International Journal of Computer Science and Engineering Communications,
Volume.5, Issue.3 (2017): Page.1627-1632
www.ijcsec.com

A Study on Various Dynamic Bin Packing Algorithms for Clouds

Sneha T Subrahmanian[1], Miji K Raju[2] ,T. Sivakumar[3]
[1,2]PG Scholar, [3]Assistant Professor
[1,2,3]Department of Computer Science and Engineering
[1,2,3]Maharaja Institute of Technology, Coimbatore, India
snehats19@gmail.com[1], mijikraju@gmail.com[2],sivakumarrts@gmail.com[3]

***Abstract**: This paper is based on the analysis of different bin-packing algorithms used for resource allocation in Clouds. Dynamic Bin-Packing (DBP) is an alternative form of classical bin- packing in which, items may arrive and depart at random time intervals. Dynamic Bin-Packing has been used to model the resource consolidation problems in Cloud computing. There are so many algorithms, being used to find an optimal solution for DBP problems in Cloud environment such as, Harmonic algorithm, Approximation algorithms, K-binary Algorithms etc.. An optimal solution to a Bin-Packing problem uses the fewest number of bins possible. The worst case and average case behavior of various bin-packing algorithms has been analyzed and the results are compared.*

***Keywords**: Bin-Packing, Dynamic Bin-Packing, Algorithms*

1. INTRODUCTION

Cloud computing is an attractive computing model since it allows for the provision of resources on-demand. The classical one dimensional bin-packing has long served as a proving ground for new approaches to the analysis of approximation algorithms. The standard Dynamic Bin Packing (DBP) problem [1] considers a set of items, each having an arrival time and a departure time. The items are to be packed into bins in an online manner such that the total size of the items in each bin does not exceed the bin capacity at all times. A bin is opened when it receives the first item and is closed when all items in the bin depart. The Objective of DBP is to minimize the maximum number of concurrently open bins in the packing process.

2. OVERVIEW

2.1 Bin Packing

Bin packing is a classical combinatorial optimization problem, given a set of items, the objective is to pack the items into a minimum number of bins such that the total size of the items in each bin does not exceed the bin capacity and can be formulated as follows: Given a sequence of items $=\{1,2,\dots,n\}$ with sizes $s1,s2,\dots,sn \in (0,1]$, find a partition of the items into sets of size 1 (called bins) so that the number of sets in the partition is minimized [1]. Furthermore, the sum of the sizes of the pieces assigned to any bin may not exceed its capacity. A bin is empty if no piece is assigned to it, otherwise it is used.

2.2 Dynamic Bin Packing

In the classical bin packing problem, we want to pack a sequence of items, each with size in the range [0, 1] into a minimum number of unit-size bins. Dynamic bin packing is a generalization of the classical bin packing problem introduced by Coffman et al. [2]. This generalization assumes that items may depart at arbitrary time. The objective is to minimize the maximum number of bins ever used over all time.

2.3 Algorithms

Consider packing problems of one dimension, the objects have a single dimension (cost, time, size, weight, or any number of other measures). Problems of higher dimension have objects with more measures under consideration (cost and weight, length, width, and depth, etc.). The object is to pack a set of items into as few bins as possible. A formal definition of the bin packing (BP) problem follows.

Given a list of objects and their weights, and a collection of bins of fixed size, find the smallest number of bins so that all of the objects are assigned to a bin. Some of the more popular algorithms are given in the following discussion.

2.3.1Approximation Algorithm

Approximation algorithms for the classical bin packing problem are First Fit (FF), Next Fit (NF) and Best Fit(BF). They are briefly described below.

Next Fit:
Next Fit is a bounded space online algorithm in which the only partially filled bin that is open is the most recent one to be started. It uses one active bin into which it packs the input. Once the free space in this bin becomes too small to accommodate the next item, a new active bin is opened and the previous active bin is never used again. This process continues until there are no more elements.

First Fit:
First Fit achieves a worse running time as it keeps all non-empty bins active and tries to pack every item in these bins before opening a new one. If no bin is found, it opens a new bin and puts the item in the new bin. So, the restriction of using a single bin is removed entirely and all partially filled bins are considered as possible destinations for the item to be packed.

