타낸다. 자원 검색 과정을 실행하고자 하는 피어는 검색하고자 하는 용어를 PML이 정한 바에 따라 설정한 후 기존 RSS 알고리즘을 기반으로 검색 과정을 시작한다. 일반 자원을 제공해주는 피어로써 P2P 검색 시스템에 참여하는 피어는 PML 기반의 분류법에 맞춰 보유한 자원을 규정한다. 이러한 과정을 거쳐 동일한 분류 접점에 위치한 피어들은 가상의 한 그룹으로써 관리된다. 이는 자원 제공 피어가 오프라인으로 전환 시 다른 피어를 선택 시 검색의 시간이 감축된다. 피어의 수와 피어를 순환하는 시간을 같은 수라고 가정할 때, 최악의 경우인 전체 피어를 순환하는 상황에서의 시간이 n이라면 하나의 그룹을 이루는 수가 g일 때 제안한 방법은 최악의 경우 n-(n-g) 시간으로 n-g만큼의 감축효과가 있다. 가상 그룹과 실제 P2P 네트워크상의 이웃은 서로 상이한 관계로 이들 간의 매핑을 이루는 작업으로 이루어진다. 기존의 RSS 검색 알고리즘에서 검색의 기반이 되는 인덱스 테이블에 가상 그룹 관리 필드가 추가되고, 검색의 범위 또한 가상 그룹을 기반으로 검색의 범위를 한정하도록 추가하여 동작된다.

```
Term_Group_Mapping {
Initialization : PML by XML, Group_Table
Start Term_Generating
do {
      PML.select();
      search++;
     Automated_RSS();
 } while(waiting_seconds())
End
Start Group_Algorithm
do {
      PML.equals();
      add(fields);
 } while(waiting_seconds())
End
Start Mapping_Algorithm
do {
      fields.search();
      link(fields);
 } while(waiting_seconds())
End
```

그림 2. 용어 그룹 매핑 알고리즘[4]
Fig. 2 Term group mapping algorithm[4]

자원 검색 요청 피어는 먼저 PML 기반의 분류된 용어를 바탕으로 원하는 용어의 범주를 직접 선택하는 과정을 거친다. 이러한 선택과정을 거친 후 실제적으로 관련 가상 그룹 관리 모듈의 내용을 기반으로 검색 범주를 한정한다. 이렇게 검색 범위를 결정한 후 RSS기반의 수정된 Automated_RSS 알고리즘에 따라 실제적인 검색 과정을 거친다. P2P 시스템을 구성하

는 피어들을 자원 분류 기준에 따라 자체적으로 가상 그룹의 일부로 등록하는 과정을 거친다. 이러한 작업들은 모두 병렬 검색 과정을 반영한다. 가상 그룹을 기반으로 검색 결과 피어를 찾은 후 매핑과정을 거쳐 실제적인 P2P 네트워크 내의 자원 보유 피어를 포인팅 하도록 그림 2의 알고리즘이 나타낸다."[4] P2P 시스템은 자원 검색 과정과 자원 전송 작업 및 관련 오류로 인해서 자원 재전송 등과 같은 동작이 발생되고 이로 인해 네트워크 트래픽이 많이 발생되므로 이에 관한 성능 향상에 기존 연구의 초점이 맞춰져 있었는데, 성능향상의 다양한 방향성 가운데 사용자 입장에서의 사용성을 개선하기 위하여 이전에 연구했던 결과물인 RSS를 바탕으로 P2P 시스템을 이루는 피어들을 자원의 동등성에 따른 그룹으로 관리하는 환경을 설계하고 개선된 알고리즘[4] 보여주고 있다.

III. 제안 시스템

본 연구에서 제안한 알고리즘은 본 연구의 이전 연구 결과인 RSS 알고리즘과 용어 그룹 매핑 알고리즘 및 TOTAL_RSS 알고리즘을 기반으로 한다.

