Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 5, No. 2, May 2014), Pages: 7-14
www.jacr.iausari.ac.ir

Improving Reliability through Selecting Data in Grid Distributed System and Comparing it with Other Presented Algorithms

Sedigheh Navaezadeh [*], Iman Zangeneh, Elham Tavakol

Sama Technical and Vocational Training College, Islamic Azad University, Mahshahr, Branch Mahshahr, Iran

snavaezade92@yahoo.com; zangeneh_i@yahoo.com; elham.tavakol90@gmail.com

Received: 2014/02/12; Accepted: 2014/04/03

Abstract

In the past, one of the biggest problems requiring long and complicated computing was lack of appropriate software and hardware facilities in order to compute them in a proper and logical time. Due to this issue, sometimes, computing lasted for several days, months or years. Therefore, computer science pioneers tried to find a solution for this problem. One of the ways was to present Grid computing. In order to present high-quality service, Grid integrated separate organizations, and represented virtual organization. In this study, we improved reliability by using iteration feature as well as distributing and duplicating resources. We considered and used the first nearest resource of the second nearest resource for duplication in order to place data. In addition, in this study, we compared our proposed method with other algorithms.

Keywords: *grid, data propagation, firefly algorithm*

1. Introduction

Grid technology which its first steps have been taken from 1996 for the next generation internet, is the result of cooperative efforts of America state, universities, scientific centers and also individual companies. This phenomenon with the best and fast quest possibilities, very high speed and strong possibilities for all types of scientific research considered as a competent alternative of internet. Internet network provides processed information to the people but Girl gives raw information, computing power, sensors and laboratory systems and indeed Grid, converted internet from a static environment to a programmable and dynamic environment. In the other hand, Grid Technology, In contrast web which is a service for exchanging information in Internet, is a software service for sharing computing power and data storage space between computers connected to the internet and its final target is creating a global broad computing and information.

Computing Grid, for the first time, presented about distributed resource and solved complicated problems in dynamic environment for scientific problems. [1] Grid is a heterogeneous distributed environment. Computing Grid is a software and hardware ultra structure which provides cheap comprehensive and stable availability to the

existing computing resources in the network in descriptions which presented later by different people for Grid. Resource subscription problem and problem solving considered in Grid. The most important problem which proposed here is about resource subscription. It must be noted that resource subscription not means direct file exchange but the possibility of easy availability to the network computers and using computing power and other feasibilities which themselves tended subscribing, have been considered. [2]

In this study, data Grid investigated and new method proposed for file management and placing version by the purpose of increasing efficiency and certainty ability in data Grid.

2. Literature Review

In Grid, in addition to users, jobs also need different data accessibility. Nay work in could be the applicant of several files. Certainly, if requested work was in local place, means in the site, response time will be zero. Otherwise, requested file must be transferred from the site to the requested place. In [3] three algorithms introduced for data placement problem:

1) greedy Algorithm: In this method, any time a version selected until optimal solution has been found. In the first step, any site evaluated individually and the site selected by the less cost (TC1) then, second version selected by comparing with the first version and finally (TC2) becomes minimal. The process continued until j was found in a way that (TCj- jctcjtl 1<TC). Final cost equals Tcj and j is the number of versions.

2) centralized area Algorithm. This algorithm just differentiated in selecting the best candidate of any step by greedy method. In this method, existing traffic of any site calculated in its site neighborhood. Then, sites determined by the most numbers of requests and its final cost have been calculated (TC1) .Then operated as greedy algorithm output density maximum Algorithm: In this method, sites selected by maximum density and in any stage, final cost calculated. This operation repeated until reaching a special condition.

In [4] without considering storage resource cost of different networks (loop, tree, …) optimal algorithm and efficiency presented for data placement and in (s) dynamic method presented based on availability weight of any version. Chang prioritized any version according to a weigh and by this Idea files used more previously, will be applied more now. We have two types of propagation in Grid, static and dynamic.

2.1 static propagation

The purpose static propagation is load balance optimization and certain ability which propagated in different places during data initiation and will not change by load amount alteration. Grid environment is a dynamic one and static method is not optimal in this environment. But used due to its algorithm facility.

2.2 dynamic propagation

In dynamic method, Because existing resources considered as a current mode as a result, it is more efficient than static method. But due to its foundation complexity, haven't been much used. In this mode, Decision making for data propagation operations based on factors such as: file size, network delay, system certain liability, network band and network delays and in [6] for both compound of greedy and genetic methods have been used.

2.3 firefly optimization algorithm

Firefly Algorithm is a type of algorithm obtained from nature and collective smart algorithm which presented by yang (2008), this algorithm is a modern technique based on collective behaviors which inspired from firefly collective intelligence is a type of artificial intelligence method social behaviors in the nature based on collective behaviors in neutralized and self-organized foundations. Fireflies generate rhythmic and short beams. Optical patterns of each firefly differentiated form others. Fireflies used these beams for two reasons. 1- pairs attraction process. 2- for attracting hunt. Moreover, these beams used as a protection Mechanism for fireflies. Rhythmic beams and rate of radiation and interval rate between light signals caused two genders attract each other. Any particle of a firefly in multidimensional quest space updated by absorbing dynamically based on a knowledge of firefly and its neighbors.

