

A Distributed Approach for Disk Defragmentation

Mahdi Abed Salman

College of Science for Women/ University of Babylon

Hasanain Ali Al Essa

Khaldoon alhussayni

University Computers Center/University of Babylon

alhussayni@uobabylon.edu.iq

Abstract

Fragmentation is a computing problem that occurs when files of a computer system are replaced frequently. In this paper, the fragments of each file are collected and grouped, thanks to ant-colony optimization ACO, in one place as a mission for a group of ants. The study shows the ability of ants to work in a distributed environment such as cloud computing systems to solve such problem. The model is simulated using NetLogo.

Keywords: disk defragmentation, multi agent system, ant colony optimization, distributed cooperation, Cloud computing.

الخلاصة

تجزئة الملفات هي واحدة من مشاكل الحاسبة الالكترونية في الجزء المخصص لخرن البيانات داخل الحاسبة (القرص الصلب) التي تحدث عندما يتم استبدال وحذف الملفات بصورة مستمرة. في هذا البحث تم تصميم نموذج لتجميع اجزاء الملف الواحد الى مجموعة واحدة في مكان واحد ويتم تنفيذ المهمة من قبل مجموعة من النمل وذلك باستخدام احدى خوارزميات النكاه الاصطناعي المعروفة بمستعمرات النمل (Ant Colony). الدراسة البحثية بينت قابلية النموذج للتطبيق في بيئة موزعة كما في انظمة الحوسبة السحابية لحل هكذا مشكلة. تم عمل محاكاة للنموذج باستخدام بيئة النمذجة البرمجية (NetLogo).
الكلمات المفتاحية: إلغاء تجزئة القرص الصلب، نظام متعدد الوكلاء، خوارزمية مستعمرات النمل، التوزيع المتعاون، الحوسبة السحابية.

1- Introduction

In the past decade, the technology of computer systems has showed a rapid development for computer systems. The capacity and speed of computer storage are increased. The ratio of read/write file to the disk becomes very high.

Usually, at the beginning of the hard disk life usage, a contiguous storage area is allocated when writing files to disk since the free space is still continuous. However, deleting some old files creates non-contiguous free spaces that make this strategy inapplicable. Hence, when willing to write a new file, disk management system searches for a sufficient free space. If no such space is available, a linked list of available spaces is created and the file is fragmented over this list. for detailed information on disk control, reader can see for example [Mamun,2006].

This will decrease the performance of the hard disk and consequentially the overall computer system. To avoid such a problem the hard disk has to be defragmented frequently and files are rewritten to contiguous spaces.

The problem of disk defragmentation is the high cost in both time and memory usage. A simple and classical algorithm may use a backup disk and move all files out of the disk under process[Sanvido,2009]. Finally, files are rewritten back as one block for each one to the original disk after reformatting it.

Many researchers proposed solutions to optimize the time and space required to defrag a hard disk.

Tabero *et al.*, [Tabero *et al.*,2008] developed a fragmentation metric, and used it to implement a location selection heuristic, as well as to fire out routine defragmentation measures in high fragmentation situations. The proposed a location

selection heuristic technique initially avoids further fragmentation by selecting a suitable storage location for each writing process.

Deduplication technique is widely used to increase hard disk (especially those used for backup) performance by not writing blocks that similar for already existing ones and just pointing for them within file structure map[Akhila,2016].

The most proposed solutions relay on decreasing the number of movements required to reach the final state of the disk. In a previous work [Salman,2008], the genetic algorithm GA is used to elect best fragment movement strategy and then applying it. The population of the GA represents set of virtual final images for the disk. The fitness is calculated as the total movements have to be made to reach the result. The objective function was minimum number of movements i.e. the chromosomes with small fitness are preferred.

The contribution of this paper is using Ant colony optimization for disk fragmentation problem. This study overcomes the problem of disk fragmentation and suitable for shared disks such as in the cloud where many users may use same physical array of disk. In such environment and as a result to high usage rate, it is highly probable such disks are suffering from fragmentation problem most of time.

