濟州大學校 尖端技術硏究所 論文集, 第14號, 1號, pp.101-108, 2003. 6
Cheju Nat'l Univ. Res. Inst. Adv. Tech. Jour., Vol.14, No.1, pp.101-108, 2003. 6

프랙탈 이론을 이용한 게시판 알고리즘의 설계 및 구현

문 남 원* · 박 충 희** · 김 영 민** · 이 상 준***

The Design and Implementation of Bulletin Board algorithm based on fractal theory

Nam-Weon Mun* · Chung-Hee Park** · Young-Min Kim** · Sang-Joon Lee***

ABSTRACT

We analyze bulletin board application currently used, and we propose the method that minimize disk I/O to enhance the I/O efficiency of the transmission of a large size of data. To improve the performance, we use fractal algorithm. Experiential result shows better performance than the other algorithm.

Our next researches include the effort to improve the speed of computation by converting floating point operation to integer operation. And the method to enhance the precision of calculation is included also.

Key Words : bulletin board, fractal theory, i/o algorithm

Ⅰ. 서 론

현재 인터넷상에서 게시판의 용도는 의사전달 및 지식 공유의 장으로써 대단히 폭넓게 활용되고 있다. 많은 커뮤니티 사이트가 수 십만개의 게시판을 운영하고 있으며 '네이버 지식in'인 경우 하루 2만건 이상의 입력과 수십만 번의 조회가 이루어지고 있으며 이를 통한 정보공유가 활발히 진행되고 있다[1].

하지만 게시판의 데이터가 축적됨에 따라 대용량화되가고 입력과 출력의 속도가 저하되고 있으며 알고리즘 개선보다는 시스템의 분산과 고성능화를 꾀하여 서비스의 질을 유지하고 있다[2,3].

이에 좀더 입출력 속도를 향상시킬 수 있는 게시판 개선 알고리즘 연구를 통하여 저비용으로 서비스를 유지할 수 있도록 한다.

일반적인 게시판의 아래와 같은 조건[4,5]을 만족해야 한다.

첫째, 답글의 개수와 깊이는 무제한이어야 한다.
둘째, 데이터 액세스를 최소한으로 줄여야 한다.
셋째, 수정 및 삭제가 용이해야 한다.

* 제주대학교 산업대학원
Graduate School of Industry, Cheju Nat'l Univ.
** 제주대학교 대학원 컴퓨터 공학과 박사과정
Doctor course of Computer Eng., Cheju Nat'l Univ.
*** 제주대학교 통신컴퓨터공학부
Faculty of Telecommunication & computer Eng., Cheju Nat'l Univ.

Ⅱ. 게시판 동작방식에 대한 고찰

1. 순서갱신형 게시판 (Board01)

1) 개요

- 입력시 인덱스된 필드를 기준으로 하여 자신의 위치를 파악하고 위치 이후의 인덱스 번호를 일괄 갱신하여 새로운 인덱스를 생성한다. 출력시는 인덱스 번호를 기준으로 하여 게시물을 나열한다[6].

2) DB 및 트리 구조

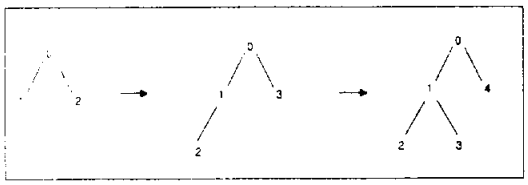

Fig. 1. Tree structure of Board01

Table 1. DB structure of Board01

Field	Type	Key
no	int(11)	PRI
uid	int(11)	
idx	int(11)	
depth	int(11)	
seqno	int(11)	
indate	int(11)	
name	varchar(20)	
passwd	varchar(20)	
email	varchar(50)	
subject	varchar(100)	
text	text	
counter	int(11)	
p_time	double	

3) 입출력 알고리즘

3-1) 원글의 입력

① 글을 입력받은 후 테이블에서 원글순서의 최종 번호를 검색한다.

② 최종번호에 하나를 더한 후 정렬순서번호와 글의 깊이를 0으로 하는 레코드를 삽입한다.

3-2) 답글의 입력

① 답글을 입력받은 후 테이블에서 답글에 대한 본글을 불러온다.

② 본글의 깊이와 본글 순서를 이용하여 답글의 깊이와 본글순서를 정한다.