Firefly optimization algorithm could be stated as follows: [7]

* all fireflies are single- gender and the factor of pairs attractiveness considered not relating to their gender.

* firefly x attracts all fireflies and attracted for all fireflies.

* attractiveness related to their glow. So for any pair of firefly, a worm with less light attracted toward a worm with more light. Attractiveness power related to their beam and the light intensity decreased by increasing the distance between two fireflies. If a firefly is not brighter than the others, their movement will be performed randomly.

* brighter firefly moves randomly (all fireflies could not attract them).

* firefly brightness determined by objective function value. Problem, light intensity could be determined easily by target function.

* firefly particles randomly distributed in quest space based on above principles, two main part existed in firefly algorithm, attracting firefly and movement toward attracted firefly.

2.3.1 General form of firefly Algorithm

General form of firefly algorithm has been shown in the figure (1)

As we can sees in the figure, at first, primary coordination and light intensity rate and the distance between firefly particles determined in quest area. Quest procedure in firefly algorithm is that any firefly compared with all of the other ones individually. If any firefly has less light than the compared one, will more toward a firefly with more light (the problem of finding maximum point) and this process caused particles

centralized around a particle with more light and if in the next generation of algorithm, there is a particle with more light, particles again will move toward particle with more light. Quest stages must be generated relating to maximum number of generation.

Firefly algorithm
Initialize algorithm parameters:
MaxGen:the maximum number of generations
Objective function of f(),where x=(x1,….,xd)T
Generate initial population of fireflies or xi(i=1,2,...,n)
Define light intensity of Ii at xi via f (Xi)
While(t<MaxGen)
For i=1 to n (all n fireflies);
For j=1 to n (all n fireflies)
If (Ij>Ij),move firefly i towards j;end if
Attractiveness varies with distance r via Exp[ϒ r2];
Evaluate new solutions and update light intensity;
End for j;
End for i;
Rank the fireflies and find the current best;
End while;
Post process results and visualization;
End procedure.

Figure 1: firefly algorithm quasi- code

3. proposed Algorithm

Suppose that Grid environment existed with the following elements and parameters:

M: The site with computing elements and separated storing in an election network with p2p topology.

Si; th rate of site(i)storing capacity (k=i<=M)i

C(i,j) the cost of relation between Site i to site c(j)=c(i,j)=(i)

N The file with the names o1 , o2 , … oN and volumes O1, O2, … ok and (1<=k<=N)

Rik: number of site i requests for reading the file k.

Wik: number of site i requests for writing over fill k.

Pk: of any file there are several versions in the sites and main version of the file ok has been shown with pk.

RSk: per any version ok, the information has been shown by RSk.

In order to have compatible process among versions , and due to the change of ok version, alterations sent to pk and then operated from pk toward other versions.

Sik: the closest site costly consisted of ok version toward site i

3.1 cost function calculation

Suppose that Grid environment was a M & N Matrix shape

A) Xik=i, if main version existed in site i.

B) Xik=0 , if main version didn't exist in Site i.

In order to find an appropriate and optimal method in relation costs, cost function calculation will be very important.

Our purpose is reaching the least final cost which will be as follows:

$$\text{Minimize } TC(X) \qquad (1)$$

Here, there are two limitations for the Matrix X.

Total files volume and existing versions in site I should not be more than site I capacity.

$$\sum_{k=1}^{N} Xik <= Si \; for \; all \; 1 \leq i \leq M \qquad (2)$$

4- therefore, the cost of reading and writing regarding to RS will be as follows. [8]

$$X\text{pk} = 1 forall \; 1 <= k <= n \qquad (3)$$

$$R(RS) = \sum_{i=1}^{M} \sum_{k=1}^{N} R_{ik} = \sum_{i=1}^{M} \sum_{k=1}^{N} r_{ik} o_k C(i, R_{ik}) \qquad (4)$$

$$W(RS) = \sum_{i=1}^{M} \sum_{k=1}^{N} W_{ik} = \sum_{i=1}^{M} \sum_{k=1}^{N} W_{ik} o_k [C(i, P_k) + \sum_{j=P_k}^{n} C(P_k, j)] \qquad (5)$$

Equation (4) showed all files final costs of readings in Grid theatrically.

Note; site ski responses reading applications which have the least cost. And we supposed by the closest node costly.

