Big Data Aware Virtual Machine Placement in Cloud Data Centers

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Logan Hall*, Bryan Harris, Erica Tomes, Nihat Altiparmak

Computer Engineering & Computer Science Department

University of Louisville

Now at UT Austin, Comp. Eng. Dept.

12/8/2017

Outline

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- Motivation
- Big Data Aware VM Placement
  - Problem Description
  - Problem Formulation
  - Low-cost Heuristics
- Evaluation
  - Bottleneck Analysis
  - Experimental Setup
  - Experimental Results
- Conclusion

Motivation

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- Cloud computing offers scalable big data storage and processing opportunities for academia and industry [1, 2]
- Cloud computing has two building blocks:
  - **Virtualization**
    - For increased computer resource utilization, efficiency, and scalability
  - **Data Replication**
    - For scalability, availability, and reliability
- Datasets are divided into equal size disjoint chunks (~128 MB), chunks are replicated (~3 replicas), distributed over clusters within a datacenter or geographically across multiple datacenters, and retrieved/processed by Virtual Machines (VMs) or tasks scheduled on Physical Machines (PMs)

Motivation

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Since data to be processed is very large, a common approach in Big Data processing is to send the computation (VM) to data (PM) and to retrieve data locally.

- This assumes that network bandwidth is always lower than storage throughput
  - Existing high speed networking interconnects (10/40/100 Gbps) can provide transfer bandwidth higher than the storage throughput of HDDs, sometimes even better than new generation NVMe devices, and can make the storage subsystem the cause of the bottleneck [3, 4].
  - **Therefore, both network and storage can be the cause of the bottleneck in data retrieval!**
- Also, local data access might not be always feasible since:
  - PMs have limited resources (processor, memory, etc.)
    - VMs' resource requirements might not be satisfied by the PMs holding their data
  - All data of a VM might not reside in a single PM
    - One VM might need to process multiple data chunks residing on different PMs

Motivation

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- The completion time of distributed big data processing applications is highly affected by data access bottlenecks that can lie both in storage and networking subsystems.
- Efficient Big Data processing in the Cloud requires a Virtual Machine (VM) placement techniques that is aware of:
  - ✓ *VM resource requirements and PM resource capacities*
  - ✓ *Data replication and replica locations*
  - ✓ *Performance of the storage subsystem in individual PMs (disk I/O throughput)*
  - ✓ *Available network bandwidth between the PMs*

Problem Description

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

We are given:

- A set of virtual machines $VM_1, VM_2, ..., VM_M$ with resource demands (CPU cores, memory, etc.)
- A set of physical machines $PM_1, PM_2, ..., PM_N$ with resource capacities
- Data requirements of the VMs
  - Every VM_j requires a set of data chunks $D_1, D_2, ..., D_{Qj}$ to be retrieved from the PMs, where every chunk is replicated on multiple (r) PMs.

In Big Data Aware VM Placement (BDP), our aim is *minimizing* the retrieval time of *all data chunks* by specifying:

- The placement of the VMs over the PMs
- Retrieval schedule of all data chunks (replica selection)

Problem Formulation

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- BDP can be formulated and optimally solved using linear programming techniques as follows:

Minimize: R

Subject to:

$$\sum_{k \in P_{ij}} \sum_{l=1}^{N} B_{ijkl} = 1; \quad i = 1, \ldots, Q_j; \quad j = 1, \ldots, M$$

$$\sum_{i=1}^{Q_j} \sum_{k \in P_{ij}} B_{ijkl} = Q_j \cdot I_{jl}; \quad j = 1, \ldots, M; \quad l = 1, \ldots, N$$

$$U_{lt} \leq C_{lt}; \quad l = 1, \ldots, N; \quad t = 1, \ldots, T$$

$$R - R_k \geq 0; \quad k = 1, \ldots, N$$

- This is a mixed integer programming formulation, which is classified as NP-hard [5]. We will use this optimal solution for comparison purposes, but we also propose low-cost heuristics.

