International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Variable Size Bin Packing Algorithm for IoT

Kshitija Kalaskar

PICT, Savitribai Phule Pune University, Pune, Maharashtra, India

Abstract: ***In the era of bigdata and IoT, large amount of data is generated, that ranges upto some petabytes, i.e. Bigdata. Everyone using a machine spends a lot of time acquiring and storing data. Also small and large enterprises use and generate large amount business data which needs to be stored. Finding enough storage space to hold all the acquired data requires larger storage servers or external storage devices or deletion of older files. Deleting older files may result in loss of important data for organizations. Data in IoT are of different varieties i.e. they are unstructured and hence available in different sizes as well as possess different attributes and are required for knowledge generation. These are required to be packed into bins of defined sizes such that minimum number of bins is used to pack maximum data, in order to reduce storage cost and ease of data access and retrieval, by designing appropriate algorithm. Hence, we design a variable-size bin-packing algorithm for data generated from IoT.***

Keywords: Bin-packing, IoT, Data Descriptors, Data Ladder

1. Introduction

In bin packing, a sequence of fixed size data items are received, each must be assigned to a bin and sum of their sizes must not exceed the bin capacity, minimizing the bins used. Variable-size bin packing problem minimizes the sum of capacities of bins used. Variable-size bin packing problem differs from the classical one: bins do not have constant capacity, but set of heterogeneous capacities. The allocation is made considering capacity constraints, availabilities of the bins and arrival times of the data items, different costs and the processing times of the bins and deadlines of the items. IoT (Internet of Things) is a network of objects that sense, communicate and share information over an IP network. Regular data collection and analysis of the data acquired is used to initiate actions providing a wealth of information for planning, management and decision making. Data that is generated from IoT will be packed into bins of different sizes.

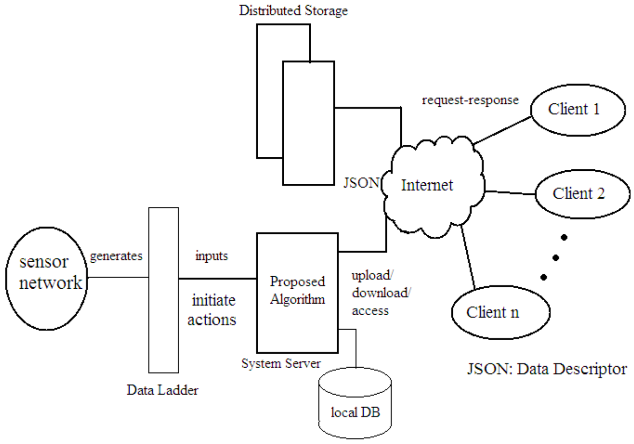
2. Motivational Survey

Papers [1], [3] consider the different ways in which each bin can be packed for completion, [5] Uses fitness function without using any GA operator, domain-specific knowledge. Resource allocation for overload avoidance and green computing [2], allocation of tasks to identical processors [13], additional capacity information for better packing, multi-resource allocation and scheduling solutions [15], allocating memory of servers for online requests, selecting minimum servers minimizing total cost [8], decision making for information that arrives in parts [11]. For bin packing, uncertain volumes and capacities [12], variable bin size, cost of packing and the possibility of splitting items [7] and heterogeneous bins of different volumes and fixed selection costs at minimum total bin-selection cost [16] are considered. Hyper boxes are packed in d-dimensional heterogeneous bins to minimize the total volume of bins [17]. Generalizing the square packing problem and the graph colouring problem, studying multi-dimensional version of the bin packing problem with conflicts [14]. [6] Applies the relative worst-order ratio to online algorithms. Authors of [4] propose parallel approximation BPA, results similar to FFD heuristic, complexity of O (logn). [10],[16],[17],[19] use bounded space bin packing problem. Authors of [20],[23] document the research trends in IoT, vision, challenges, usage and building blocks using RFID, IP stacks and web servers for smart objects for social and governance issues, [24] presents the Cloud centric vision. Authors of [18] propose architecture of smart hospital overcoming disadvantages of hospital information systems: fixed information point, inflexible networking mode, etc., [21] solutions for food supply chain (FSC) and in-home healthcare (IHH) and challenges, [19] adapting unified data spaces and semantics for design and implementation of MIDIS (Multi-Intelligence Data Integration Services) uses Big Data technologies with ontological models, semantic-based analysis services and [22] recent developments for semantic technologies like information modelling, ontologies and semantic data processing and challenges.

