Final Mini Project Demonstration UE18CS257F

Project Title : Energy efficiency for HDFS

Project ID : MPW20HLP03

Project Guide : Prof. H L Phalachandra

Project Team :

- Abhishek Das (PES1201800177)
- N Sanketh Reddy (PES1201800389)
- Bhargav SNV (PES1201800308)

Project Abstract and Scope

- In our project, we propose a more dynamic approach to Hadoop Distributed File Systems (HDFS) that provides an energy efficient paradigm for clusters using HDFS.
- The project does not involve working on the actual Hadoop codebase. Instead, we work with a simulator which handles the default policies of HDFS and accurately simulates its functioning.

Design Approach

Design approach followed

- We implemented a custom zone layout over Hadoop clusters. The cluster was split into two zones, the Hot zone and the Cold Zone.

- To set up our working environment, we set up a simulator to handle the functioning of HDFS.

Benefits and drawbacks

- The use of a simulator helped us avoid physical hardware setup. This type of real world setup would be infeasible.

- However, the simulator does not take into account real world issues.

PES UNIVERSITY

Design Constraints, Assumptions & Dependencies

Constraints

- The original simulator could not measure energy consumption. Hence, additional modules to measure power were setup.

Assumptions

- The values assigned to power consuming events in the simulator are assumed to be close to real world values.

Design Description

New/Modified Policies

A new block placement policy and heartbeat mechanism have been devised.

- Blocks are placed strategically in an energy efficient manner. Nodes that are not active (Cold nodes) are put to sleep and hence do not have a heartbeat like the other active nodes.

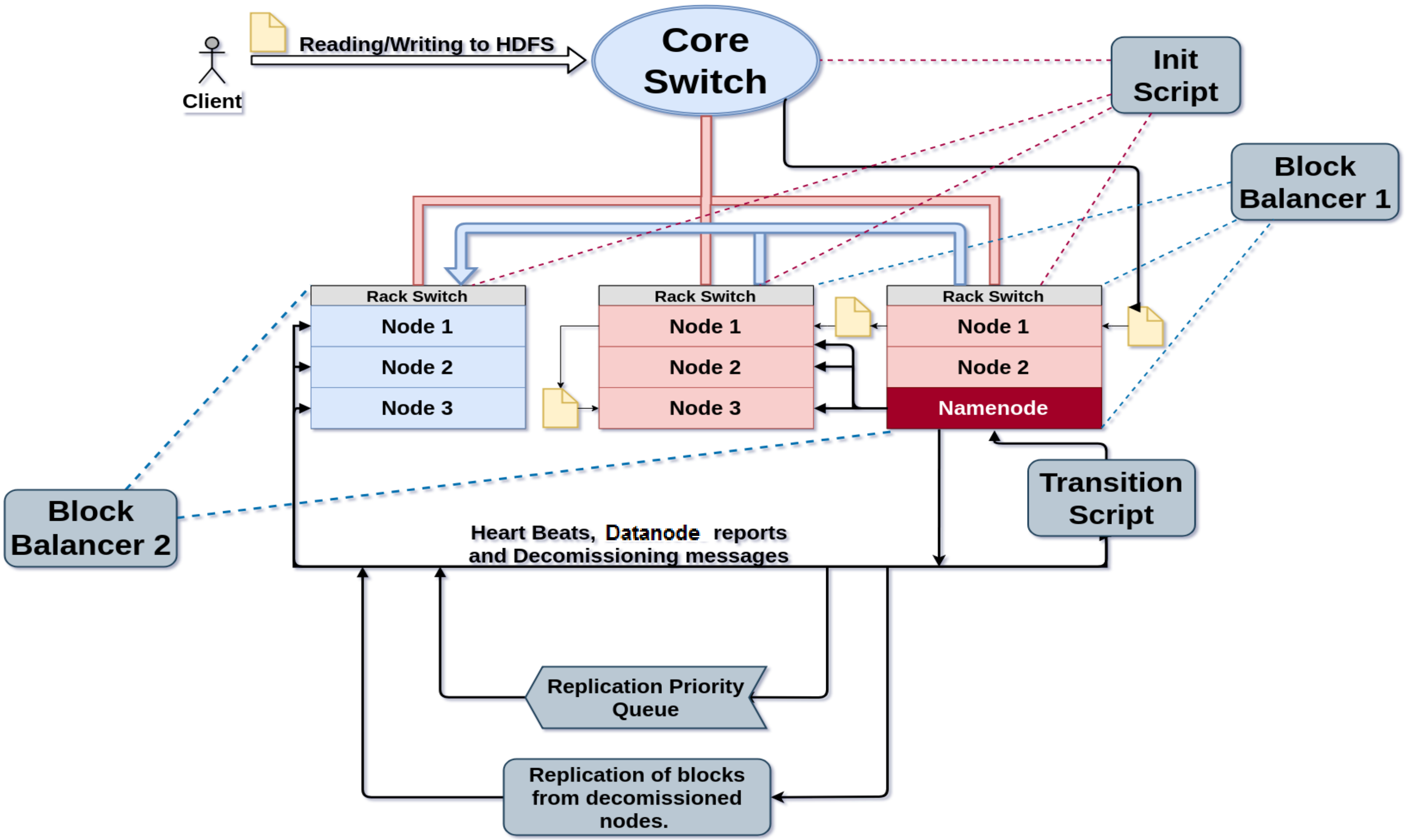
Dynamic data storage/transfer mechanism between hot and cold zones using a transition script.

