Allocation Optimization for the Rebalancing Data Service

Ralf Vamosi [1], Mario Lassnig [1], Erich Schikuta [2]
on behalf of the ATLAS Collaboration

[1]CERN, Geneva, Switzerland

[2]University of Vienna, Austria

2018-07-09

Overview

Objective

Objective

Allocate files according to available storage and minimize waiting time for input data.

$\longrightarrow$ Put interesting files as close as possible to job executing node.

C3PO Data Popularity

A dynamic data placement agent replicating datasets

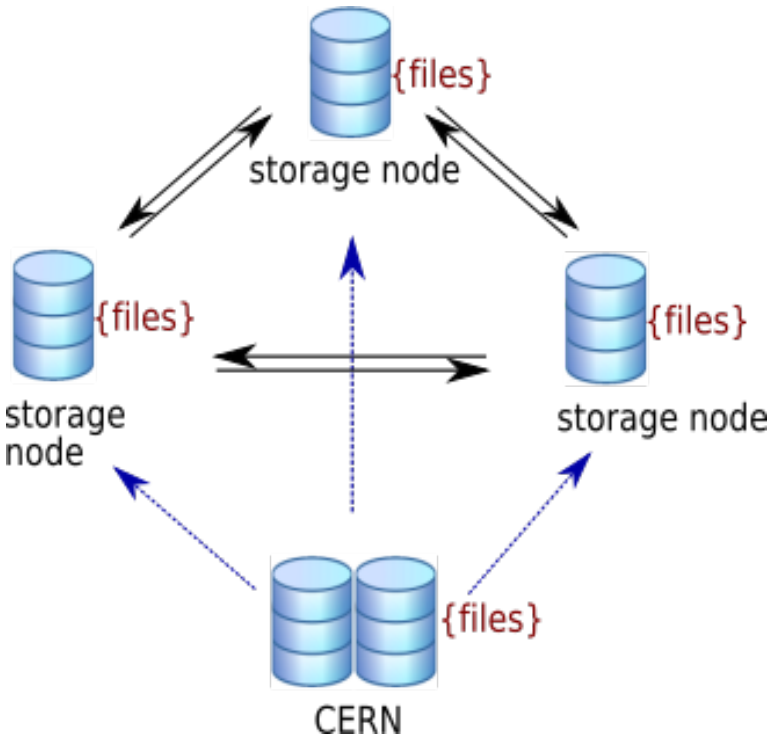
BB8 Rebalancer

An automatic agent rebalancing data volumes between data centers to avoid overloads

Approach

Proposal of an evolutionary algorithm for a continuous update of file allocation

- Stream of new files: Allocation
- Continuous update of files in file system: Reallocation

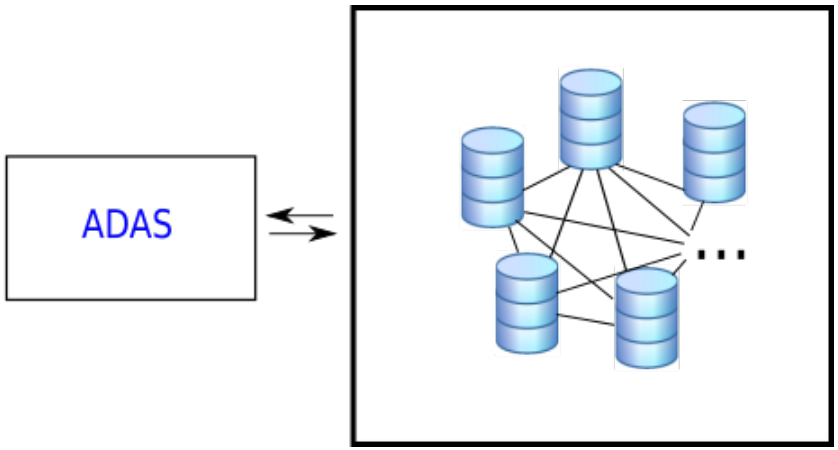


Properties and Constraints

- Cluster data on dataset level (DS) and map to storage nodes $DS : (\{files\}, Pop, DSD_{vect})$
  1. Operating on subsets only and a relaxed capacity constraint
  2. Select some of these important datasets
  3. Update locations according to cost metric
- *Heuristic* dataset popularity Pop_i - How likely the DS_i will be accessed by a job
- *Heuristic* dataset dependency $DSD_{i,j}$ - How likely files of two DSs (i,j) get into a new DS used by a job
- Most important data centers are put into a feasible subset
  - 10 of 130 data centers hold > 50 % data

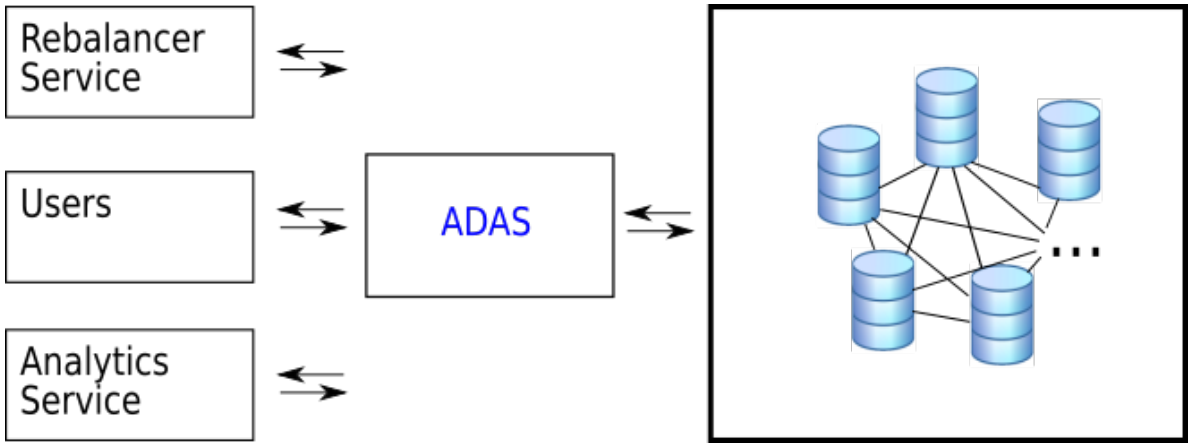
ATLAS Data Allocation Optimization Service (ADAS)

