EAFR: An Energy-Efficient Adaptive File Replication System In Data-Intensive Clusters

Yuhua Lin and Haiying Shen
Dept. of Electrical and Computer Engineering
Clemson University, SC, USA

Outline

- Introduction
- System Design
  - Motivation
  - Design of EAFR
- Performance Evaluation
- Conclusions

Introduction

- File storage systems are important components for data-intensive clusters., e.g., HDFS, Oracle's Lustre, PVFS.

Introduction

Uniform replication policy:

- Create a fixed number of replicas for each file
- Store the replicas in randomly selected servers across different racks

Advantages:

- Avoid the hazard of single point of failure
- Read files from nearby servers
- Achieve good load balance

Introduction

Uniform replication policy:

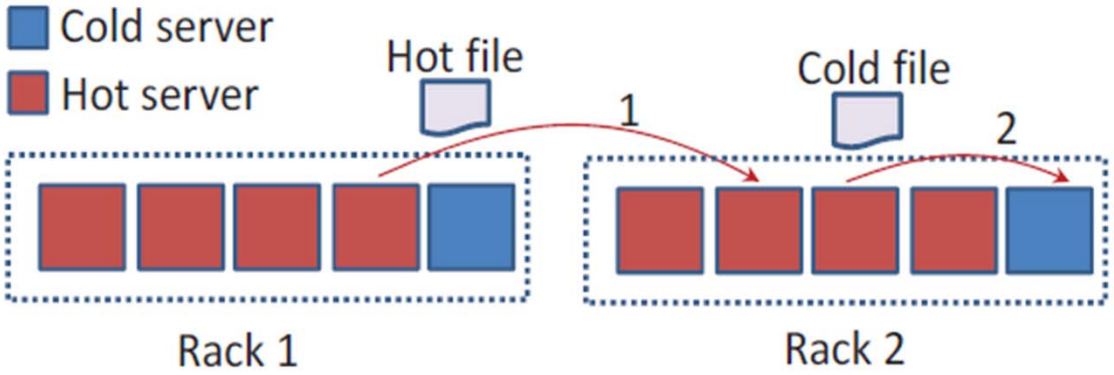
- Create a fixed number of replicas for each file
- Store the replicas in randomly selected servers across different racks

Drawbacks: neglects the file and server heterogeneity

- Cold files and hot files have equal number of replicas
- Not energy-efficient
- Random selection of replica destinations neglects server heterogeneity

Introduction

Energy-Efficient Adaptive File Replication System (EAFR)



- Adapts to file popularities
- Classifies servers into hot servers and cold servers with different energy consumption
- Selects a server with the highest capacity as replica destination

Outline

- Introduction
- System Design
  - Motivation
  - Design of EAFR
- Performance Evaluation
- Conclusions

Motivation: Server Heterogeneity

CPU Utilization	0%	20%	40%	60%	80%	100%
Power (in Watts)	93.7	101	110	121	129	135
Server status	cold	hot	hot	hot	hot	hot

Energy consumption for different CPU utilizations [1]

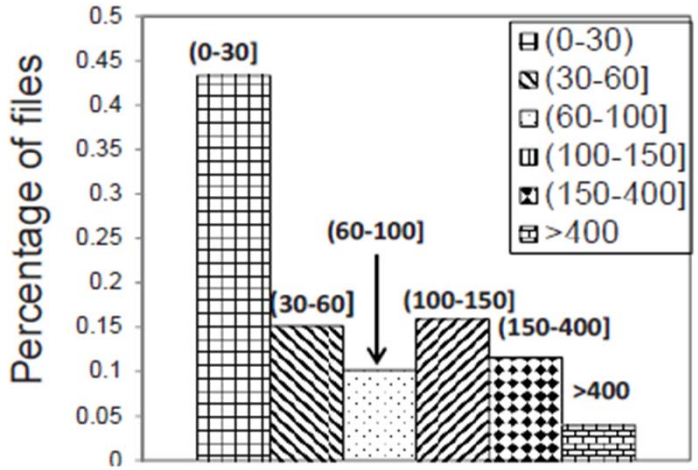
- Hot servers: run at the active state, i.e., with CPU utilization greater than 0
- Cold servers: sleeping state with 0 CPU utilization and do not serve file requests
- Standby servers: temporary hot servers, collect all cold files and turn into cold servers when storages are full

[1] A. Beloglazov and R. Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. CCPE, 24(13):1397–1420, 2012.