Best Fit:
Best Fit is the best known algorithm for on-line bin packing which emerges as the winner among the various online algorithms: It is simple and behaves well in practice, and no other algorithm has a better both worst case and average uniform case. Best Fit (BF) picks (among the possible bins for the item) the one where the amount of free space is minimal. It picks the bin with the least amount of free space in which it can still hold the current element.

2.3.2 Offline Algorithms

An offline algorithm simply repacks everything each time an item arrives. Packing large items is difficult with an online algorithm, especially when such items occur later in the sequence. There are two important offline algorithms for bin packing. They are First Fit Decreasing and Best Fit Decreasing.

First Fit Decreasing

This algorithm first sorts items in non-increasing order with respect to their size and then processes items as the First Fit algorithm. For example, let us consider the following eight items of sizes 4, 1, 2, 5, 3, 2, 3, 6, 3, that need to be packed into bins of size 8. With the First Fit Decreasing algorithm we sort the items into descending order first.

Best Fit Decreasing

Like First Fit Decreasing, Best Fit Decreasing initially sorts items in non increasing order with respect to their order and then processes them sequentially. The difference between the two algorithms is the rule used for choosing the bin in which new item is inserted while trying to pack. Best Fit Decreasing chooses a bin with the minimum empty space to be left after the item is packed into a bin.

2.3.3. Online Algorithms

Harmonic Algorithm

The algorithms define types for the items based on their sizes and for each type the items are packed in dedicated bins which contain only items of one specific type.. Items of one type releases at a time, then departs most of them from the packing, without emptying any bin, and proceeds to release items of a different type. Previously opened bins will not pack items of a different type, thus the wasted space in existing bins can be maximized.

Epstein's Algorithm

The bounded space algorithm of each value of $m > 1$ such that for every $m>1$, changes on the lower bin are made. Epstein's algorithm works the same as the Harmonic algorithm except that in type M bins (the small type bins), at most M items are stored at any time. Whenever M items are stored in the small type bin, we close the bin and open a new one. For item classification, we partition the interval [0, 1] into sub-intervals.

K-BINARY Algorithm

K-Binary is one of the simplest unsupervised learning algorithms to group the bins according to the interval. To achieve this, the interval [0,1] is partitioned into sub intervals. A bin which received the full amount of items (according to its type) is closed and therefore new bin is opened. Each bin will contain only items from one sub-interval (type).

3. LITERATURE SURVEY

Yusen Li et al(2016) [1] proposed a MinTotal Dynamic Bin Packing problem that aims to minimize the total cost of the bins used over time. They have analyzed the competitive ratios of the commonly used First Fit, Best Fit and Any Fit packing algorithms for this problem. Dynamic Bin Packing problem in which each item is allowed to be assigned to only a subset of bins to cater for the interactivity constraints of dispatching playing requests among distributed clouds in cloud gaming.

E. G. Coffman et al (2013) [2] studied the problem of allocating memory of servers in a data center based on online requests for storage. Over-the-net data backup has become increasingly easy and cheap due to cloud storage. Given an online sequence of storage requests and a cost associated with serving the request by allocating space on a certain server one seeks to select the minimum number of servers as to minimize total cost.

Ye Hu et al (2005) [6] The results in this paper provide valuable insights into the performance of alternative resource allocation strategies and job scheduling disciplines for a cloud computing infrastructure. In the investigation, the service level agreement is based on response time distribution, which is more relevant than the mean response time with respect to the performance requirement of interactive applications. They have developed an efficient and effective algorithm to determine the allocation strategy that results in smallest number of servers required.

Suganya et al (2012) [7] multiplexes virtual to physical resources adaptively, based on the changing demand. Service Interruption can be avoided and the services are provided to the respected users based on the user's request. Green computing can be achieved by terminating the idle virtual machines and to adjust the CPU power dynamically. It achieves both overload avoidance and green computing for systems with multi resource constraints.

