6B-3

Performance Comparison of Power-Proportional Approaches in Power Saving of Storage Systems

Hieu Hanh LE† Satoshi HIKIDA† Haruo YOKOTA†

†Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, 2–12–1 Ookayama, Meguro, Tokyo, 152–8552 Japan

1 Introduction

Recently, achieving power-proportionality in datacenters has been gained more and more attention and considered as an important design factor [1]. Based on this idea, a number of researches have been proposed in order to provide this metric to disk-based storage systems [2], [3], [4]. The technique that supports these approaches is data replication which benefits the possibility of selecting a replica in current active disks rather than choosing another replica stored in a disk which is in sleep mode in storage systems.

However until now, the above approaches have still not been compared with each other in similar environment yet. Aiming to identify the good and also the weak points of each proposal, it is necessary to perform a performance comparison of so called methods. In this paper, we decide to choose two representative proposals, i.e. PARAID [2] and RABBIT [4], and perform empirical experiment on actual machines to compare their performance. While PARAID inspires many other researches by its idea of controlling system's power over small groups, RABBIT is a novel work adapted to HDFS (Hadoop Distributed File System) [5] which is very popular in distributed computing area. Here, the impact of data placement method to power-proportionality in these two approaches is evaluated relating to time consuming for completely reading certain dataset.

2 Data Placement

In this section, the data placement to achieve power-aware storage systems used in PARAID and RABBIT are described.

Not like RABBIT, PARAID was originally designed inside a RAID unit, however the idea can be expanded to distributed environment that contains a large number of nodes connected through network. In this context, a node is defined as an array of disks managed together with respect to energy control. Thus, a node is a collection of disks and there is no disk sharing between nodes.

Given a dataset D with total B blocks, a total number of nodes N are divided into G groups. Each group contains a different number of nodes. In detail, each node is symbolized as $n_{(g,i)}$, where g $(1 \leq g \leq G)$, i $(1 \leq i \leq N)$ indicate i-th of node at g-th group. E.g., nodes $n_{(1,1)}$, $n_{(1,2)}$ belong to Group 1, while nodes $n_{(2,3)}$, $n_{(2,4)}$ are in Group 2.

2.1 PARAID

At first, all data D of B blocks are allocated evenly to all nodes. We denote by $B_i(\frac{1}{m})$ an $\frac{1}{m}$ fraction of B_i. After replication, each node will hold a certain replica in addition to its original data as follows: (a) Each node in Group 1 nodes gets an *equal* fraction of the replicated data from each node of other groups; (b) Remaining nodes keep replicas of original data from specific other non-Group 1 nodes in *skewed* way. Specifically, the original data B_i of a non-Group 1 node $n_{(g,i)}(g > 2)$ are replicated equally to other non-Group 1 nodes. This is done by selecting $\frac{1}{i-1}$ blocks of B_i for each node $n_{(g,j)}$ $(j < i)$.

2.2 RABBIT

Supposedly r replicas of B blocks from dataset D are desired to be stored to n nodes with G group. At first, one replica of all B blocks are equally stored in first primary p nodes at Group 1. Consequently, each node in Group 1 contains $\frac{B}{p}$ blocks. The remaining $(r-1)$ replicas are distributed to $(N-p)$ nodes in the way that the node $n_{(g,i)}$, where $g > 2$ and $p < i <= N$, stores $\frac{B}{i}$ blocks. Here, in the constrain of keeping number of replica r small with fixed number of nodes RABBIT can guarantee that the number of blocks stored by i-th node must not be less than $\frac{B}{N}$ for all $i \leq N$ when N nodes are active. Obeying this constrain makes it possible for the load to be shared equally among active nodes.

Through above data placement, both PARAID and RABBIT can organize disks into certain gears and make the system be able to operate in different modes. For example, in Gear 1, with disk 1 and 2 are powered and disk 3, 4 and 5 can be powered off. Once load increases, the system implement up-shifts into second gear by powering up disk 3, 4 and so on. Leveraging these techniques, PARAID and RABBIT are con-

Performance Comparison of Power-Proportional Approaches in Power Saving of Storage Systems
Hieu Hanh LE†, Satoshi HIKIDA†, Haruo YOKOTA†
†Department of Computer Science, Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, 2–12–1 Ookayama, Meguro, Tokyo, 152–8552 Japan
hanhlh@de.cs.titech.ac.jp hikida@de.cs.titech.ac.jp
yokota@cs.titech.ac.jp