This paper is organized as follow: section one (this one) presents an introductory of the study. Section two defines and characterizes features of multi agent systems. In section three, the principals of Ant colony optimization are presented. Section four describes the method by which the proposed model is built and simulated. The results are presented and discussed in sections five. Finally, section six concludes the study.

2- Multi-agent system

A multi agent system MAS is a computer based solution methodology in which the system is modeled through a set of entities called agents working with specified environment. An agent (usually software) has a state and capable of making decisions autonomously depending on its state and environment state[Ashri,2003].In the environment of MAS often need to be agents of the conversation with each other so as to achieve a particular goal or to improve performance. Agents communicate through the environment; share its resource and change it.

The key element of using MAS is that agent can learn from historical actions, compete for available resource or cooperate somehow to reach a global solution [Vidal,2006]. Learning leads the agent to benefit from old decisions and accelerate achieving the design objectives.

3- Ant-colony algorithm

Ant colony optimization (ACO) is distributed optimization technique that is inspired from foraging behavior of some species of ants. It is noticed that ants leave a chemical trail called pheromone on the ground along with their return path from founded food sources to their nest in order to mark that path that should be used by other members of the colony [Dorigo,2006].

The first ant colony optimization algorithm was proposed on the early nineties. From that time, ACO continues attracting the attention of researchers and many successful applications have been presented.

The ant colony optimization is used in computer science to optimize solution for some problems like scheduling[T'kindt, 2002], routing[Bell,2004] and image processing [Hinduja,2016], etc. To the best of our knowledge, we are the first who use such optimization in disk fragmentation problems.

An ant is modeled as agent of a multi agent system, in this paper the term ant and agent is used interchangeably.

4- Methodology

The model is built as a multi-agent system and simulated using NetLogo which is a multi-agent programmable modeling environment. It is developed at Northwestern's Center for Connected Learning and Computer-Based Modeling (CCL) and authored by Uri Wilensky [Wilensky,1999].

A set of agents have been defined to be kept moving in the space and looking for a fragment of specified file. When a fragment is found, the agent brings it back to where others fragments are collected. Along its returning path, it drops some trail to show the route for other agents. Searching agents follow the smell of old trail to find more fragments. The trail evaporates over time.

For the sake of simplicity, the model considers solving a simple problem e.g. a set of blocks belonging to a single file that are fragmented over the disk space (the environment) have to be collected at one area. Furthermore, an assumption is made that file has been fragmented according to normal distribution. Furthermore, for multiple file processing, each file is processed by different colony. i.e using multiple or colored pheromone.

The behavior of an agent depends on its state (looking for fragments or looking for nest) and its partial view (existing of fragments in the neighborhood). The ant behaves according to the pseudo codes that are shown in figure 1. The model stops when each file composes one contiguous fragment.

```

to go
  if hand_has = nothing           // ant departs the nest
    look_for_fragment
  else                             // ant returns back to the nest
    look_for_nest
  endif
end
*****

to look_for_fragment
  if pcolor = color
    set hand_has fragment
  end
end
*****

to look-for-nest
  let num count (neighbors with [reserved?])
  ifelse (nest? or (num > 1)) and not(reserved? or frag?)
    set hand_has = nothing
end

```

Figure 1: ant's behavior pseudo code

5- Results and Discussion

The simulated model is run with the following parameters: for sake of simplicity, the number of fragments were made as 1/5 of total number of patches. The number of agents are empirically chosen to 10 as it usually taken in tradeoff between resource availability of nowadays computers and accepted solution speed. The deposition and evaporation rates of the pheromone are adapted empirically to have a high performance. i.e minimum time (number of ticks) to complete the defragmentation process.

