Energy Proportionality for Storage

Presenter: Manish Kumar
Opponent: Thaer Samar

Presented on: July 6th, 2011

This report tries to critically assess the strong and weak points of the paper titled “Energy Proportionality for Disk Storage Using Replication ” by Kim and Rotem. We first try to present positive aspect of their work and later part we give some critical analogy with other work done in the same area.

In the paper authors have tried to address the energy saving techniques without compromising the performance. They have tried to rationalize the evaluation close to real world scenario. They tried to show that as much as 90% of theoretical limit for energy saving is possible with their algorithm. Their work is motivated by the fact that energy saving and response time penalty conflicts when dealt with in isolation. Other thing they observed that expected length of inactivity of disk storage device are very small. They observed that servers disk storage systems typically takes one-third of total power consumption and servers using these disk systems for service are typically run at half of their full utilization level. This implies that energy saving techniques applied when server are idle would result in a significant power savings. This underlines the energy proportionality as a better design metric than simple energy saving. Though they emphasizes energy proportionality but they didn't show how their technique satisfies the energy proportionality principle and also there is no experiment to show the relation between energy consumed and the workload. So it was better to have some experiments about the proportionality especially this is not only a claim in the paper, but also it is the goal of the paper.

We will first enlist some good points about the working of FREP and its promising experimental result even if it was dealt in isolation and in later section we critically assess the implementation by comparing it with other similar implementations.

FREP model

It maintains a set of partitions of disks on which the energy management algorithm runs. It gives a better control in sense of partitioning data according to their uses and adjusting energy conserving models accordingly. FREP algorithm is run on per partition basis.

Next level of abstraction is on a disk array being treated as one node in terms of energy management. Typically this is a RAID array. While this hides inner disk details, advantages of data redundancy and fault tolerance come along for free.

FREP emphasizes on replication model for the system as it is used in most of the distributed file systems anyway for data redundancy as well as for better performance. So, the integration with existing system becomes easy and useful. Essentially two kinds of node namely, Covering set (CS) and non covering set nodes (non-CS) nodes, replicate data among themselves for performance as well as fault tolerance reason.

FREP replication strategy is simple and robust. It can be realized that cost of disk space per unit of data is quite affordable, so the replication cost of data is affordable. FREP maintains two kind of replication - (i) balanced replication (ii) skewed replication. CS nodes maintain among themselves disjoint copies of data blocks of non-CS node and non-CS node maintains disjoint copies of CS node data. Similarly, non-CS nodes also maintain data from other non-CS node also with nodes with lower index keeping data blocks from non-CS nodes with higher index. These two ways of replication gives good fault tolerance and a greater freedom of energy saving as CS-nodes which are always spinning, have data from all the nodes. So in the best case all non-CS nodes can be turned off and all CS node will be able to handle the request.

Looking carefully we observe that the load has to be shared by some nodes when a non-CS node is turned off. This is cumulatively shared by non-CS node. CS nodes are anyway serving the request. In order to facilitate this, FREP's skewed replication mechanism play a role. The good thing about this replication is that it allows non-CS nodes with lower index to be sent to standby without affecting the service. This replication mechanism though gives better data redundancy and hence greater fault tolerance, but also brings load imbalance among CS and non-CS nodes as there is more data on nodes with lower indices and data requests have to be served by other nodes when non-CS nodes are sent to standby for energy management purpose. These nodes tend to get overloaded on the long run. FREP uses probabilistic redirection beautifully to offload the request to active non-CS nodes. They also showed in the experiments that this fits really well with real world data also. When half of the configured nodes are active, every node has equal load and there is no imbalance of load on nodes. This is far better compared to the case when only CS nodes take the load. This mechanism is facilitated by FREP mapping table which keeps information about the replica on other nodes.

FREP gear shift mechanism

FREP has been designed with SLA in mind. It tries to maximize the energy conservation without penalizing the performance or compromising on SLA. The best part of FREP is the gear shift mechanism to dynamically handle load to meet SLA and at the same time maximize the energy savings. It uses De-Bruijn graph prediction mechanism to predict probability of not violating the SLA in some given observation period. This lies at the heart of FREP.

Configuring FREP with proper observation window gives a greater ability to adapt disk state according to load at run-time. All the previous similar implementations lacked this mechanism and hence their response to work load variations were very poor. Adaptability provided by the probabilistic method is huge gain on earlier implementation. De-Bruijn graphs tries to look into past to derive information for future state. It maintains the state transition probabilities which changes when an event corresponding to energy management occurs. Simply relying on prediction graphs sometime leads to unexpected latency. Though they tried to address this issue, but it is not perfect. They simply spin all disks up. As variations in work load increases, FREP mechanism makes lot of faults and energy savings goes for a toss. They accept this fact and emphasize that properly adjusting the observation window size can drastically reduce the number of fault. But this assumption is not entirely helpful also. Workload variation in real world is a prominent factor to be taken care of, and probability calculations by De-bruijn graph will invalidate in most of the case and all the disks will be spun up.