③ 정렬순서번호가 본글보다 크고 본글과 같은 깊이의 기존글이 있는지 검색하고 있을 경우 기존글의 정렬순서번호를 가져온다.

④ 본글과 같은 깊이의 기존글이 없을 경우 답글의 깊이보다 깊고 본글의 정렬순서보다 가장큰 정렬순서번호를 가지는 기존글의 정렬순서번호를 가져온다.

⑤ 위의 경우가 없을 경우 본글의 정렬순서번호에 하나를 더하여 답글의 정렬순서번호를 정한다.

⑥ 답글의 정렬순서번호보다 같거나 큰 정렬순서번호를 가지는 모든글에 대해 정렬순서번호를 하나씩 증가시킨다.

⑦ 답글을 삽입한다.

3-3) 게시물의 출력

① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지당 글의 개수로 나누고 전체 페이지를 산출한다.

② 페이지당 작업을 위하여 원글번호의 역순과 정렬번호순으로 원글번호와 정렬순서번호를 가져온다.

③ 각페이지당 시작글의 원글번호와 정렬순서번호를 배열에 배당한다.

④ 검색하고자 하는 페이지의 시작글의 원글번호와 정렬순서번호를 배열에서 찾고 페이지당 글의개수 만큼 테이블에서 불러내어 출력한다.

⑤ 출력 질의

```
Query="select * from $table  where (uid=$a
and idx>='$b') or (uid <$a) order by uid
desc,idx limit $page_quota" ;
```

2. 고정위치형 게시판 (Board02)

1) 개요

- 입력시 기본글과 답글사이에 관계번호와 답글의

위치번호를 넣고 관계번호를 우선하여 문자로 된 위치번호순으로 글을 가져온다.

2) DB 및 트리 구조

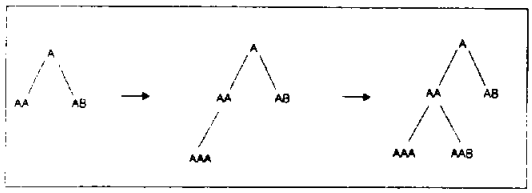

Fig.2. Tree structure of Board02

Table 2. DB structure of Board02

Field	Type	Key
no	int(11)	PRI
uid	int(11)	
thread	varchar(8)	
depth	int(11)	
indate	int(11)	
name	varchar(20)	
passwd	varchar(20)	
email	varchar(50)	
subject	varchar(100)	
text	text	
counter	int(11)	
p_time	double	

3) 입출력 알고리즘

3-1) 원글의 입력

① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.

② 최종번호에 하나를 더한후 문자위치번호순서를 "A"로 하고 글의깊이를 0으로 하는 레코드를 삽입한다.

3-2) 답글의 입력

① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.

② 본글의 깊이를 이용하여 답글의 깊이를 정의한다. 답글의 깊이가 7을 넘을 경우 에러를 표시하고 작업을 빠져나간다.

③ 답글의 깊이와 깊이가 같고 본글의 문자위치번호 순서와 같은 순서를 갖는 마지막 기존글을 가져온다.

④ 위의 마지막 기존글이 없을 경우 본글의 문자위치번호에 "A"를 더한다.

⑤ 마지막 기존글의 마지막 문자열을 아스키값으로 변환하여 이에 1을 더하고 이값이 90을 넘을 경우 에러를 표시하고 작업을 빠져나간다.

⑥ 아스키값을 문자로 변환하고 본글의 문자위치번호에 문자를 더한다.

⑦ 답글을 삽입한다.

3-3) 게시물의 출력

① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지당 글의 개수로 나누고 전체 페이지를 산출한다.

② 페이지당 작업을 위하여 원글번호의 역순과 문자위치번호순으로 원글번호와 문자위치번호를 가져온다.

③ 각페이지당 시작글의 원글번호와 문자위치번호를 배열에 배당한다.

④ 검색하고자 하는 페이지의 시작글의 원글번호와 문자위치번호를 배열에서 찾고 페이지당 글의개수 만큼 테이블에서 불러내어 출력한다.

⑤ 문자위치번호의 우선순위는 A > AA > AAA > AB > ABA 가 된다.

⑥ 출력 질의

```
$query="select * from $table  where (uid=$a
and thread>='$b') or (uid <$a) order by uid
desc,thread limit $page_quota" ;
```

3. 연결리스트 형태의 게시판 (Board03)

1) 개요

- 입력시 본글과 답글사이에 순회 할 수 있는 자식노드를 넣고 출력시에 노드를 따라서 글을 가져온

다[7].