Table 1: Data placement

Method	Data Attribution	Group 1 nodes		Group 2 nodes		
		$n_{(1,1)}$	$n_{(1,2)}$	$n_{(2,3)}$	$n_{(2,4)}$	$n_{(2,5)}$
PARAID	Original	B_1	B_2	B_3	B_4	B_5
	Replica (Equal and Skew)	$B_3(\frac{1}{2})$	$B_3(\frac{1}{2})$			
		$B_4(\frac{1}{2})$	$B_4(\frac{1}{2})$	$B_4(\frac{1}{3})$		
		$B_5(\frac{1}{2})$	$B_5(\frac{1}{2})$	$B_5(\frac{1}{4})$	$B_5(\frac{1}{4})$	
RABBIT	Power proportionality	$\frac{B}{2}$	$\frac{B}{2}$	$\frac{B}{3}$	$\frac{B}{4}$	$\frac{B}{5}$

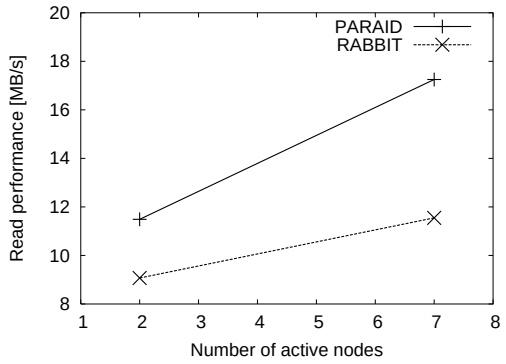

Figure 1: Read only performance comparison

sidered to be able to provide the power-proportional characteristic that performance is proportional with the power consumption of the system.

Table 1 gives an example of data placement of both methods for 5-node cluster storage system with 2 groups. Here, the system can operate in 2 gears, Gear 1 with Group 1 nodes are power on and Gear 2 with all nodes are active.

3 Experiments

In this section, the empirical experiments to compare complete read performance of two approaches is reported. We decided to implement the data placement of both approaches over HDFS. Our testbed consisted a server performing functions of a *namenode* and a rack containing a number of storing nodes which play roles of *datanodes* as in HDFS. Each storing node was designed for low power consumption, which is used as a node of autonomous disks and consisted Transmeta Efficeon TM8600 1.0 GHz processor, 512 MB DRAM memory, 250 GB disk (2.5 inch) with Linux 2.6.18 kernel, JDK-1.6.0 and HDFS's stable version 0.20.2. The block size was kept as default value in HDFS (64MB).

The 10 GB-dataset was at first written into the system and then was requested to be fully read by a client. The number of replica in RABBIT was fixed to 2. The number of active nodes was specified through command line and the memory cache was cleared between runs.

Figure 1 shows the performance result for reading all the storing dataset from system when the number of active nodes was set to 2 and 7. Here, it means that Gear 1 needs 2 active nodes while Gear 2 need all 7 nodes to be active. Note that in our case, RABBIT needs at least 7 nodes to fully store 2-replica 10 GB dataset. It can be seen from this results is that both two approaches were success in providing the power-proportionality to system as the read throughput was improved as the number of active nodes increased. In addition, PARAID gained better performance than RABBIT for both cases because of better load balancing between active nodes. Here, it is well recognized that the load balancing function in [4] is still not implemented yet. The further experiment to reconfirm these results is left as future work.

4 Conclusion

The empirical experiment to evaluate performance of PARAID and RABBIT was reported in this paper. Through the results, it is seen that both approaches were able to provide power-proportionality to systems. In the future, evaluations on power consuming and other performance metric with workloads would be considered.

Acknowledgement

This work is partly supported by Grants-in-Aid for Scientific Research from Japan Science and Technology Agency (A) (#22240005) and MEXT (#21013017).

References

[1] B.L. Andre, and H. Urs, "The Case for Energy-Proportional Computing," Computer, vol. 40, no. 12, pp. 33–37, 2007.

[2] W. Charles, O. Mathew, Q. Jin, W.A.I. Andy, R. Peter, and K. Geoff, "PARAID: A Gear-Shifting Power-Aware RAID," Trans. on Storage, vol. 3, 2007.

[3] B. Mao, D. Feng, H. Jiang, S. Wu, J. Chen, and L. Zeng, "GRAID: A Green RAID Storage Architecture with Improved Energy Efficiency and Reliability," in the Proc. IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems., pp. 1–8, 2008.

[4] A. Hrishikesh, C. James, G. Varun, G. Gregory R., K. Michael A., and S. Karsten, "Robust and Flexible Power-Proportional Storage," in the Proceeding of the 1st ACM Symposium on Cloud Computing, pp. 217–228, ACM, 2010.

[5] "Hadoop," `http://hadoop.apache.org`.