Low-cost Heuristics: *bdp*

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Best-Data VM Placement (*bdp*) (shown as Alg.1 in the paper)

- Aims to place VMs on the PMs in a greedy fashion depending on which PM yields the best overall retrieval time
  - Considers previous VM placements and their requests, network bottlenecks, and storage bottlenecks
- First sorts the VMs in ascending order of the **data requirements** by the VM (to achieve a balanced data retrieval load across the PMs)
- Then for every VM, the heuristic iterates through every PM and checks its compatibility based on VM resource requirements. If the PM is compatible, it hypothetically places the VM on that PM, and also selects replicas using a greedy retrieval technique (shown as Function 2).
  - The idea is to consider a data retrieval cost for each PM as in the LP formulation, but to update the PM loads in a greedy manner based on local optimal values for each VM
  - The hypothetical placement that yields the minimum data retrieval cost is then selected for the placement of the VM

Low-cost Heuristics: *ff-data*

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

First Fit Data (*ff-data*) (shown as Alg.2 in the paper)

- The motivation behind *ff-data* is to achieve a better fitness in VM placement that reduces the total number of PMs used, thus yielding a **reduced energy consumption**.
- In addition, our aim is to propose an alternative heuristic to *bdp* and evaluate their performance in both energy consumption and data retrieval.
- As with *bdp*, *ff-data* also starts by sorting VMs; however, the sorting is performed here in decreasing order by **resource requirements** of the VMs so that the VMs with the largest resource requirements are placed first, as there may be a limited number of compatible PMs.
- Next, for every VM, the first compatible PM is determined as the placement. Then, replicas are selected using the greedy retrieval technique as in *bdp* based on local optimals.
- *ff-data has a slightly lower time complexity than bdp (details in paper)*

Evaluation: Bottleneck Analysis

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- Data transfer between two PMs is expected to be governed by the bottleneck of two important properties of a distributed system:
  1. Local storage system throughput for the source PM
  2. Network bandwidth between the source and the destination PMs
- In order to validate this, we performed a set of experiments:

Storage System	Local Read Times			
	1 chunk	2 chunks	4 chunks	8 chunks
4 SSDs, RAID-10	155ms	287ms	566ms	1.13s
4 HDDs, RAID-10	431ms	848ms	1.68s	3.27s
1 SSD	394ms	779ms	1.54s	3.09s
1 HDD	825ms	1.62s	3.24s	6.41s
Storage System	**Remote Read Times via 1 Gbps Network**			
	1 chunk	2 chunks	4 chunks	8 chunks
4 SSDs, RAID-10	1.37s	2.61s	5.20s	10.2s
4 HDDs, RAID-10	1.35s	2.63s	5.42s	10.2s
1 SSD	1.37s	2.60s	5.15s	10.2s
1 HDD	1.35s	2.61s	5.16s	10.2s
Storage System	**Remote Read Times via 100 Mbps Network**			
	1 chunk	2 chunks	4 chunks	8 chunks
4 SSDs, RAID-10	11.5s	22.9s	46.1s	91.3s
4 HDDs, RAID-10	11.5s	22.9s	46.1s	91.3s
1 SSD	11.5s	22.9s	46.1s	91.3s
1 HDD	11.5s	22.9s	46.1s	91.3s

- These experiments emphasize the importance of bottleneck analysis in Big Data transfer, where both storage throughput and network bandwidth play an important role.