- As and when data turns cold/hot, the blocks are transferred across zones.

Customized block balancers for each zone.

- Custom block balancers work on each zone to keep data fault tolerant and strategically placed

Custom Zone Layout



Cluster and Zone creation

- The cluster is brought into an active state and Hot, Cold Zones are created.
- The percentage of Hot Zone is pre determined based off the number of nodes with SSDs as a storage device.

Initial Block Distribution

- New data on entering the cluster is distributed among the hot nodes.
- These blocks are placed such that minimum number of nodes are used to store maximum number of blocks along with replicas.

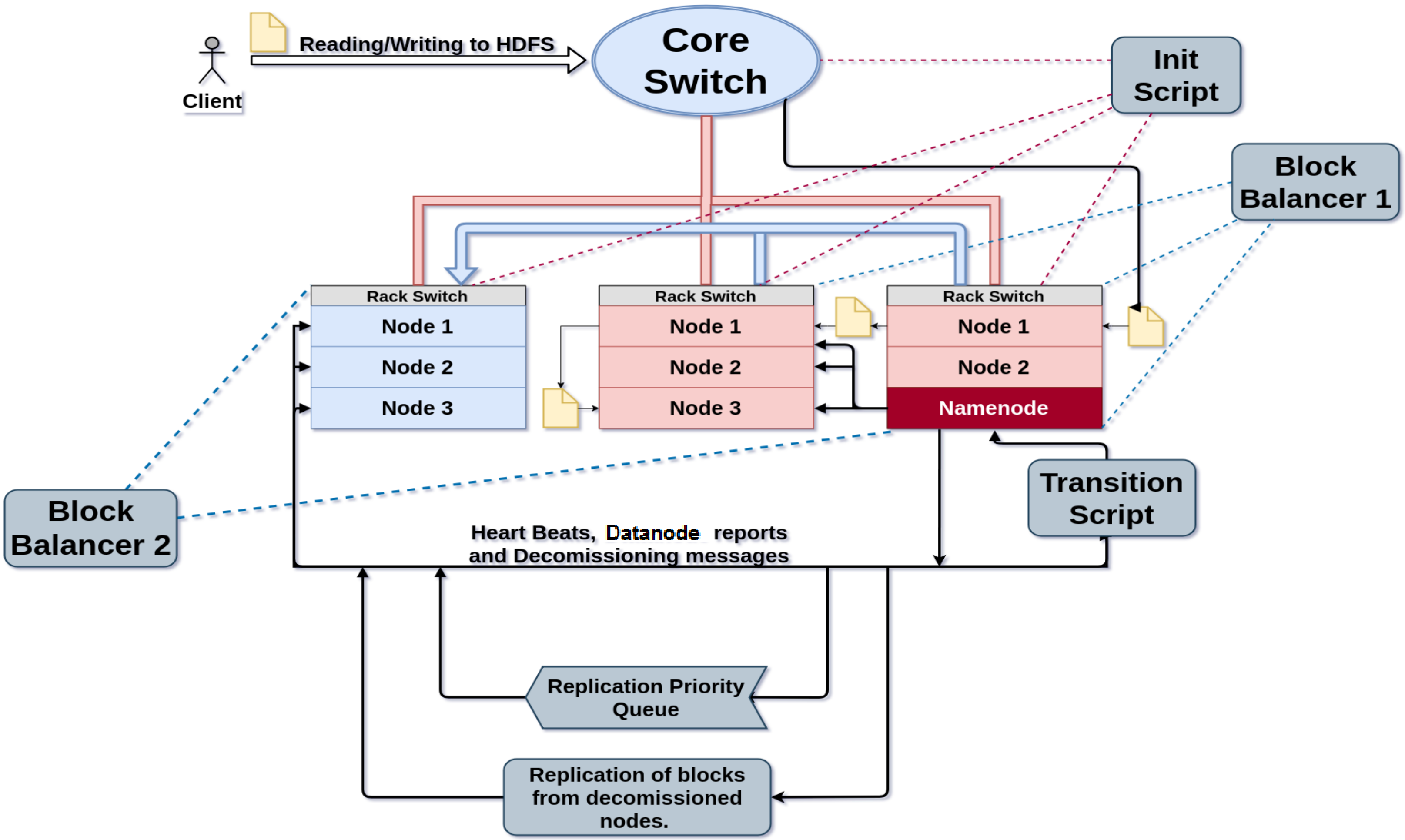
Block Transfer

- As blocks in the hot zone are unused over time, they turn cold.
- The transition script recognises cold blocks and moves them to the cold zone. Likewise, if a cold block is accessed frequently, it is moved to the hot zone.