3. Proposed System

For large organizations which require large amount of storage, there is a need to utilize the existing storage well so that excess of storage is not provisioned due to improper data (object) placement in bins. For this, we design the variable size bin packing algorithm for data generated via IoT.

3.1 System Architecture



Figure

Licensed Under Creative Commons Attribution CC BY

The Sensor network is responsible for generating data in IoT. The data ladder is the container of the data coming at regularly intervals from IoT. It is a means of storing and forwarding the data as input to the system server in order to initiate the actions. System Server is the main processing unit that performs system operations such as generation of data descriptors, serving the client requests by giving appropriate responses, providing access to the data by access, storage, upload and download functionality. Also, the main proposed algorithm is executed on this server. This processed data is stored on the Local DB and transferred to the distributed storage via internet using JSON. The Android application is the client through which the end users send requests and receive responses in the form of JSON file to and from the system server.

3.2 Mathematical Modelling

Let, S : y = f(x) be the proposed system. Let 'S' be the programmer's perspective to the system such that: S ={s, e, Y, X, f_s, f_f, mem_s, DD, NDD, ...|ϕ} where, s is the start of the system. Start includes the initialization of the system by verifying the readiness of input data and functions.

e is the end of the system say, allocation of object sequence to different buckets for efficient storage utilization and exit.
Y is the output of the system i.e. the sequence (order) of objects to be placed in each bin.

Let, Y = {B_1 = {O_1, O_2, O_3... O_n}, B_2 = {O_1, O_2, O_3, ...O_m}, ...B_x = {O_1,O_2, ...O_p}} where, B_i є B and O_k є O.
X is the input to the system i.e. the objects and their sizes, bins and their capacity, etc.
Let, X = {B, O, C, size, Q} such that,

B is the set of bins available, where B = {B_1, B_2, B_3, ...B_x} and C is the capacity of each bin such that C is the set of different capacities of the bins. where, $B_i \rightarrow C_j$ i.e. each bin has a capacity C_j such that, C_j є C, i.e. each capacity belongs to some capacity class i.e. C є Cap.

Here, we say capacity class as we are using variable size bin-packing algorithm. We are sorting the capacities in order to place the like capacities into one capacity class. Therefore, there will be number of capacity classes, Cap = {CC_1, CC_2, CC_3, ...CC_n}, depending upon the sorted capacities. Best case will be one to one relation between the bin and the capacity class, B_i є CC_k, i.e. only one bin Bi with capacity C_j in capacity class CC_k which will avoid the searching and the complexity of search will be improved to O(1).

O is set the objects of data to be stored in the bins, where each object has some size such that $O_k \rightarrow$ size.

Q_c is the number of bins in each capacity class. Q_c decides the number of objects to be placed in the bins such that, size(O) = C_j*Q_c, where size(O) is the summation of sizes of all the objects and C_{sum} is the total capacities of all the bins i.e. C_j*Q_c. But, this is the best case. This wont be true in most of the cases as the object sizes and their arrangement in bins for best case performance results, i.e. placement of more number of objects in less number of bins, needs more parameters to be added. f_{pack} is the system function which is the bin packing function, responsible for placing the objects into the respective bin(s). The bin packing algorithm will be executed on each capacity class as if it was a classic bin-packing algorithm.

$$\text{Let, } f_{pack}: X \rightarrow Y .$$

The benefit of executing f_{pack} on each capacity class as a classic bin packing algorithm will help in exploiting the multi-core feature of the algorithm. This can be achieved by creating morphs of the system function f_{pack} and executing it on the different cores of the server machine for different capacity classes.

It also needs some supporting functions which are obtained from the set ff. ff is set of the friend functions for f_{pack} say, sorting function for objects is one of the friend of f_{pack}. Another one will be of classifying the buckets according to their size and putting them into particular capacity class.

Let, f_{f1}: sort(O, asc), be the function which sorts the set of objects in the ascending order.

Let, f_{f2}: sort(O, desc), be the function which sorts the set of objects in the descending order.

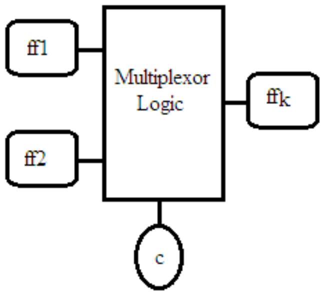


Figure 2: Multiplexor Logic

Functions f_{f1}, f_{f2} can be executed on different cores by selecting one of the two and creating morphs of the selected friend function. The time complexity of sorting will hence be reduced to O(1). The selection process can be done using multiplexor logic given in the figure below.