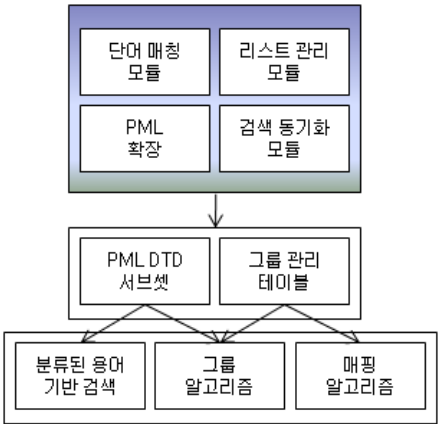

그림 3. 시스템 구성
Fig. 3 System construction

자원 검색 결과물과 사용자 요구의 일치도를 높여 사용자 입장에서의 만족도를 높이기 위해 의미가 중첩되는 검색 용어들에 대한 세부적인 분류 과정과 유사 자원을 보유한 피어들을 그룹핑 과정을 거쳐 가상의 그룹으로 관리하는 과정을 유지한다. 그림 3은 본 논문에서 제시한 검색 시스템의 구성 모듈들을 보여준다. 이전 시스템의 구성인 PML DTD 서브셋, 그룹 관리 테이블, 분류된 용어 기반 검색, 그룹 알고리즘, 매핑 알고리즘 모듈은 그대로 적용할 수 있는 객체로써 이용하며 단어매칭 모듈, 리스트 관리 모듈, PML 확장 모듈, 검색 동기화 모듈이 추가되며, PML 모듈의 경우 기존의 모듈에서 재사용 과정을 거쳐 본 시스템에 적합한 형태로 확장하는 과정이 적용된다.

```
List_Management_Algo {
Initialization : PML by XML, Group_Table
Start List_Management
do {
      tableRowManagement();
      tableColumnManagement();
      if (request)
           orderingTable();
      for(;new;){
           updateTable();
           reOrdering();
      }
      DBstore();
 } while(waiting_seconds())
End
Start Ordering_Table
do {
      analysisRequest();
      switch(request){
           case 0: oneStepOrdering();
           case 1: twoStepOrdering();
           case default: stopOrdering();
      }
 } while(waiting_seconds())
End
```

그림 4. 리스트 관리 알고리즘
Fig. 4 List management algorithm

그림 4는 사용자가 유저 인터페이스의 검색 블록에 입력한 검색 단어에 대해 타이핑하는 매 순간 완성된 단어를 예상하여 우선순위 별로 보여주고 내부적으로 최선순위 단어에 대해서는 하이딩 프로세스(hiding process)의 과정으로 처리한다. 이러한 과정에서 결과적으로 완성된 사용자 검색 단어와 하이딩 프로세스의 결과물이 같을 수도 있고, 우선순위 절차에 의해 보여주는 결과물에 대해 사용자의 선택과정을 통해 검색 단어를 완성할 수 있다. 이러한 모든 과정에서 하이딩 프로세스가 동작되고 이는 예상 단어별로 병

렬 수행됨으로써 전체 수행 효율을 높일 수 있다.

그림 5에서 리스트 관리 모듈의 결과물인 전체 시스템의 검색 단어 집합은 단어 매칭 알고리즘에서 선행 자원으로 활용된다. 리스트 관리 알고리즘의 결과물을 기반으로 수행되는 단어 매칭 알고리즘은 이벤트 프로그래밍(event programming) 기법이 적용되어 이벤트 프로그래밍의 고유한 특징인 메시지 발생에 동기화 되어 수행되는 과정을 기반으로 동작된다. 사용자가 검색 단어의 문자를 하나하나 입력할 때 마다 원래의 화면에 대해 리후레쉬 과정을 거쳐 재설정하게 되고 이러한 과정에서 발생할 수 있는 깜빡임 현상은 프로그램 차원에서 제거하여 사용자의 시각에서는 자연스럽게 변화된 검색 리스트를 볼 수 있다. 이러한 과정에서 검색 단어 문자별 매칭 포인트에 따라 선순위로 리스트업된 결과물은 순차적으로 화면상에 레이아웃 되어 선택의 기회를 높이는 효과를 보인다.

```
Term_Matching_Algo {
Initialization : result of ListManagement
Start Term_Matching
do {
  following user type:
  switch (iMsg) {
    case WM_TYPE:
            searchTable(request_term);
            break;
    case WM_PAINT:
            hdc = BeginPaint();
            for(;n;)  ListOut();
            refreshScreen();
            EndPaint();
            break;
    case WM_CREATE:  break;
    case WM_DESTROY:
            PostQuitMessage(0); break;
  }
End
Start Read_User_Type
do {
    while (GetMessage(&msg)){
       TranslateMessage(&msg);
       DispatchMessage(&msg);
    }
 } while(waiting_seconds())
End
```