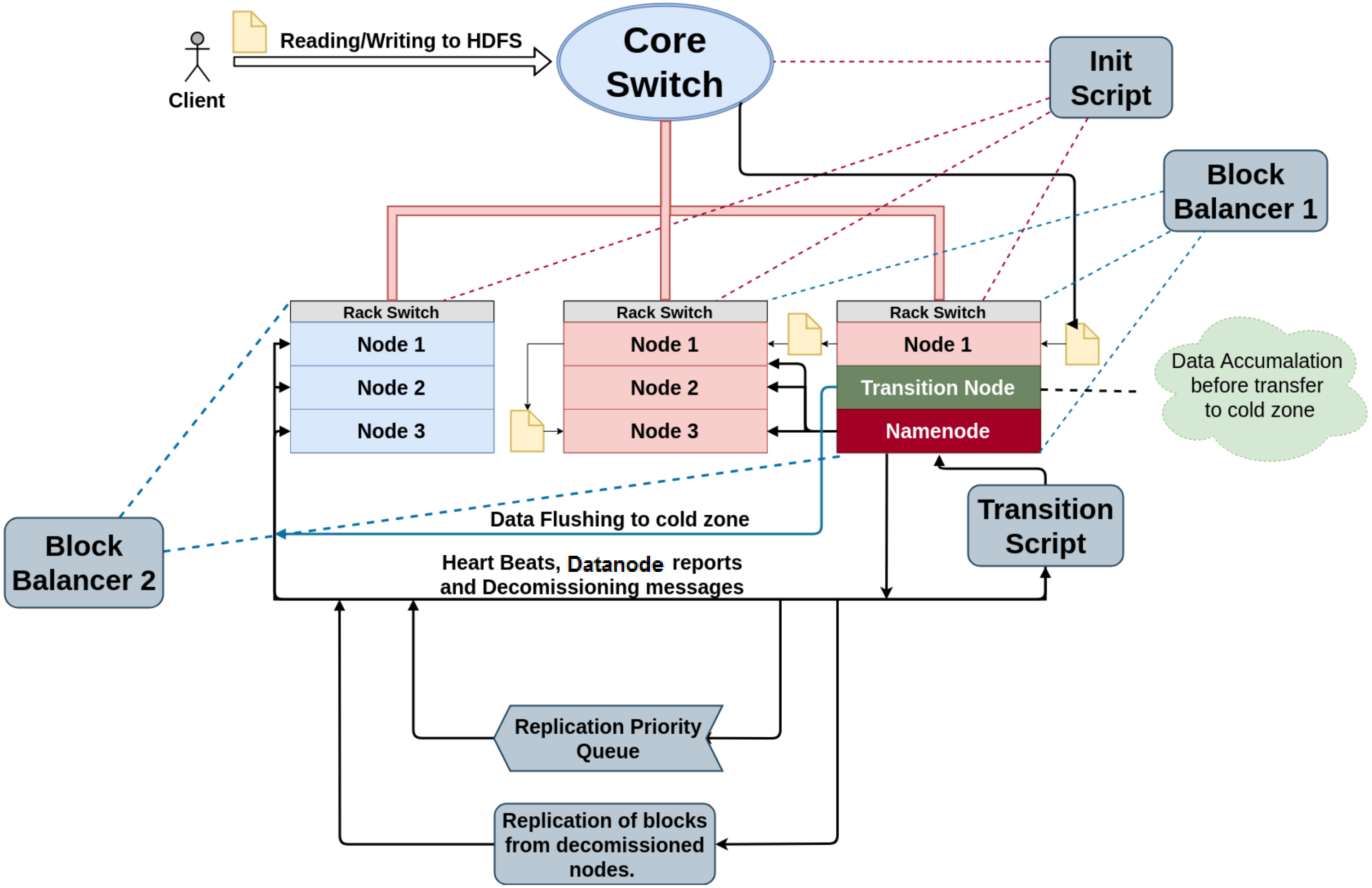
Block Balancing

- As blocks are transferred between zones, they need to be replicated. The hot and cold zone have different replication factors. Usually with the cold zone having 2/3rd the value of hot zone.
- To accommodate different replication factors in zones, different block balancers are employed.