Let, f_{f3}: $B_i \leftarrow$ search(B, C, O_k, size(O_k)), be the function to search for the bin B_i є B in which we want to place the object O_k having some size, and this function returns bin B_i which satisfies the constraints $ϕ_1$, $ϕ_2$, $ϕ_3$, etc.

Let, f_{f4}: [B_i, C_j] $\rightarrow$ Cap, be the friend function which gives the capacity class of bin B_i. mem_s is the shared memory i.e. the bin B_i.

DD is the deterministic data i.e. the objects to be stored in to the bins. DD represents the objects and their size associated with each of them and the bin capacity with their capacity class is a part of DD. NDD is the non-deterministic data i.e. the sequence of the objects coming from the user which is unknown and the placement of objects in a particular bin of a particular capacity class. The objects are created, inserted, accessed by users. They can be created randomly with no order defined. Hence this kind of unknown data is said to be non-deterministic. NDD can only be determined at runtime, and hence every time the algorithm needs to run in order to

Licensed Under Creative Commons Attribution CC BY

place the different objects in given bins at different time instances.

ϕ is the set of constraints on the system.

For placement of each object O_k in bin B_i, we need to consider the following constraints,

ϕ_1 = size(O_k) <= $C_j(B_i)$, if O_k is the first object to be placed in the bin B_i.

ϕ_2 = size(O_k) <= ($C_j(B_i)$-size(O_j)), if O_k is the second object to be placed in bin B_i.

ϕ_3 = size(O_k) <= ($C_j(B_i)$-size(O_p)), where O_p is the set of already placed objects in the bin B_i.

Thus, the system S can be represented as:

$S = \{s, e, Y = \{B_1 = \{O_1, O_2, ...O_n\}, B_2 = \{O_1, O_2, O_3, ...O_m\}, ...B_x = \{O_1, O_2, O_p\}\}, X = \{B, O, C, size, Q\}, f_{pack}, f_{f1}, f_{f2}, f_{f3}, f_{f4}, mem_s, \phi_1, \phi_2, \phi_3\}$

The serialized execution of the functions will be as follows:

$f_{f1}/f_{f2} \rightarrow f_{f4} \rightarrow f_{f3} \rightarrow f_{pack}$

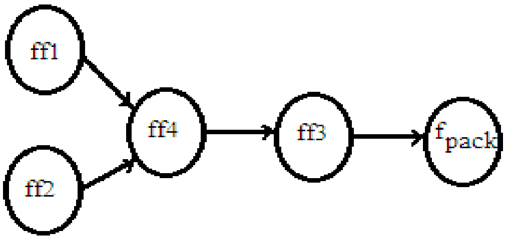


Figure 3: Task State Diagram

Success: All the objects are placed in the limited number of bins, hence giving the sequence of objects to be stored in each bin of each capacity class.

Failure: Error or exception caused forcing the system to exit.

3.3 Data tables and discussion

Table 1: Results

Inputs	Outputs	
Data Descriptor size(O)	Bin Packing Algorithm	Proposed Algorithm
10	8	17
15	15	31
18	36	54
28	96	108
30	158	194
35	222	249
40	347	389
50	513	578
61	627	767
86	796	853
98	1054	1310

The proposed algorithm is tested using the above research methodology comparing its results with the classic bin packing algorithm. The results in table 1, give the values of calculated for the existing bin packing algorithm in accordance with the proposed solution.

4. Conclusion

The performance graph in figure 4, displays the performances of the bin packing algorithm and the proposed algorithm. The number of data objects is at the Y axis and the number of bins at the X axis. The performance of the proposed algorithm is better than the existing BPA. A constant rise in performance can be noted right from the beginning when the number of objects is the least till the maximum number of objects is reached.

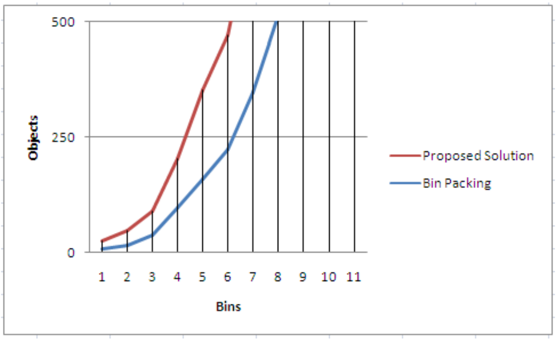


Figure 4: Performance graph

References

[1] Richard E. Korf, "A New Algorithm for Optimal Bin Packing", AAAI-02 Proceedings, 2002.