In equation (5) writing cost consisted two stages. Part one, writing in PK and part two, the cost of publication from Pk to all sites consisted of (ok) versions.

$$TC(RS)=R(RS)+W(RS) \qquad (6)$$

3.2 proposed algorithm code

Figure (2) proposed algorithm initiated by an initial population that any of these particles showed a solution. In the end, creating a solution generated until a specified time s expiredi . During any generation particles superiority has been found particles in any generation obtained the best places by using their personal experience and collective intelligence. Finally, any particle revealed its place by any load Here, in order to satisfy the constraints, adjustment function has been used. Adjustment algorithm code has been shown in figure (3) finally, the best particle selected as the best solution.

```
Particle firefly()
{
    For (p=1 ; p<= swarm-size;p++)
          {
   Generate particle Xp randomly
   Generate initial x_ij
Xp= adjustment(Xp)  }
  Do {
    For each particleXp {
  Calculate fitness value of Xp using 1
If fitness value is better then the best fitness value                    }
Choose the particle with the best fitness value
For each particle Xp {
 Generate particle x_ij
Update particle position
Xp= adjustment(Xp)
}
}
While (one of the termination  condition  is not satisfied)
Return best particle
}
```

Figure 2: proposed quasi – code.

```
Particle adjustment (particle Xp)
  For each file
  j=(((p(k)-1)*N)+k);
    Xpj=1;
  }
s=1;
for each site i { if constraint no satisfy
   j=(((i-1)*N)+1);
   g=i*N;
   while j<=g
      if((a(j)==1)&&(i~=p(j-(i*N)+N)))
         calculate Δci(j-iN+N)
         diff1(s)=ch;
         diff2(s)=j;
         s=s+1;
      end
   end
end
sort(diff1)
   while sum<=s(h)
      j=diff2(s1);
      x(j)=0;
      s1=s1+1;
   end
end
```

Figure 3: adjustment Algorithm.

4. Simulation

Method of showing ability of an algorithm according to the other algorithm is evaluating that algorithm with other ones.

This process performed by the aid of simulation. In this study, MATLAB software used for simulation. We compare our algorithm other same ones by using propagation such as genetic and pso. Simulated Data have 10 to 150 nodes. Network structure is E-shape and cost of nodes is between 1 to 10 which shows the interval between packets for reaching the objective. The least value of files were 2 k byte and the greatest one considered 7 megabyte.

Variable process numbers were between 50 to 450. initial site considered randomly and files volume considered 25% to 90%.

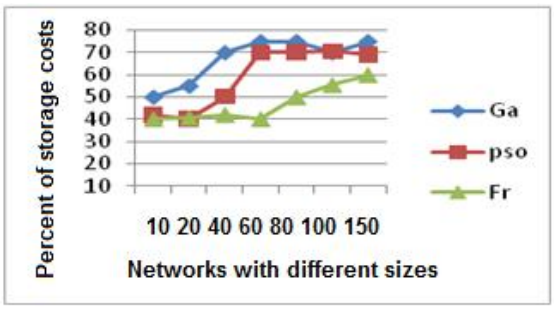

Figure 4: rate of final cost optimization with different sizes networks.

Figure(4) Results of mentioned algorithm simulations in networks showed with different size.

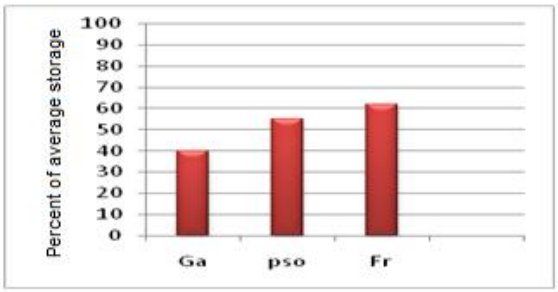

Figure 5: rate of algorithm final efficiency in different networks.

Figure (5) shows middle rate of efficiency for mentioned algorithm in different modes with various parameters.

5. Conclusion

Some of active experts in this field believe that Grid technology accounted as the second chance of internet and developed very fast and everyday new Issues proposed for this Issue or expanded existing discussions. Date generation in distributed system is a method which ensured these systems efficiency optimization. In Data generation, we need number and place of versions determination. In this study, by using cost-based

method, we present a new algorithm. Supposed algorithm presents more optimized results for investigated algorithm.

6. References

[1] Foster I., Kesselman C, *The Grid Blueprint for a new computing infrastructure.* Morgan Kaufman. .2004.

[2] Tanenbaum A S., Van Steen M., *Distributed systems Principles and Paradigms.* Prentice Hall, 2007.

[3] Yang,m.Fei Z. ,A Model for Replica placement in Content Distribution Networks for Multimedia Application ,*IEEE Intl .Conference on communication* ,vol.l. pp.557-561, 2003.

[4] Wolfson ,o.;Milo,A; The Multicast Policy and Its Replicated Data Placement, *ACM trans.Database systems* ,Vol.16,NO.1,pp.181-205, 1991.

[5] Chang R, Chang H.. A dynamic weighted data replication strategy using access-weights in data grids. *Journal of Supercomputing*.45:277-295, 2008.

[6] Ranganathan K. and Foster I, Identifying dynamic replication strategies for a high-performance data grid, *Proceedings of International Workshop on Grid Computing*, pp. 75–86, 2001.

[7] Yang , x. s;"firefly algorithm for multimodal optimization' in: stochastic Algorithm foundations and applications, *SAGA,lecture notes in computer science*,2009.

[8] Manghui Tu, Member, IEEE, Peng Li, I-Ling Yen, Member, IEEE, Bhavani Thuraisingham, Fellow, IEEE, and Latifur Khan, Member, IEEE" Secure Data Objects Replication in Data Grid "; *IEEE Transactions on dependable and secure computing,* Vol. 7, No. 1, January-March 2010.