Evaluation: Experimental Setup

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- Performed simulations supported by real data transfer times (*from the table*)
- Used three different network configurations: (i) 1 Gbps homogeneous, (ii) 10 Gbps homogeneous, and (iii) 1/10 Gbps heterogeneous (mixed).
  - In homogeneous networks, all links have the same transfer rate, but in heterogeneous networks, the link rates are randomly selected between 1 Gbps and 10 Gbps.
- Used four storage configurations: (i) 1-HDD homogeneous, (ii) 1-SSD homogeneous, (iii) 4-SSDs homogeneous, and (iv) heterogeneous (mixed).
  - In the homogeneous storage scenarios, all PMs have the same storage system; in the heterogeneous scenario, storage systems of the PMs are randomly selected from the 1-HDD, 1-SSD, and 4-SSDs cases.
- Used two resource types, CPU cores and memory, and used the following Amazon EC2 instances [6] to determine our VM resource requirements:
  i. t2.small (1 CPU Core, 2 GB Memory)
  ii. t2.medium (2 CPU Cores, 4 GB Memory)
  iii. t2.large (2 CPU Cores, 8 GB Memory)
  iv. t2.xlarge (4 CPU Cores, 16 GB Memory).
- PM capacities are randomly selected and results were averaged over 100 runs.

Evaluation: Experimental Setup

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

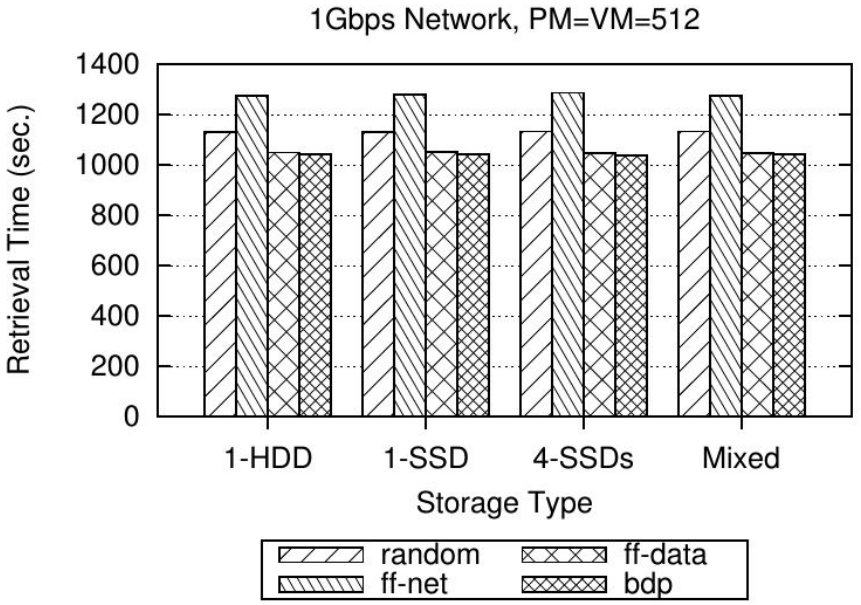
- Implemented the following algorithms:
  - ***random*** places VMs on randomly selected PMs. Local replicas are selected if available; otherwise, replicas are also selected randomly.
  - ***ff-net*** uses a first-fit decreasing strategy to place VMs on PMs [7], and it follows an HDFS-like network-aware replica selection strategy [8], where if a local replica exists, the data is retrieved locally; otherwise, it selects a replica from the PM with the smallest network transfer time to the host machine. If a tie occurs for the nearest replica, then the tie is broken randomly.
  - ***ff-data*** also uses a first-fit decreasing strategy to place VMs on PMs; however, it uses a greedy replica selection that considers the retrieval cost of selecting the replica from each source PM. The source chosen is the one with the lowest retrieval cost considering the machine load and transfer time.
  - ***bdp*** uses a greedy strategy for placing VMs on PMs; all PMs that satisfy VM requirements are considered for placement. Greedy replica selection is performed for each PM candidate and the PM placement that leads to the minimum total data retrieval time out of all PMs (local optimal) is chosen.
  - ***optimal*** implements the LP formulation and guarantees the optimal data retrieval time

Evaluation: Experimental Results

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Data Retrieval Perf., 512 VMs and PMs, 1 Gbps Homo. Network