[2] Weijia Song, Zhen Xiao, Qi Chen, and Haipeng Luo, "Adaptive Resource Provisioning for the Cloud Using Online Bin Packing", IEEE TRANSACTIONS ON COMPUTERS, VOL. X, NO. X, X 201X

[3] Ethan L. Schreiber and Richard E. Korf, "Improved Bin Completion for Optimal Bin Packing and Number Partitioning", Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.

[4] Richard J. Anderson, "Parallel Approximation for Bin Packing", Information and Computation, Academic Press, Newyork and London, Vol. 82, No. 3, September 1989.

[5] Sami Khuri, Martin Schutz, Jorg Heitkotter, "Evolutionary Heuristics fro the Bin Packing Problem", ICANNGA '95, Apr 18-21.

[6] Leah Epstein, Lene M. Favrholdt, and Jens S. Kohrt, "Comparing Online Algorithms for Bin Packing Problems".

[7] Doina Bein, Wolfgang Bein, and Swathi Venigella, "Cloud Storage and Online Bin Packing".

[8] Teodor Gabriel Crainic, Guido Perboli,Walter Rei, Roberto Tadei, "Effcient Heuristics for the Variable Size Bin Packing Problem with Fixed Costs", Crainic, Perboli, Rei, Tadei and CIRRELT, 2010.

[9] Rob van Stee, "On-line scheduling and bin packing", ISBN 90-77017-60-7, Freiburg, 2002.

[10] Jin Peng, Bo Zhang, "Bin Packing Problem with Uncertain Volumes and Capacities", orcs.edu.cn

[11] Alejandro Masrur, Sebastian Dr Fiossler, Georg Farber, "An offline variable-size bin packing for EDF".

Licensed Under Creative Commons Attribution CC BY

[12] Leah Epstein, Asaf Levin, and Rob van Stee, "Multi-dimensional packing with conflicts", Alexander von Humboldt Foundation.
[13] William Leinberger, George Karypis, Vipin Kumar, "Multi-Capacity Bin Packing Algorithms with Applications to Job Scheduling under Multiple Constraints", TR 99-024: Draft May 27, 1999, ICPP'99.
[14] Teodor Gabriel Crainic, Guido Perboli, Walter Rei, Roberto Tadei, "Effcient Lower Bounds and Heuristics for the Variable Cost and Size Bin Packing Problem", Crainic, Perboli, Rei, Tadei and CIRRELT, 2010.
[15] Leah Epstein, Rob van Stee, "On Variable-Sized Multidimensional Packing", Netherlands Organization for Scienti_c Research (NWO), project number SION 612-061-000.
[16] Lei Yu, Yang Lu , XiaoJuan Zhu, "Smart Hospital based on Internet of Things", JOURNAL OF NETWORKS, VOL. 7, NO. 10, ACADEMY PUBLISHER, 10.4304/jnw.7.10.1654-1661, OCTOBER 2012.
[17] Anne-Claire Boury-Brisset, "Managing Semantic Big Data for Intelligence", STIDS 2013 Proceedings Page 41-47.
[18] Dlodlo N, Foko T, Mvelase P, and Mathaba S "The State of Affairs in Internet of Things Research", The Electronic Journal Information Systems Evaluation Volume 15 Issue 3 2012, (244- 258).
[19] Zhibo Pang, "Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well- being", ISBN 978-91-7501-736-5, December 2012.
[20] PAYAM BARNAGHI, WEI WANG, CORY HENSON AND KERRY TAYLOR, "Semantics for the Internet of Things: early progress and back to the future".
[21] Friedemann Mattern and Christian Floerkemeier, "From the Internet of Computers to the Internet of Things", Karl Steinbuch, German computer science pioneer, 1966.
[22] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems 29 (2013) 1645-1660.
[23] C. Aggarwal, N. Ashish, and A. Sheth, "The Internet of Things: A Survey from The Data-Centric Perspective", Book Chapter in "Managing and Mining Sensor Data", Springer, 2013.
[24] Karl A. Hribernik, Carl Hans, Christoph Kramer, Klaus-Dieter Thoben, "A Service-oriented, Semantic Approach to Data Integration for an Internet of Things Supporting Autonomous Cooperating Logistics Processes", BIBA-Bremer Institut fr Produktion und Logistik GmbH, Germany.

Author Profile

Kshitija Kalaskar, ME Computer, PICT, Savitribai Phule Pune University, Pune

Licensed Under Creative Commons Attribution CC BY