그림 5. 단어 매칭 알고리즘
Fig. 5 Term matching algorithm

리스트 검색 알고리즘의 핵심은 검색의 동기화에 있다. 그림 6은 사용자가 P2P 검색 시스템을 통해 자원을 검색할 때 사용자의 타이핑하는 과정을 매순간 검색 단어 매칭 알고리즘과 연결하여 그 결과를 화면에 테이블 형식으로 반영하는 것이다. 사용자의 타이핑 과정에서 예상된 단어와 계속 달라 질 수도 있지만, 예상 했던 대로 타이핑이 이어질 수도 있는데, 이에 대해 예상과 같다면 화면에 보여지는 리스트 테이블에 대해 리프레쉬의 과정으로 반영된다. 그렇지 않을 경우에는 현재의 테이블에서 컬럼 부분을 하나 늘려 해당 컬럼에 결과물을 보여주는 역할을 한다.

```
List_Search_Algo {
Initialization : Data Result by List
Start List_Search
do {
    following user type:
    SYNC.
       user_type -> Term_Matching_Algo();
       result_table_divide(
            if (same term) same_table_refresh;
            else add_table_column;
       );
       repaint();
    End SYNC.
 } while(waiting_seconds())
End
```

그림 6. 리스트 검색 알고리즘
Fig. 6 List search algorithm

본 알고리즘의 복잡도는 문제의 크기를 나타내는 파라메터 n을 검색의 과정에서 거치게 되는 노드의 수로 정하였다. 그리고 6홉을 모두 거쳤을 경우의 n을 확인했을 때 n에 관해 가장 빨리 증가하는 항만 남기고 다른 항은 무시해도 되는 점근적인 복잡도 평가에서 이전 연구인 TGM(Term Group Mapping Search Algorithm)의 경우 $O(\log n)$ 으로 매우 낮은 복잡도를 보였는데, 본 알고리즘의 경우 첫 문자에서 바로 검색 단어와 매칭된 경우는 기존의 TGM과 복잡도가 같은 결과를 보여 알고리즘의 우수성을 보여주고 있다. 그러나 단어 매칭 포인트의 평균율에 따라 그 복잡도는 비례적으로 증가하지만 단어를 이루는 문자의 완성 개수가 K일때 최악의 단어 매칭 포인트를 기반으로 한다고 할 때 $K * O(\log n)$ 이고, 평균적으로는 중간 완성 단계에서 대부분 결정됨으로 K의 값

은 비례적으로 낮아진다. 실제 최악의 경우인 $K*O(\log n)$ 복잡도로 결정된다 하더라도 Gnutella의 $O(n^n)$ 에 비해 그 복잡도는 확연히 낮음으로 그 유효성을 확인할 수 있겠다.

Ⅵ. 실험 평가 및 결론

이 장에서는 본 논문이 제안한 사용성 개선 효과 및 검색 속도를 높이기 위한 P2P 검색 알고리즘에 대해 실험을 통해 평가하고자 한다. 실험환경은 이전 연구와의 일관성을 유지하기 위하여 Window Server 2000 운영체제 하에 JDK 개발 환경을 기반으로 소프트웨어 플랫폼을 구성했고, 시스템 사양은 Intel Pentium Ⅲ 871MHz, 하드디스크 40 Gb, 메인 메모리 256 Mb 환경에서 실험하였다. 이전의 RSS 알고리즘에서의 실험환경과 마찬가지로 Peersim 기반의 시뮬레이션 인수 및 기본값을 설정하였다. 그룹 알고리즘에서 하나의 그룹으로 묶이지 않는 단일 검색 용어는 없는 것으로 가정하며 다의어의 배정에 있어 실험 환경을 감안하여 가상의 값으로 2에서 10으로 이전 연구와 마찬가지로 동등 분포에 의해 임의적으로 적용한다. 일치하는 단어 맥락의 범위는 시뮬레이션 속성상에서 문자 개수가 증가될수록 일치 단어가 단계적으로 줄어드는 1차 함수로 적용하고 12개 이하 낱자로 한정한다.