- **Network is the bottleneck**!
- ***ff-net*** takes ~140 sec more than even ***random*** to retrieve the entire dataset
  - The reason is *tight fitness* and *poor replica selection*; ***ff-net*** prefers nearest replicas and generates bottlenecks in the PMs holding these replicas.
  - ***random*** yields a more uniform distribution over the PMs for both VM placement and replica selection.
- Both ***bdp*** and ***ff-data*** consistently perform better than the others since they balance the load on the PMs better.
  - ***Each VM retrieves ~100 GB***
  - ***~50 TB is retrieved in total***



(a) 1 Gbps Homogeneous Network

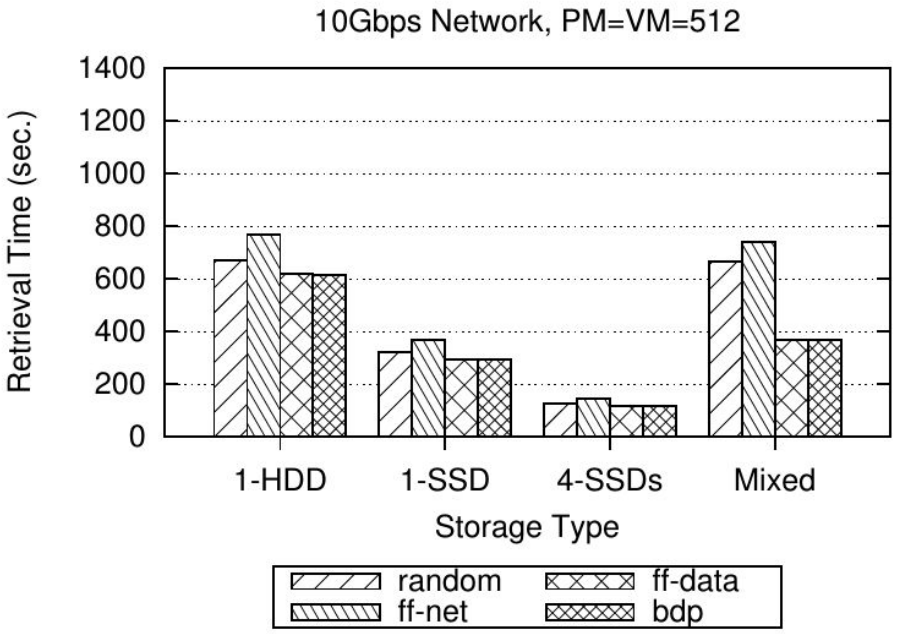
- ***bdp*** retrieves the dataset 9 seconds faster than ***ff-data***

Evaluation: Experimental Results

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Data Retrieval Perf., 512 VMs and PMs, 10 Gbps Homo. Network

- **Storage is the bottleneck**!
- The gap between ***random*** and ***ff-net*** narrows, but ***random*** still performs better due to the same reason as in the 1 Gbps case.
- For the faster storage system, the performance gap between ***random*** and ***ff-net*** is the smallest outlining the storage bottleneck ***ff-net*** experiences.
- The proposed ***ff-data*** and ***bdp*** heuristics again outperform the others since they are aware of storage bottlenecks in this case and they are able to retrieve replicas accordingly.
  - ***Each VM retrieves ~100 GB***
  - ***~50 TB is retrieved in total***