Custom Zone Layout



Alternate Algorithm



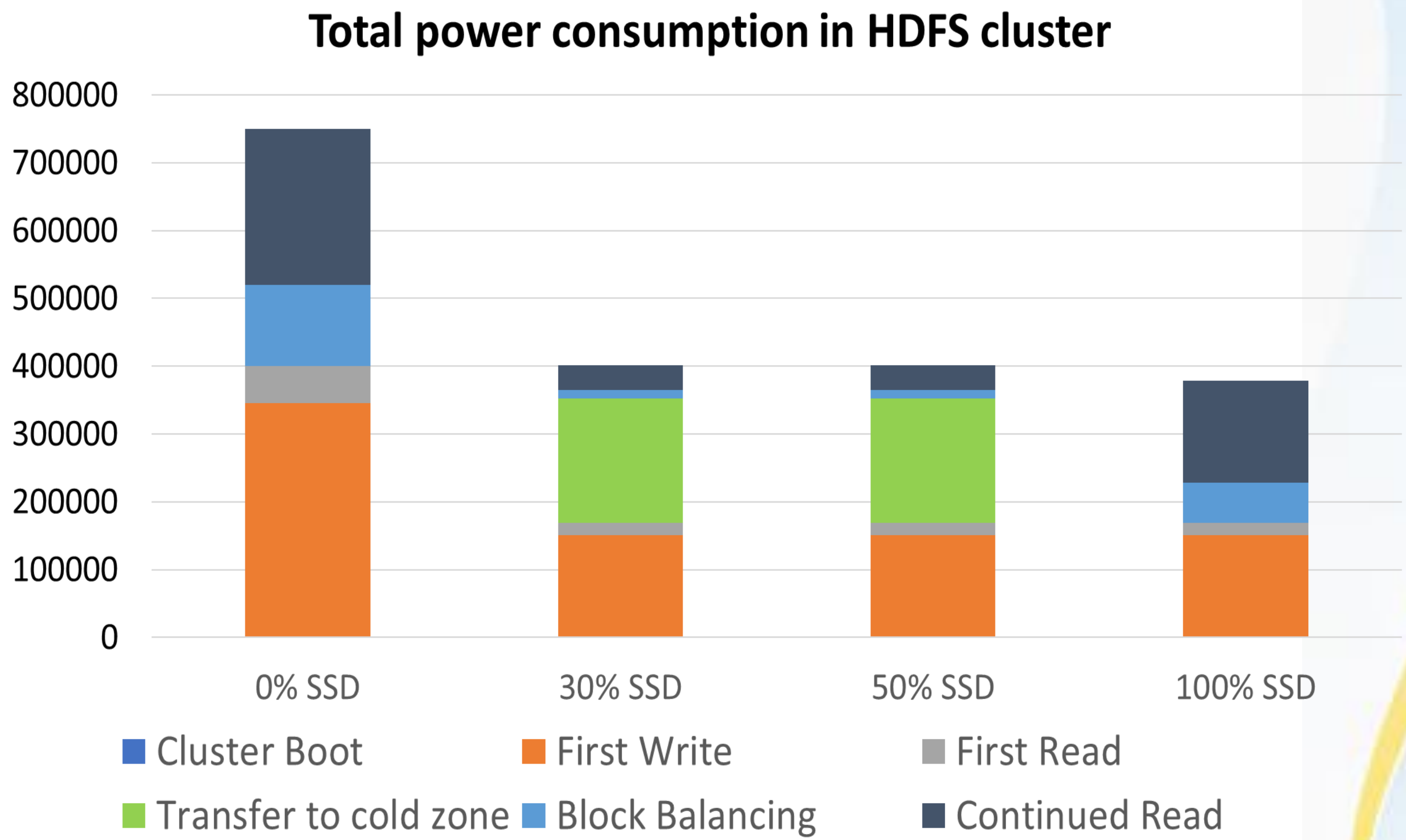
Technologies Used

- HDFS replication simulator
- IntelliJ IDEA, Java
- Apache Hadoop
- Vagrant, Virtual Box
- Draw.io