- Runs in the background and is independent of the existing rebalancing mechanism
- Collects data from other services
- Takes periodically subsets and flux of files



Communication of ADAS

- Communicates decisions and takes feedback
  - Rebalancer service which automatically and dynamically rebalances data between data centers
  - Users: accept, decline, or override the new dataset placement
  - Analytics service: Dataset Popularity, Dataset Dependency etc. are updated



First version

Introduced in
Ralf Vamosi, Mario Lassnig, Erich Schikuta - Data Allocation Based on Evolutionary Data Popularity Clustering - Computational Science ICCS 2018: 18th International Conference ... - Springer

- Sub-optimization: 20 data centers
- Metric = # WAN transfers/job decreased by over 10 %

What's next?

Code improvements:

- Runtime improvements for bigger file sets and more network nodes
- Optimizer as a service

Improvements in the optimization model:

- Network characteristics, especially I/O bandwidth between data centers are usually different
- Differences in data centers, in terms of computing performance
- Various job types which use different classes of data

Thank you!

BACKUP

Target metric: WAN transfers per job

Optimization target: expected number of WAN transfers, $\#tx_{WAN}$
$\rightarrow$ less network load $\rightarrow$ less waiting time
Parameter **S**: Storage matrix specifying file arrangement

$$\underset{\mathbf{S}}{argmin} \sum_{j \in \{jobs\}} \#tx_{WAN,j}$$

such that

$$\mathbf{S}^T \times \mathbf{w}_{file} \leq \mathbf{w}_{storage} \quad \textit{(storage constraint)}$$

Re-allocation occupies network: *Move as few files as possible*

1st Heuristic: Data Popularity

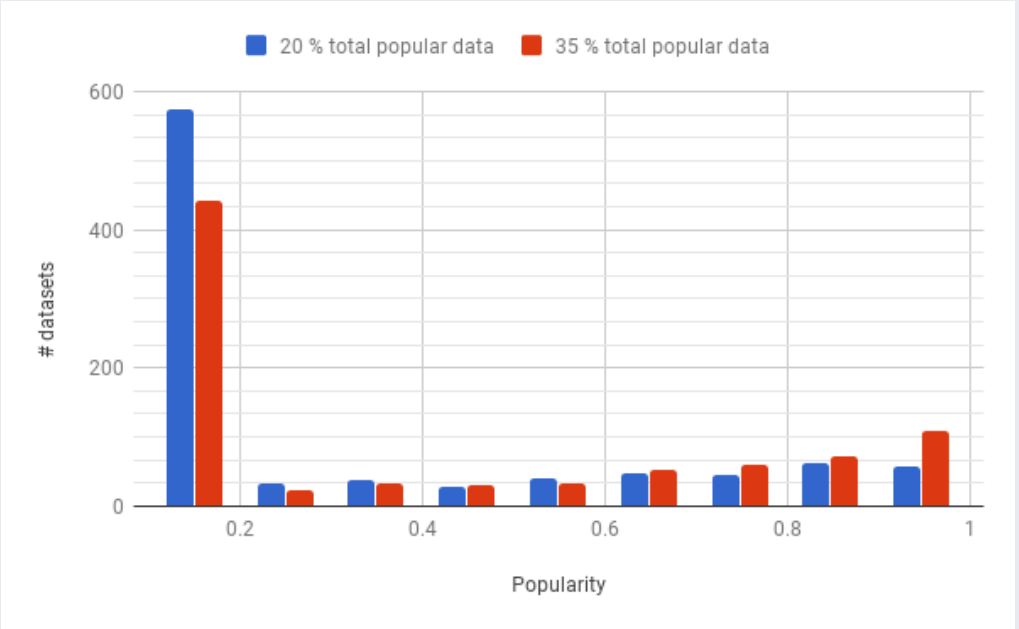
Data popularity

Dataset
$DS = (\{files\}, Pop, DSD_{vect})$

$$Pop: \quad \{DS\} \rightarrow [0,1]$$
$$DS_n \mapsto Pop_n$$

Data popularity =
How likely will the DS get accessed

20 % / 35 % total popular data



2nd Heuristic: Dataset Dependency

DSD

Dataset
$DS = (\{files\}, Pop, DSD_{vect})$

$$DSD: \quad \{DS\} \times \{DS\} \rightarrow [0,1]$$
$$(DS_n, DS_k) \mapsto DSD_{n,k}$$

DSD = How likely files of two DSs get into a new DS used by a job

Dependencies between random datasets

$$\mathbf{DSD} = \begin{bmatrix} 1 & 0.2 & \dots & 0 \\ 0.2 & 1 & \dots & 0.12 \\ \dots & & & \\ 0 & 0.12 & \dots & 1 \end{bmatrix}$$