Experiments

They tried to capture two kind of workload scenario - a continuous workload where inter-arrival time of requests are small (umass logs) and other burst traffic workload(cello logs). Though this describes two scenario of workloads generally found, but in most of the cases workload attributes are a combination of both. They tried to show that FREP gear shift mechanism properly adjust their gears and energy saving is possible without SLA violation. The results are promising under tight SLA also and fare considerably well compared to other similar implementation like PARAID simply because of probabilistic prediction.

Their probabilistic model of De-Bruijn graphs works well to avoid performance penalty, but they have to give up on energy savings to avoid SLA violations. This is evident from the energy saving graph for umass-2 logs where there is continuous data request. Energy savings in these cases are only around 18 percent of no energy saving at all model. Though they tried to show energy saving in different FREP configuration, they actually didn't capture it for all of the work loads. They showed significant energy savings for cello workload for which inter-arrival rate of request were large. Though this test cases was highly idealized, still it showed the opportunities available for energy saving without penalizing the performance. Similarly they pointed out that in these scenarios, FREP redirection performance is better and there is almost zero performance penalty.

Critical assessment of FREP

Achieving energy proportionality in datacenters is much important than just energy saving, and the core principle behind energy proportionality is that computing equipment should consume power proportional to load level, such that if a computing equipment consumes x watts at full load, it should consume x.(p/100) when running at p-% load. In the paper they talked about this principle, but they didn't show how their technique satisfies the energy proportionality principle and also there is no experiment to show the relation between energy consumed and the workload. So it was better to have some experiments about the proportionality especially this is not only a claim in the paper, but also it is the goal of the paper.

The authors claimed that they will present a novel replication strategy, that achieves energy benefits while maintaining performance and fault tolerance, but actually they didn't come with something new. The data replication has been used by many data centers for fault tolerance and load balancing reasons, and this replication is the same as the balanced replication they are using, and for the skewed replication, the idea of skewed replication has been presented by Rabbit “Robust and flexible power-proportional storage”.

1. FREP vs. PARAID

PARAID uses skewed stripping pattern to adapt to the system load by varying the number of powered disks. The gear shift mechanism was introduced by the power aware RAID(PARAID) based on the system load as reconfiguration, the main design issue of PARAID is to show how to skew disk stripping to allow opportunities for energy saving , while preserving performance and reliability.

In the paper they mentioned that the main difference between their approach and PARAID is that PARAID spin up/down one or more disks in the array, but I see that, this is not a main difference, and not totally correct, since PARAID can vary the number of of powered on disks by gear shifting or switching among sets of disks to reduce energy saving by exploiting unused storage to replicate and stripe data in skewed fashion so that disks can be organized into hierarchical overlapping sets of RAID, each set contains a different number of disks, and each set is analogous to a gear in an automobile.

The second difference mentioned in paper between FREP technique and the PARAID is the condition leading to shifting the gear, they claimed that PARAID relies on the disk utilization, again this is not totally correct, PARAID relies on disk utilization while taking into account the performance and response time.

Third point which makes PARAID better than FREP which is not mentioned in the paper is that PARAID has a middle range disks that are powered-cycled more frequently, PARAID limits the power cycling of disks by inducing a bimodal distribution of busy and idle disks , busier disks stay powered on, more idle ones stay off, leaving a set of middle range disks, but FREP has to maintain disk states, active or idle/standby.

2. FREP vs. Rabbit

Rabbit provides a wide range of power performance settings, from a low minimum power to a high maximum performance, and provides ideal power proportionality for each setting, which means that the performance to power ratio at all performance levels is equivalent to that at maximum performance level when having all nodes powered up.

There are some similarities between FREP and Rabbit in terms of skewed replication and energy proportionality, I see that these similarities are the main idea of this paper, and moreover the Rabbit presented some experiments on the energy proportionality and FREP didn't. Rabbit introduced power proportionality of equal-work data-layout policy, and an improvements by introducing the idea of gearing, they have two groups, primary and secondary, secondary are grouped into gear and recovery. In this paper they claimed that one of the FREP's main concerns is to save energy while satisfying the SLA requirement, and this is done by observing the workloads and SLA violations and adjust the number of active nodes, but Rabbit doesn't have this mechanism, but we can say that FREP didn't provide experiments about this, and at the same time Rabbit took into account the performance level while reducing energy, and they presented experiments to show the relation between the performance and active nodes, the second difference that has been mentioned, is that Rabbit didn't show the storage requirement, but actually it is not necessary, since it is clear. The third difference is that additional replicas grows exponentially and depends on the number of primary nodes, and these are constraints, actually Rabbit has such constraint, but this because they wanted to satisfy the ideal power proportionality which has been violated by their naïve policy , and also they added a lower and upper bound on the spread of replicas. The Rabbit met the goals of low-minimum power, high-maximum performance, fast, fine-grained scaling and ideal power proportional.

By approving the similarities between Rabbit and FREP which are the main ideas in FREP's paper, and refuting the differences, and presenting the goals met by Rabbit, it is clear that Rabbit is better that FREP.

Finally FREP paper ideas have been inspired from other papers, and it didn't come with experiments to support claims in some cases, like power proportionality and the adaptation of the number of active nodes when workloads violate the requirement of SLA.