Project Demo

Evaluation Results

Energy consumed in a single life cycle

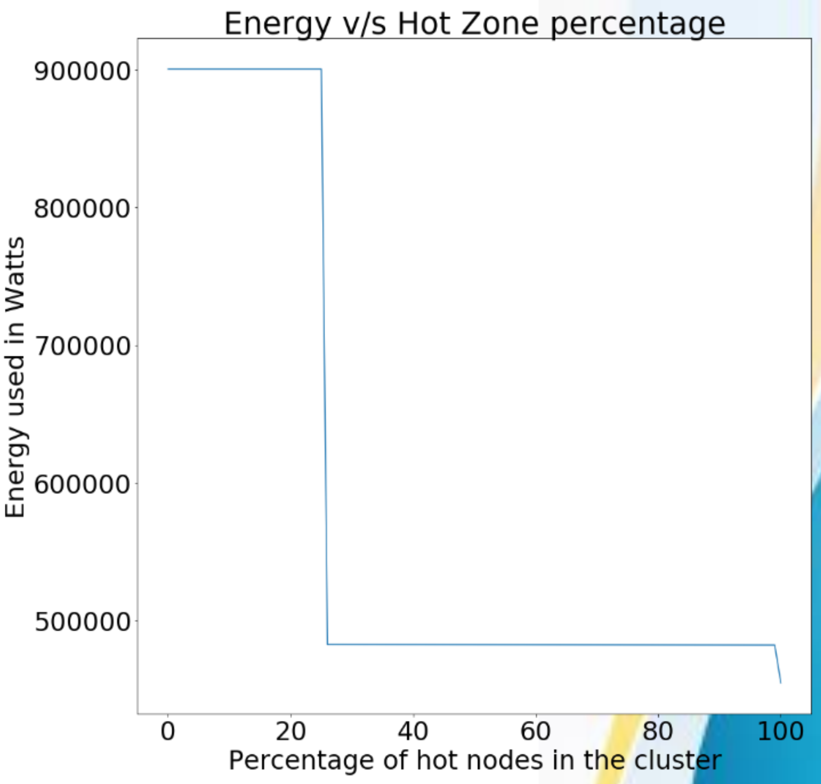


Energy v/s Hot Zone Percentage

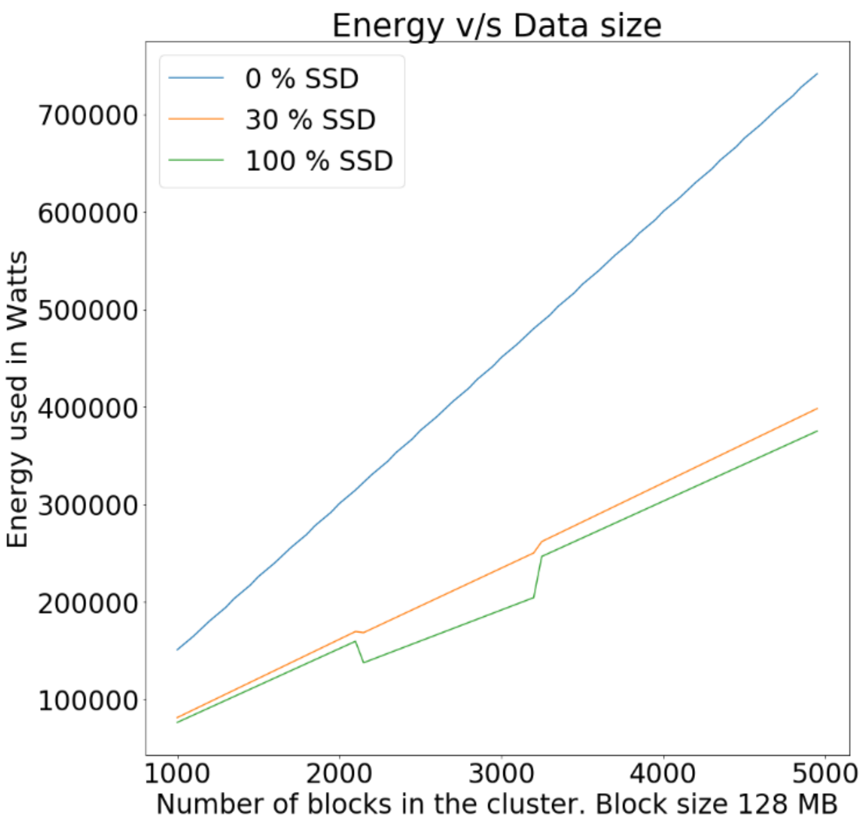
The graph shows how the total energy consumed varies when we vary the Hot-Zone percentage in the cluster.

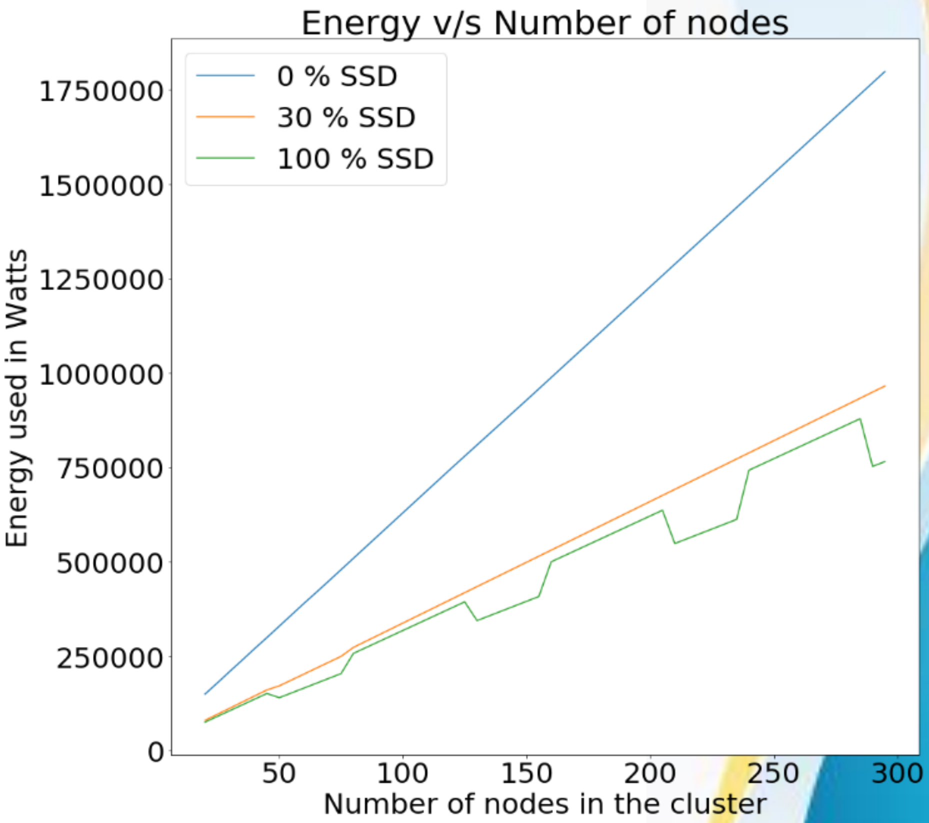
Here there is a sudden drop in the energy consumption at around, Hot Zone % = 30 because, a minimum of about 30% of the cluster must belong to the hot zone to handle peak workloads, else it defaults to the default Hadoop configuration (no hot/cold zone).

The second dip at HZ% 100 is because there are no HDDs in the cluster, this indicates that all the nodes in the cluster make use of SSDs, thus saving a significant amount of energy



Scalability of Algorithm





Thank You

Any Questions?