4. ANALYSIS

In this paper a few methods are discussed in order to obtain the best possible solution for any given sequence of requests and any value of M. Different sequences of requests with different partitions produce different results depending on size and data used. As the requests get large, the number of bins generated by K-Binary algorithm compared to Harmonic does not give good results because of the chosen size. The performance of the K- binary algorithm based on the total number of bins. An algorithm, K-binary, is proposed that uses fewer bins than the algorithm proposed by Epstein, and a slightly more bins than the algorithm proposed by Lee and Lee. At the same time, the K-Binary algorithm works better than both previously proposed algorithms. These algorithms are executed with extensive simulations with small or large sequences of requests and random values for the requests.

5. CONCLUSION

In this paper, different bin-packing algorithms used to minimize the number of bins used for serving a fixed number of requests are discussed. The algorithm proposed by Lee and Lee has a competitive ratio that is very close to the optimum, but has a high variation on the types of bins used, in the sense that for random requests it either uses a lot of small type bins or a lot of large type bins. The algorithm proposed by Epstein has a worse competitive ratio, i.e. uses more bins for packing the items than Lee and Lee, but it has a much smaller variation than Lee and Lee. The K-binary, that uses fewer bins than the algorithm proposed by Epstein, and a slightly more bins than the algorithm proposed by Lee and Lee. At the same time, the K-Binary algorithm works better than both previously proposed algorithms.

ACKNOWLEDGEMENT

We would like to thank Mrs. Sherly Prakash and Mrs. Jilsa Ali of Calicut University, Malappuram for their valuable suggestions and advices. Also we would like to thank the reviewers for the improvement of my paper.

REFERENCE

[1] Yusen Li., Xueyan Tang., Wentong Cai On Dynamic Bin Packing for Resource Allocation in the Cloud.

[2] E.G. Coffman Jr., M.R. Garey, D.S. Johnson, Dynamic bin packing, SIAM J. Comput. 12 (2) (1983) 227–258.

[3] E. G. Coffman, J. Csirik, G. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: Survey and classification. Handbook of Combinatorial Optimization (second ed.), pages 455–531, 2013.

[4] Andreas Wolke , , Boldbaatar TsendAyush , Carl Pfeiffer , Martin Bichler Dynamic resource allocation strategies in cloud data centers Volume 52, August–September 2015, Pages 83–95

[5] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. Approximation Algorithms for NP-hard Problems, pages 46–93, 2007.

[6] Ye Hu, Johnny Wong, Gabriel Iszlai and Marin Litoiu Resource Provisioning for Cloud Computing 2nd International Conference on Autonomic Computing, Seattle, 2005.

[7] Swathi Venigella, Dr. Wolfgang Bein Cloud Storage and Online Bin Packing Department of Computer Science University of Nevada, Las Vegas.2012

[8] S.Suganya, P.RajiPriyadharshini, A.Mas anam Resource Provisioning in Cloud Computing Based on Virtual Machines International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 3, March 2014

[9] www.wikipedia.com

[10] Gaikai. http://www.gaikai.com/.

[11] Onlive. http://www.onlive.com/.

International Journal of Computer Science and Engineering Communications,
Volume.5, Issue.3 (2017): Page.1627-1632
www.ijcsec.com

ABOUT AUTHORS

	**Sneha T Subrahmanian** Received B.Tech (Computer Science & Engineering) Degree from Kerala University, Trivandrum, Kerala and now pursuing M.E(Computer Science and Engineering) in Maharaja Institute of Technology, Coimbatore, Tamilnadu, India. Interested in Mobile Computing, Networking and Cloud Computing
	**Miji K Raju** Pursuing M.E(Computer Science and Engineering) in Maharaja Institute of Technology. Have keen interest in the Artificial Neural Network, Data Mining, Big Data, Mobile Computing .Done projects in the Data Mining and GSM Technology. 2 years of Teaching experience in India.
	**T. Sivakumar** Head of the Department and Assistant Professor in Maharaja Institute of Technology. Have keen interest in the Data Mining. Done projects in Data Mining, Data Source and Data Warehousing. 10 years of Teaching experiences in India.