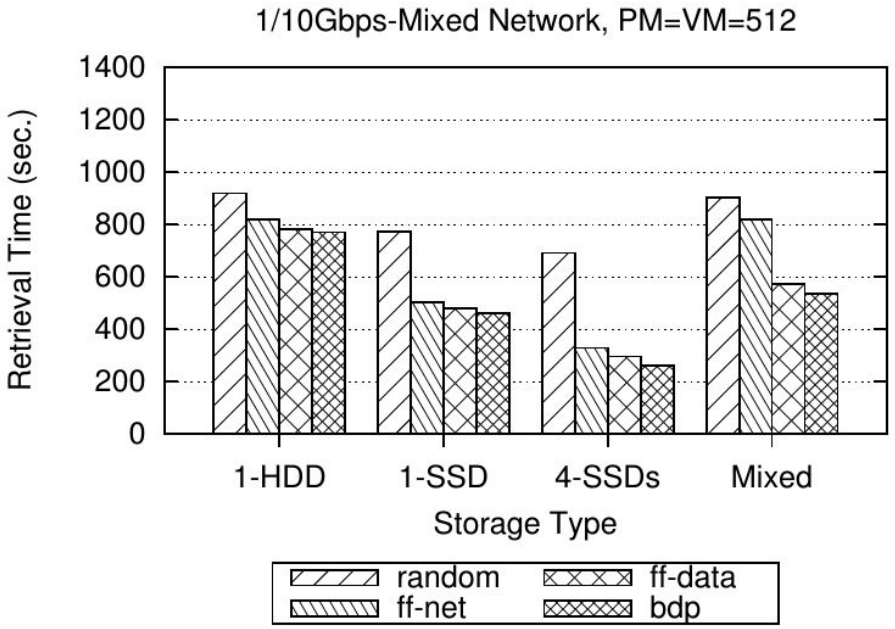
(b) 10 Gbps Homogeneous Network

Evaluation: Experimental Results

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Data Retrieval Perf., 512 VMs and PMs, 1/10 Gbps Het. Network

- **Mixed bottlenecks in storage and network**!
- ***ff-net*** passes ***random*** in performance, especially when the storage is faster since ***ff-net*** is network-aware and able to select better network links in retrieval compared to ***random***.
- The proposed ***ff-data*** and ***bdp*** heuristics still outperform both ***random*** and ***ff-net***
- Performance difference between ***bdp*** and ***ff-data*** becomes even larger (up to 36 sec.) in this heterogeneous case.
  - ***Each VM retrieves ~100 GB***
  - ***~50 TB is retrieved in total***



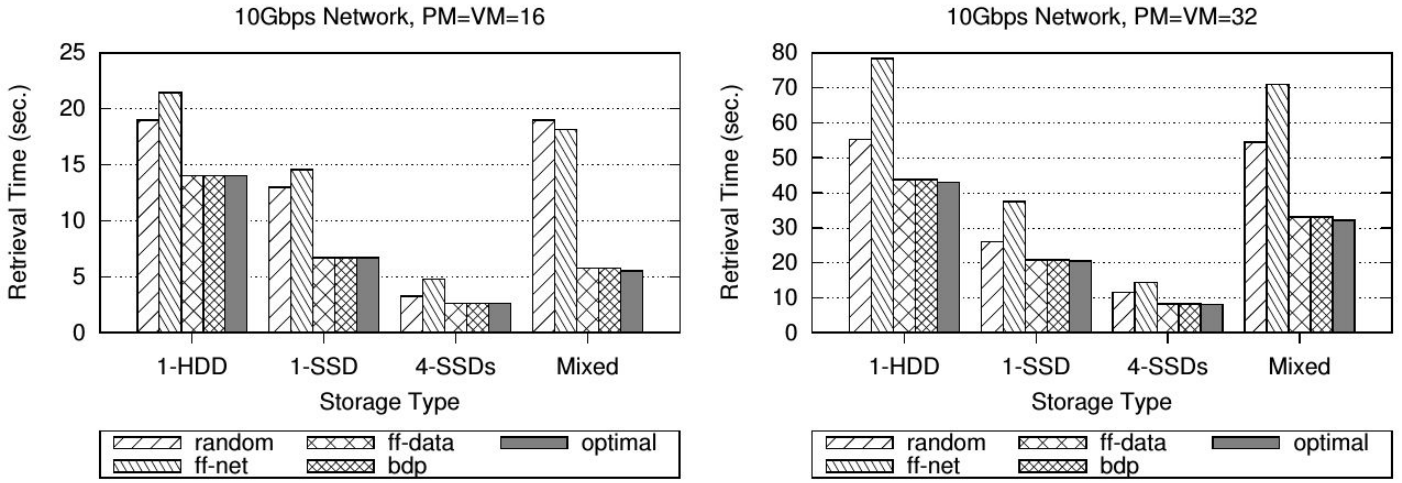
(c) 1/10 Gbps Heterogeneous Network

Evaluation: Experimental Results

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Data Retrieval Perf. compared with the ***optimal*** values:

- 16 and 32 VMs and PMs, 10 Gbps Homo. Network



(a) M = N = 16 (b) M = N = 32

- In three out of eight storage configurations, the proposed heuristics (***ff-net*** and ***bdp***) achieved the optimal data retrieval value, and in the other five configurations, their performance was within 5% of ***optimal***.

Evaluation: Experimental Results

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

We also evaluated the **energy efficiency** of the proposed algorithms by comparing the number of PMs used, graphs are in the paper. In summary:

- ***random*** achieves the worst performance by using the most number of PMs in the placement in all cases.
- First-fit based VM placement heuristics ***ff-net*** and ***ff-data*** both achieve the same energy efficiency as being slightly better than ***bdp*** for the 1 Gbps homogeneous network and 1/10 Gbps heterogeneous network cases
- ***bdp*** achieves the best energy efficiency in the 10 Gbps homogeneous network case, where the storage system is the cause of the bottleneck. This is mainly due to the fact that ***bdp*** places VMs over PMs that are closest to each other (around the PMs with fastest storage devices) and therefore achieves a very tight fit.
- As also discussed by Ananthanarayanan et al. [3], with the availability of 40 and 100 Gbps network bandwidths in today's clusters, the storage system generally becomes the main source of the bottleneck in data transfers. Our 10 Gbps network configuration is a good representation of this case, where the proposed bdp algorithm consistently achieves the best performance in both data retrieval and energy efficiency!

Conclusion

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

- We formally defined and formulated Big Data aware virtual machine Placement (BDP) problem and solved it using linear programming techniques.
- In addition, two low-cost heuristics (***ff-data*** and ***bdp***) were proposed for efficient big data processing in the cloud that considering the data retrieval time of large datasets and energy consumption of the cloud infrastructures.
- In our evaluation, the proposed ***ff-data*** and ***bdp*** heuristics achieved a data retrieval performance within 5% of the optimal data retrieval value.
- Furthermore, both data retrieval time and energy efficiency of the proposed ***bdp*** heuristic outperformed other VM placement heuristics in the cases where the storage subsystem was the cause of the bottleneck in data transfer.
- As high-speed networking interconnects of 10/40/100 Gbps become more common in private clusters and cloud infrastructures, storage throughput generally cannot keep up with the available network bandwidth. Therefore, we believe that the proposed heuristics can provide a tremendous value for big data processing in the cloud by reducing both data analysis times and energy consumption.

Thank You!

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

Questions?

References

UNIVERSITY OF LOUISVILLE
J.B. SPEED SCHOOL OF ENGINEERING

[1] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing. Inf. Syst., 47(C):98–115, January 2015.

[2] Domenico Talia. Clouds for scalable big data analytics. Computer, 46(5):98–101, May 2013.

[3] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Disklocality in datacenter computing considered irrelevant. In Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems, HotOS'11, pages 12–12, Berkeley, CA, USA, 2011. USENIX Association.

[4] White Paper. NVMe SSD 960 PRO/EVO, December 2016.

[5] R M Karp. Reducibility among combinatorial problems. Complexity of Computer Computations, 40(4):85–103, 1972.

[6] Amazon. Amazon EC2 VM Instance Types, 2017. https://aws.amazon.com/ec2/instance-types/.

[7] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder. Heuristics for vector bin packing. January 2011.

[8] K. Shvachko, Hairong Kuang, S. Radia, and R. Chansler. The hadoop distributed le system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pages 1–10, May 2010.