The result is summarized by showing system states at different phases. Some measures are also computed to assess system behavior over time. Figures 2a shows the system state at initial phase where the file fragments are normally distributed in

the space. Agents did not dispatch yet. Figure 2b shows a transition state where agents still collect file fragments onto contiguous space.

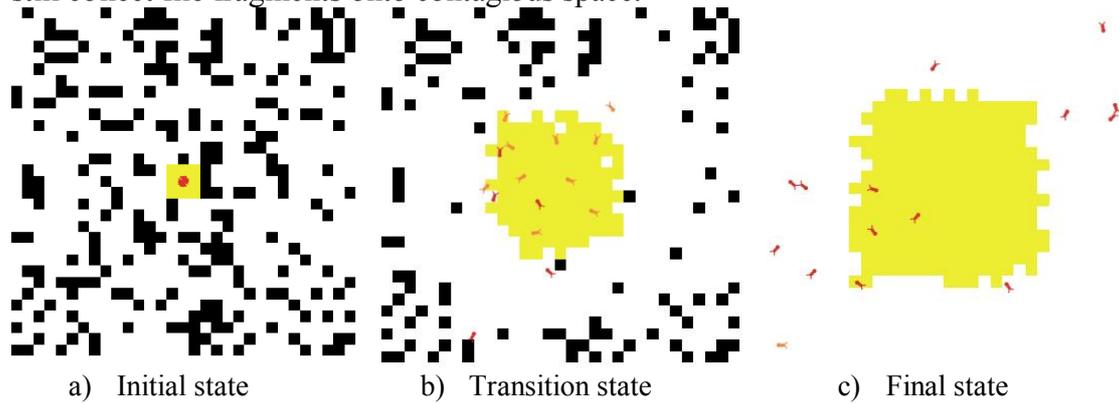


Figure 2: different phases for model simulation in NetLogo
 Black : to be moved, yellow: already moved.

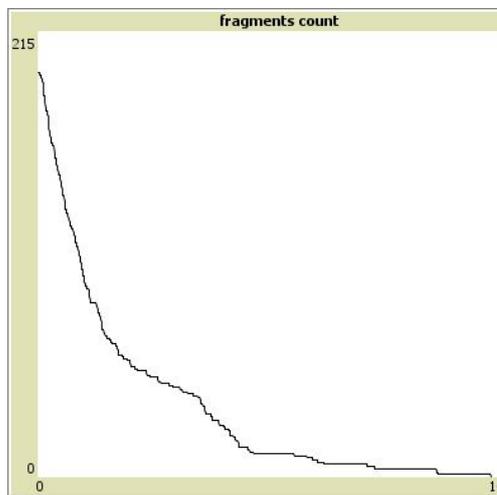


Figure 3: number of fragments

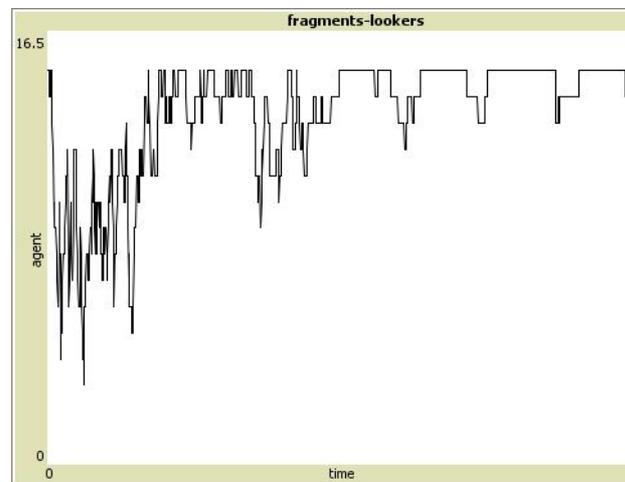


Figure 4: Number of free agents

Figure 3 shows the number of fragments that still have to be brought by ants. At the beginning, ants search for fragments arbitrary. As fast as some ants discover some fragments near the nest, other ants are guided through pheromone trail. Hence, the number of fragments are rapidly decreased. The number of free-hand ants are small as shown in figure 4. At the end, the small number of ants carrying fragments, the amount of pheromone evaporate rapidly that cannot guide other ants to where they found fragments.