2) DB 및 트리 구조

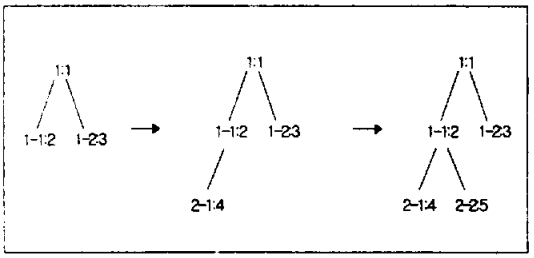

Fig 3. Tree structure of Board03

Table 3. DB structure of Board03

Field	Type	Key
no	int(11)	PRI
uid	int(11)	
cnode	char(1)	
pid	int(11)	
seqno	int(11)	
depth	int(11)	
indate	int(11)	
name	varchar(20)	
passwd	varchar(20)	
email	varchar(50)	
subject	varchar(100)	
text	text	
counter	int(11)	
p_time	double	

3) 입출력 알고리즘

3-1) 원글의 입력

① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.

② 최종번호에 하나를 더한후 모노드,깊이,깊이에 따른 순서를 "0"으로 하고 자식노드를 "N"으로 하는 레코드를 삽입한다.

3-2) 답글의 입력

① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.

② 본글의 절대번호를 부모노드로 하는 글들의 깊이에 따른 순서(seqno)중 기존글의 제일 큰번호를 가지고 온다.

③ 위의 번호가 없을 경우 순서번호를 1로 하고 있을 경우 기존글의 번호에 1을 더한 순서번호를 갖는다.

④ 본글의 깊이에 하나를 더한 깊이를 갖고 본글의 자식노드 유무를 "Y"로 변경한다.

⑤ 답글을 삽입한다.

3-3) 게시물의 출력

① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지당 글의 개수로 나누고 전체 페이지를 산출한다.

② 먼저 원글만을 원글번호의 역순으로 가져온다.

③ 원글에 자식노드가 있을 경우 자식노드를 가져오는 함수 get_cnode()를 실행한다.

④ get_cnode는 자식노드가 없을때까지 실행되는 재귀함수이다.

⑤ 출력번호(s_no)가 원하는 페이지의 출력번호보다 높을 경우 출력을 중지한다.

⑥ 출력 질의

$query_no="select no,cnode from $table where uid != 0 order by uid desc" ;

4. 게시판간 장단점 분석

- 정렬갱신형 게시판은 가장 빠른 속도로 출력 할 수 있으나 입력시 최악의 경우 전체글의 정렬번호를 갱신하여야 하며 이때 입력 속도의 저하가 발생한다.

- 고정위치형 게시판은 글의 개수와 무관하게 입출력 속도를 보장할 수 있으나 답글의 개수와 깊이가 제한된다.

- 연결리스트형 게시판은 무한 확장 할 수 있으며 안정된 입력속도를 보장하지만 입력된 글의 개수와 비례하여 출력의 속도가 현저하게 저하된다.

III. 입출력 속도 향상을 위한 프랙탈 알고리즘

1. 프랙탈(Fractal)

언제나 부분이 전체를 닮는 자기 유사성(self-similarity)과 소수(小數)차원을 특징으로 갖는 형상. '프랙탈'이란 이름은 1975년 B.B.만델브로트에 의해 지어졌으나[8], 이러한 형상들에 관한 추상적 논의는 훨씬 이전부터 있었다. 칸토르집합, 코흐눈송이, 시어핀스키삼각형, 페아노곡선, 줄리아 집합 등이 그 예이다. 만델브로트가 정의한 만델브로트집합은 현대에 소개된 프랙탈의 예이다.

프랙탈의 개념은 조각나거나 가지친 자연구조의 배열뿐 아니라 브라운 운동(Brown motion)에서부터 물방울 운동에 이르기까지 구조의 역동적인 성질들을 묘사하는데 사용될 수 있다. 프랙탈은 과학자들이 자연현상을 측정하는데 사용할 수 있는데, 예를 들면 전기를 전도하는 방식들을 연구하는 데 사용할 수 있다. 그러나 수학 프랙탈은 자연적인 물체에서는 실제로 발견할 수 없는 성질들을 가지고 있다. 무한히 되풀이해서 확대되면서 똑같이 보이는 구조는 실제로 없다. 그럼에도 불구하고 프랙탈 모델은 적어도 한정된 범위에서 실제와 비슷한 접근방법을 제공해 준다.