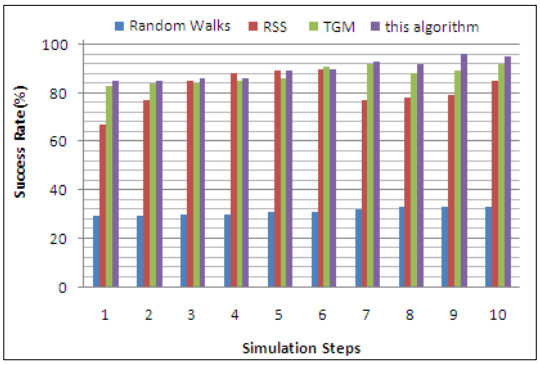

그림 7. 성공률 비교
Fig. 7 Success rate comparison

제안한 알고리즘은 이전 연구인 TGM 알고리즘을 근간으로 확장하여 TGM 고유의 문제점과 이점이 전이되어 있어 사용자 만족도 측면에서 이전연구의 실험 결과와 마찬가지로 만족할 만한 결과를 보이고 있으며, 성공률 측면에서는 그림 7의 결과와 같다. 비교대상과 검색 성공률을 실험 초기와 실험 단계가 높아 짐으로써의 변화된 측정값을 반영한 결과 기존의 방법론에 비해 월등히 높은 성공률을 보이고 있으며 TGM에 비해서도 소폭 상승한 결과를 보임과 동시에 각각의 최대값에 대한 단순 비교에서도 4%대의 의미있는 차이를 보이며 최소값 대비 최고 13%의 차이를 보이며 만족할 만한 성공률을 보이고 있다.

검색 속도 측면에서 본 알고리즘을 평가해보면 다음과 같다. 최대값 0과 최소값 1의 범위 내에서의 상대적 비율로써 측정되었으며 개인적인 타수의 편차는 없는 것으로 가정하여 실험한 결과이다. 검색 속도 실험에서의 정확도는 성공률 상에서 보여지는 결과가 반영된 상태임을 가정한 결과이다. 사용자의 검색 용어에 대한 검색 리스트 기반의 전처리 과정에서 발생하는 시간적인 요인을 배제한 결과는 평균 0.431이고 전처리 과정에서 발생하는 시간적 요인을 적용한 결과는 0.427로써 그 차이는 미미하였다. 이러한 결과를 바탕으로 전처리 과정의 유무가 미치는 영향이 없는 것으로 가정하여 기존 연구의 검색 속도를 측정하였을 때 평균 0.782의 값을 나타내어 검색 속도 측면에서 기존 연구와는 의미있는 차이를 보임으로써 사용자 만족도 개선 및 검색 속도 향상으로 인한 사용의 용이성이 높아짐을 알 수 있었다. 본 연구의 향후 연구로는 현재의 성과를 바탕으로 검색 용어 리스트의 정규화 과정을 통해 좀 더 통계적인 결과를 보이고, 사용성 개선 측면에서의 연구의 유효성을 감안하여 좀 더 의미치 있는 연구를 진행하고자 한다.

참고 문헌

[1] D. S. Milojicic, et al., "Peer-to-Peer Computing," Technical Report(HPL-2002-57), Hewlett-Packard Company, 2002.

[2] D. Tsoumakos and N. Roussopolulos, “Analysis and Comparison of P2P Search Methods,” University of Maryland, Dept. of Computer Science, 2003.

[3] 김동현, P2P 환경에서 모바일 데이터베이스 서비스, 한국전자통신학회, 2권1호, 2007.
[4] 김분희, “사용성 개선을 위한 P2P 그룹 검색 알고리즘”, 한국전자통신학회, 5권2호, 2010.
[5] D. Tsoumakos and N. Roussopoulos, "A Comparison of Peer-to-Peer Search Methods," WebDB, pp.61-66, 2003.
[6] B. Yang and H. G.Molina, “Improving search in peer-to-peer networks,” ICDCS'02, pp.103-113, 2002.
[7] 김분희, XML-eXtensible Markup Language, 진한M&B 출판사, 2006.

저자 소개

김분희(Boon-hee Kim)

2005년 2월 중앙대학교 컴퓨터공학과 (공학박사)

1999년~(주)CEDAR.com 연구원
2005년~현재 동명대학교 미디어공학과 전임강사
※ 관심분야 : 분산시스템, P2P 검색 기법, HCI 응용