Motivation: Files Heterogeneity

Trace data:

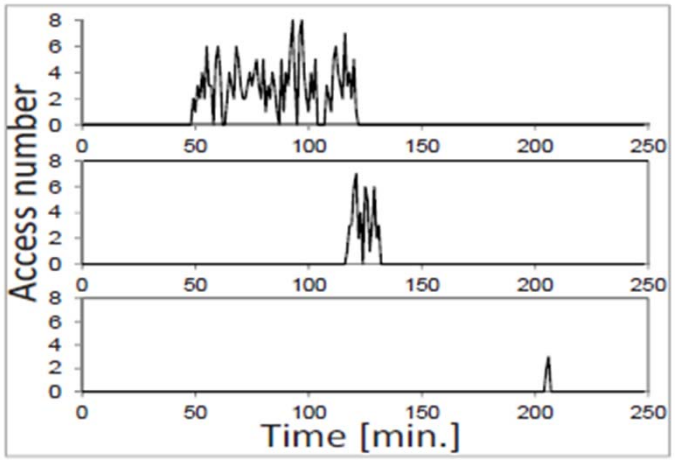
- File storage system trace from Sandia National Laboratories
- Number of file reads for 16,566 files during 4 hour run



- Observation 1: 43% files receive less than 30 reads, 4% files receive a large number of reads (i.e., >400)

Motivation: Files Heterogeneity

- Sort the files by the number of reads, identify the 99th, 50th, and 25th percentiles



- Observation 2: files tend to attract a stable number of reads within a short period of time
- Hint: group files into different categories based on popularity, perform different operations according to their popularities

Adaptive File Replication: Hot Files

A hot file:

1. Average read rate per replica exceeds a pre-defined threshold

$$v_i = \sum_{j=1}^{r_i} v_{ij}$$

$$v_i / r_i > \tau_u$$

2. More than a certain fraction (denoted by γ_v) of a file's replicas attract an excessive number of reads

$$\sum_{j=1}^{r_i} I(v_{ij} > \sigma_u) > r_i \gamma_v \ (0 < \gamma_v < 1)$$

Adaptive File Replication: Hot Files

When to increase the # of replicas for a hot file?

Sever capacity (c_{c_j}): max # of concurrent file requests a server can handle

h_j : # of concurrent reads a server receives

A server is overloaded if: $h_j/c_{c_j} > \tau_c$

An extra replica is needed when a large fraction of servers storing a hot file are overloaded.

$$\sum_{s_j \in S_i} I(h_j/c_{c_j} > \tau_c) > r_i \gamma_s \ (0 < \gamma_s < 1)$$

S_i=(s_1, s_2,..., s_{r_i}) : a set of servers storing a hot file

Where to place the new replica?

Select a server with the highest remaining capacity

Adaptive File Replication: Cold Files

A cold file:

1. Average read rate per replica bellows a pre-defined threshold

$$v_i = \sum_{j=1}^{r_i} v_{ij}$$
$$v_i / r_i < \tau_l$$

2. More than a certain fraction (denoted by γ_v) of a file's replicas attract a small amount of reads

$$\sum_{j=1}^{r_i} I(v_{ij} < \sigma_l) > r_i \gamma_v \ (0 < \gamma_v < 1)$$

Adaptive File Replication: Cold Files

When a file gets cold:

1. Maintaining at least ϵ replicas in hot servers to guarantee file availability
2. Move a replica from a hot server to a standby server
3. When a standby server's storage capacity is used up, turn the standby server to a cold server

Outline

- Introduction
- System Design
  - Motivation
  - Design of EAFR
- Performance Evaluation
- Conclusions

Performance Evaluation: Settings

Trace-driven simulation platform: Clemson University's Palmetto Cluster

- 300 distributed servers
- Storage capacities: randomly chosen from (250GB, 500GB, 750GB)
- 50,000 files, randomly placed on the servers
- Distributions of file reads and writes: follow CTH trace data [2]