요크는 로렌쯔가 주장한 '초기 조건의 민감성'은 일상생활의 도처에 존재한다고 생각했다. 요크는 생물학자인 로버트 메이(Robert May)와의 공동연구에서 카오스계에서 나타난 질서를 찾았다.

만델브로트는 자연의 경향성을 밝히려 했고, 사회의 복잡한 무질서 속에서 일정한 질서가 있음을 찾으려 했다. 이와 같은 질서는 뉴턴역학에서 보여지는 단순 명쾌한 질서는 아니었다. 그의 업적은 자연이 가지고 있는 자체 유사성에 대한 연구에서 절정에 이르게 된다. 카오스는 현재 비선형 동력학 이론과 실험도구로서의 컴퓨터 발전과 맞물려 성장하고 있다. 또한 카오스는 수학, 물리학, 생물학, 화학, 지질학, 공학, 생태학, 사회학, 경제학, 과학 철학 등 과학 및 사회 전반에 걸쳐 근본적인 사고의 변화를 가져오고 있으며, 현재공학, 산업에서의 응용이 매우 활발하다[9].

2. 제안 알고리즘을 위한 프랙탈

1) 게시판 알고리즘을 위한 프랙탈

① 기본모양

프랙탈 알고리즘을 도형화 하면 Fig.4 와 같다

② 함수의 정의

- 맨 처음 가지는 길이 "R1"과 각 "θ1"를 가지는 선분을 그린다.

$x1 = R1 * \cos(\theta 1)$

$y1 = R1 * \sin(\theta 1)$

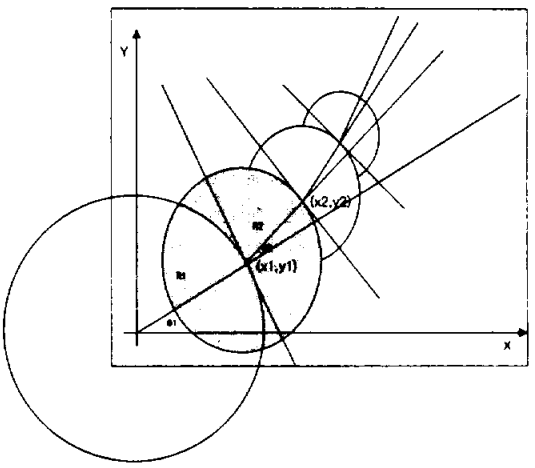

Fig.4. Figure of fractal algorithm

- 두 번째의 가지는 길이와 각이 일정하게 줄어드는 선분을 그린다. 두 번째 가지의 길이(R2)는 R1*(VAR_RED_R)가 되고 각"θ2"는 θ1*(VAR_RED_T)가 된다. 여기서 VAR_RED_R 은 선분이 줄어드는 비율이고 VAR_RED_T는 깊이에 따라 각이 줄어드는 비율이다.

$x2 = x1 + R2 * \cos(\theta 1+\theta 2)$

$y2 = y1 + R2 * \sin(\theta 1+\theta 2)$

- 세 번째의 가지는 길이와 각이 동일비율로 줄어드는 선분을 그린다. 여기서 세 번째 가지의 길이(R3)는 R2*(VAR_RED_R)가 되고 각" θ3"는 θ2*(VAR_RED_T)가 된다.

x3 = x2 + R3 * cos(θ1+θ2+θ3)
y3 = y2 + R3 * sin(θ1+θ2+θ3)

- 그러므로 n번째의 가지가 가지는 좌표 xn,yn은
xn = x(n-1) + Rn * cos((∑θ(n-1)) + θn)
yn = y(n-1) + Rn * sin((∑θ(n-1)) + θn)
가 된다.