To further explain the system behavior, figure 4 shows the number of agents that are looking for file fragments i.e the hands-free ants.

The findings of this study clearly show that the multi-agent based model can be used to have a high performs distributed defragmentation solution. Hence, it useful for shared storage such as in cloud Computing. One explanation for using the technique of Ant colony optimization cannot enhance the performance of the model when the file fragments are distributed normally on the disk space. The only optimized performance is obtained when that fragment form a pile at different area.

The system behavior depends on the number of ants (agents) and the fragments distribution. When the number of agent is small, it takes long time for complete the solution contrary, a large number of agent makes some of them keep moving without finding a file fragment.

This study is limited to the system resource when it uses a single computer, however it should perform better in a cloud environment.

6- Conclusion

We have showed a new possible solution to the disk defragmentation problem. The multi-agent based optimization is used to avoid a critical point of failure problem especially in distributed systems such as cloud computing systems. In addition to its successes in many application domains, the ant colony algorithm can be used efficiently for IT solutions.

The further work is related to extend the model through adding learning capabilities and permitting agents to exchange information for reliable decision making process.

References

- Akhila, K., Ganesh, Amal, Sunitha, C. A, 2016, Study on Deduplication Techniques over Encrypted Data. *Procedia Computer Science*, 87: 38-43.
- Ashri, Ronald, Rahwan, Iyad, Luck, Michael, 2003, Architectures for negotiating agents. In: *International Central and Eastern European Conference on Multi-Agent Systems*. Springer Berlin Heidelberg. p. 136-146.
- Bell, John E., McMullen, Patrick R. , 2004, Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18.1: 41-48.
- Hinduja, P., Suresh, K., Kiran, B. Ravi., 2016, Edge Detection on an Image Using Ant Colony Optimization. In: *Proceedings of the Second International Conference on Computer and Communication Technologies*. Springer India. p. 593-599.
- Jesús Tabero, Julio Septién, Hortensia Mecha, Daniel Mozos, 2008, Allocation heuristics and defragmentation measures for reconfigurable systems management, *Integration, the VLSI Journal*, Volume 41, Issue 2, February, Pages 281-296.
- Kaczmarczyk, Michal and Barczynski, Marcin and Kilian, Wojciech and Dubnicki, Cezary, 2012, "Reducing Impact of Data Fragmentation Caused by In-line Deduplication", In *Proceedings of the 5th Annual International Systems and Storage Conference*, pp :1-12, New York, NY, USA.
- Sanvido, M. A. A., Bandic, Z. Z., and Kirsch, C. M., 2009, "Combo Drive: Optimizing Cost and Performance in a Heterogeneous Storage Device," in *In the 1st Workshop on Integrating Solid-state Memory into the Storage Hierarchy*.
- Mahdi Abed Salman, Mohammed Ubed and Faez Al-Mamory, 2010, "A Genetic Algorithm to Eliminate Disk Fragmentations", *Journal of University of Babylon*, Volume 18.
- Mamun, A., Guo, G. & Bi, C., 2006, "Hard Disk Drive: Mechatronics and Control", CRC Press.
- Dorigo, M., Birattari, M. & Stützle, T., 2006, *Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique*. TR/IRIDIA/2006-023.
- Srinivasan, Kiran and Bisson, Tim and Goodson, Garth and Voruganti, Kaladhar, "iDedup: Latency-aware, Inline Data Deduplication for Primary Storage", 2012, In *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, pp 24-24, San Jose, CA, USA.
- T'kindt, Vincent, *et al.*, 2002, An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, 142.2: 250-257.
- Wilensky, U., 1999, NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.