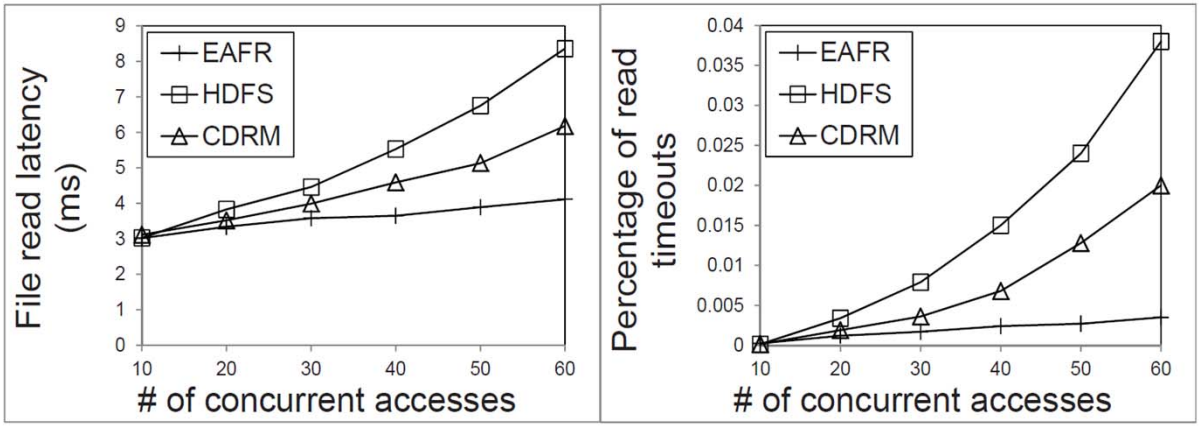
Comparison methods

- HDFS: 3 replicas placed in random servers
- CDRM: 2 replicas initially, increases replicas to maintain the required file availability 0.98 for server failure probability 0.1

[2] Sandia CTH trace data. http://www.cs.sandia.gov/Scalable IO/SNL Trace Data/

Performance Evaluation: Results

- File Read Response Latency

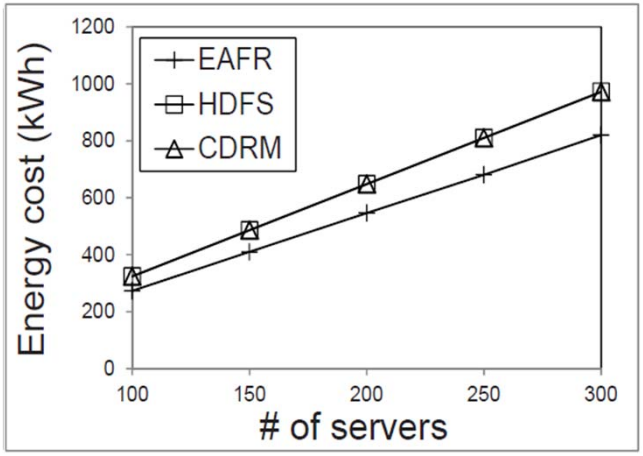


(a) File read response time. (b) Percentage of file read timeouts.

- Observation: HDFS>CDRM>EAFR
- Reason: EAFR adaptively increases the number of replicas for hot files, and the new replicas share the read workload of hot files.

Performance Evaluation: Results

- Energy Efficiency

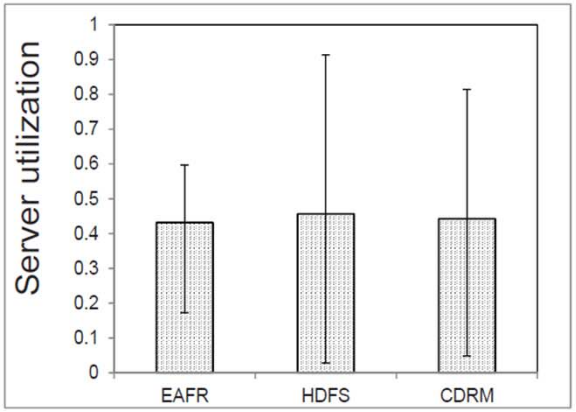


(b) Energy consumption per day.

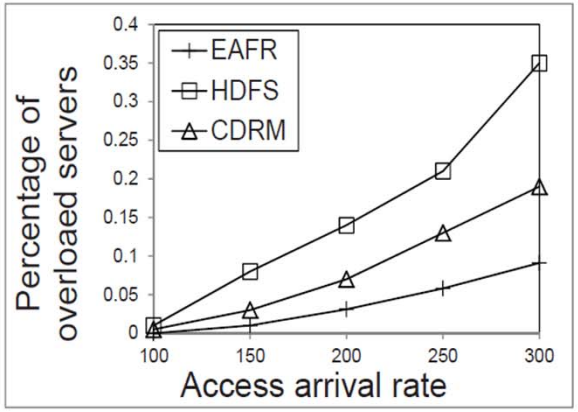
- Observation: EAFR manages to reduce the power consumption by more than 150kWh per day
- Reason: EAFR stores some replicas of cold files in cold servers (in sleeping mode), which results in substantial power saving

Performance Evaluation: Results

- Load Balance Status



(a) Server utilization.



(b) Percentage of overloaded servers.

- Observation: EAFR achieves better load balance than CDRM and HDFS
- Reason: EAFR places new replicas in servers with the highest remaining capacity

Outline

- Introduction
- System Design
  - Motivation
  - Design of EAFR
- Performance Evaluation
- Conclusions

Conclusion

- EAFR: energy-efficient adaptive file replication system
- Trace-driven experiments from a real-world large-scale cluster show the effectiveness of EAFR:
  - Reduce file read latency
  - Save power consumption
  - Achieve better load balance
- Future work: increasing data locality in replica placement

Thank you!

Questions & Comments?

Yuhua Lin

yuhual@clemson.edu

Electrical and Computer Engineering

Clemson University