3. 프랙탈 알고리즘을 이용한 게시판 구현

1) 구축환경

- CPU : celeron 266
- RAM : 96M
- OS : RedHat Linux 8.0
- Web Server : Apache 2.0
- Language : PHP 4.2.2
- DBMS : MySQL 3.23

2) DB 구조

Table 4. DB structure of Board04

Field	Type	Key
no	int(11)	PRI
uid	int(11)	
theta	double	
px	double	
py	double	
thetaN	double	
theta_sum	double	
radiusN	double	
seqno	int(11)	
depth	int(11)	
indate	int(11)	
name	varchar(20)	
passwd	varchar(20)	
email	varchar(50)	
subject	varchar(100)	
text	text	
counter	int(11)	
p_time	double	

3) 입출력 알고리즘

3-1) 원글의 입력

① 글을 입력받은후 테이블에서 원글순서의 최종 번호를 검색한다.
② 최종번호에 하나를 더한후 자신의 반지름(radius N)을 5로 하고 진행각의 합(theta_sum),원점좌표각(theta),자신의 진행각(thetaN)을 동일하게 초기값 π/1000 으로 설정한다.
③ 깊이(depth)를 0으로 하고 레코드를 삽입한다.

3-2) 답글의 입력

① 답글을 입력받은후 테이블에서 답글에 대한 본글을 불러온다.
② 답글과 깊이(depth)가 같으면서 일련번호가 제일 큰 글을 불러온다.
③ 위의 글이 없을 경우 본글에 대해서 반지름을 일정비율(99/100)로 줄이고 진행각을 일정비율(33/100)로 줄인다.
④ 글이 있을 경우 제일 큰 글에 대해서 반지름을 동일하게 하고 진행각을 일정비율(49.3/100)로 줄인다.
⑤ 자신의 반지름과 진행각을 가지고 바로 전 글의 좌표에 더하여 자신의 좌표를 계산한다.
⑥ 자신의 좌표를 이용하여 원점으로부터의 각을 계산해 낸다.
⑦ 답글을 삽입한다.

3-3) 게시물의 출력

① 전체페이지 수 계산을 위하여 테이블에서 모든글의 개수를 가져와서 페이지당 글의 개수로 나누고 전체 페이지를 산출한다.
② 페이지당 작업을 위하여 원글번호(uid)의 역순과 원점좌표각(theta)순으로 원글번호와 원점좌표각을 가져온다.
③ 각페이지당 시작글의 원글번호와 원점좌표각을 배열에 배당한다.
④ 검색하고자 하는 페이지의 시작글의 원글번호와 원점좌표각을 배열에서 찾고 페이지당 글의개수만큼 테이블에서 불러내어 출력한다.
⑤ 출력 질의
$query="select * from $table where (uid=$a

and theta>='$b') or (uid <$a) order by uid desc,theta limit $page_quota" ;

IV. 결 과

1. 테스트 전제 조건

1) 네트워크상의 데이터 전송속도와 무관하게 측정하도록 한다.
2) 입력되는 데이터의 양은 동일하게 한다.
3) 에러 발생시 입력 및 출력을 중지한다.
4) 입력시간은 데이터를 서버로 모두 전송된 후부터 데이터베이스에 입력 직전 시간까지를 측정한다.
5) 출력시간은 클라이언트 환경과 무관하게 클라이언트 요청 후 서버에서 클라이언트 전송직전까지의 시간을 측정한다.

2. 입력속도 비교

1) 원글 1만개 입력 시간 측정 비교(원글 수직달기)
네 가지 게시판 유사한 입력 시간 측정됨

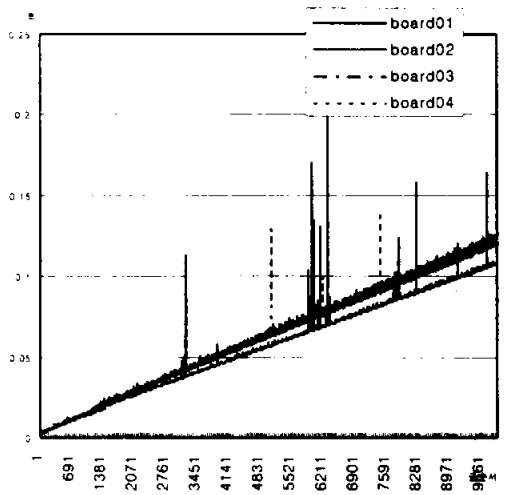

Fig.5. Original text input process time graph

2) 원글에 깊이 1만개의 답글 입력시간 측정 비교(답글 수평달기)

board01 > board03 ,
board04 에러발생(668개) ,
board02 에러발생(7개)

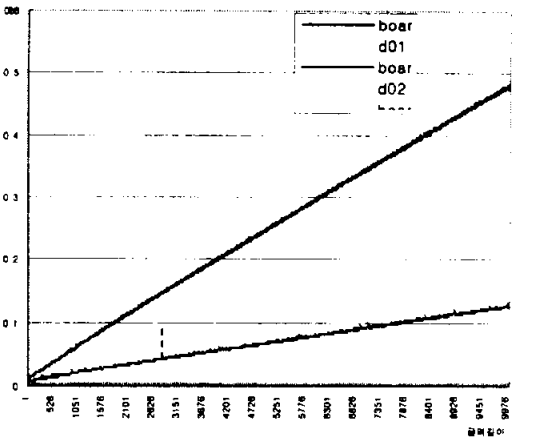

Fig.6. Reply text input process time graph 1

3) 원글 1번에 1만개의 답글 입력 시간 측정 비교(답글 수직달기)

board01 > board03 ,
board04 에러발생(48개),
board02 에러발생(26개)

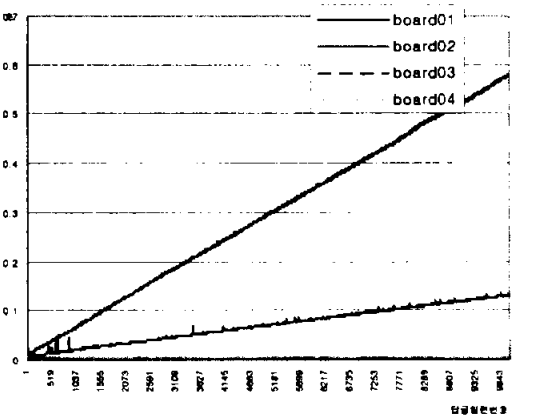

Fig.7. Reply text input process time graph 2

2. 출력속도 비교

게시물의 출력은 비교를 위해서 정렬갱신형과 연결리스트형의 게시물수를 제안알고리즘 게시판의 개수와 같은 669개로 제한하였다.

제안알고리즘 게시판과 정렬갱신형 게시판은 출력속도가 일정한 반면 연결리스트형은 급격한 기울기로 출력속도가 저하되고 있다.

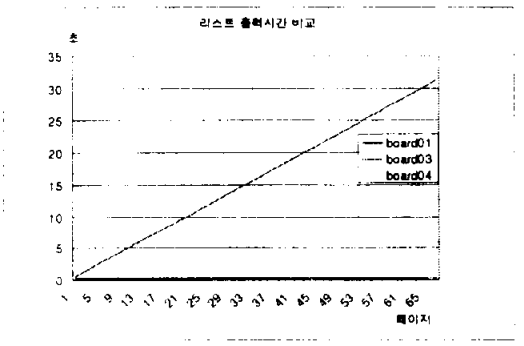

Fig.8. List output process time graph

V. 결론 및 향후 연구

본 논문에서는 기존 게시판의 입출력 속도를 향상시킬 수 있는 개선된 알고리즘을 제안하였다. 기존 알고리즘과 비교하여 본논문에서 제안한 프랙탈 함수를 이용한 알고리즘이 입출력에 대해서 안정된 속도를 보장하고는 있으나 답글의 깊이 제한과 개수의 제한은 역시 현재 컴퓨터 시스템상에서 극복해야 할 점으로 남아있다.

DBMS에서의 실수형인 double은 2.225E-308의 수를 표현할 수 있고 PHP에서 원주율은 float형으로 14자리의 정밀도를 갖는다. 이로 인해 제안 알고리즘에서 깊이와 답글이 달리는 개수가 제한이 생기며 이를 극복할수 있는 알고리즘의 확장형태가 연구되어야 하겠다.

참고문헌

1) http://kin.naver.com
2) http://kr.yahoo.com
3) http://www.daum.net
4) http://www.phpschool.com/bbs2/inc_board.html
5) http://jsboard.kldp.net
6) 이석호, 2003, 파일처리론, 정익사, 서울, p.223
7) 카일루든,2002,C로 구현한 알고리즘 (Mastering Al-gorithms with c), 허욱 역, 한빛북, 서울, p. 124
8) Shishikura,M,1991, The Hausdorff dimension of the boundary of the Mandelbrot Set and Julia Set, SUNY Stony Brook, Institute for Mathematics Sciences, Preprint p.332
9) 제임스 글리크,1993, 카오스:현대과학의 대혁명, 박배서 · 성운하 역, 동문사, 서울, p.76