Classifying Temporal Characteristics of Job I/O Patterns Using Machine Learning Techniques

IODC Workshop
June 25, 2019

Eugen Betke
Research Group
German Climate Computing center

Julian Kunkel
Compute Science
University of Reading

Table Of Contents

Mistral, the HPC system for Earth system research (HLRE-3)

Peak performance	*3.14 PetaFLOPS*
Compute nodes	*3,300*
Compute cores	*100,000*
Memory	*266 Terabytes*
Storage (two file systems)	*54 Petabytes*

Motivation

- Goals: Finding jobs with
  - high I/O load, but inefficient data access
    - e.g., for application optimization
  - critical I/O load, that can degrade file system performance
    - e.g., for better job scheduling
- Approach
  - Identify an individual job of any kind (e.g., I/O intensive job)
  - Cluster similar jobs
  - Look-up for the cluster, that contains the job

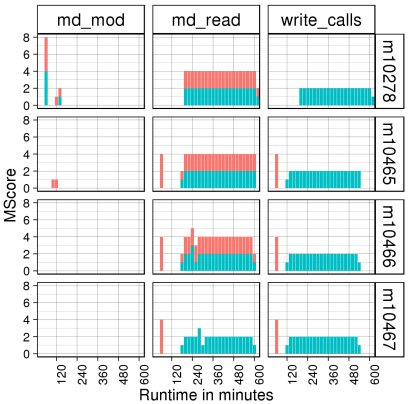
Table Of Contents

Monitoring data

- Raw data (captured by monitoring system)
  - 4-dimensional
    - Client nodes, file systems, I/O metrics, job runtime
  - Capture interval: 5 seconds
  - Domain
    - Different for all metrics
    - Different units (e.g., Ops/s, MiB/s)
- Segmented data (derived from raw data)
  - 4-dimensional
    - Client nodes, file systems, I/O metrics, job runtime
  - Segment interval: 10 minutes
  - Domain
    - The same for all metrics (Mscore)

$$\text{Mscore} = \begin{cases} 0 \text{ if low I/O performance} \\ 1 \text{ if high I/O performance} \\ 4 \text{ if critical I/O performance} \end{cases}$$

    - Unitless



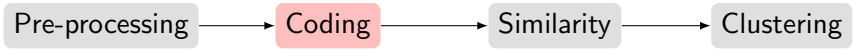
Example for segmented data. Colors represent access to different file systems.

Table Of Contents

Clustering process overview

- Data pre-processing
  - Reduce dimensionality
- Coding
  - Select appropriate time series representation
- Similarity
  - Define how to compare the jobs
- Clustering
  - Create clusters of similar jobs

- Creation of job profiles
  - Reduction of all dimensions to a fixed set of values
- Dimensionality reduction
  - Available dimensions: node, file system, and metric dimension
  - Reduction functions: sum() or mean()



- Binarization
  - Maps segments to 9-bit numbers
  - $v_i = \begin{cases} \text{false if segment}_i = 0, \\ \text{true otherwise.} \end{cases}$
    with bit position $i \in$ [enumerated metrics]
- Zeros aggregation
  - Aggregates continuous occurrences of zero segments to one zero segment.

Bit position i	Metric	Binary	Decimal
0	md_file_create	000000001	1
1	md_file_delete	000000010	2
2	md_mod	000000100	4
3	md_other	000001000	8
4	md_read	000010000	16
5	read_bytes	000100000	32
6	read_calls	001000000	64
7	write_bytes	010000000	128
8	write_calls	100000000	256

Example

Binarization

```
'coding': [1:5:0:0:0:0:0:0:96:96:96:96:96:96:96]
'length': 15
```

Binarization + Zeros aggregation

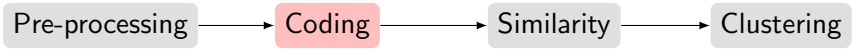
```
'coding': [1:5:0:96:96:96:96:96:96]
'length': 15
```

Decoding examples

$1 =$ md_file_create

$5 = 4 + 1 =$ md_file_create + md_mod

$96 = 64 + 32 =$ read_calls + read_bytes



- Hex-Quantization
  - Quantizes mean segment performance to 16 levels
- Phase extraction
  - Splits time series at zero segments in sub time series (I/O phases)
  - Removes zero segments.
  - Preserves order of I/O phases.

Example

Hex-Quantization

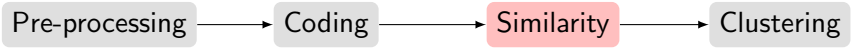
```
md_file_create [0:0:2:2:2:9:3:0:9:1:1:1:0]
md_file_delete [0:0:0:0:0:0:0:0:0:0:0:0:0]
md_mod         [0:0:0:0:0:0:0:0:0:0:0:0:0]
md_other       [0:0:0:0:0:0:0:0:0:0:0:0:0]
md_read        [0:0:0:9:3:0:0:0:0:1:0:0]
read_bytes     [0:0:0:0:0:0:0:0:0:0:0:0:0]
read_calls     [0:0:0:0:0:0:0:0:0:0:0:0:0]
write_bytes    [0:0:0:0:0:0:0:0:0:0:0:0:0]
write_calls    [0:0:0:0:0:0:0:0:0:0:0:0:0]

length: 13
```

Hex-Quantization + Phase extraction

```
md_file_create : [[2,2,2,9,3], [9,1,1,1]]
md_file_delete : []
md_mod         : []
md_other       : []
md_read        : [[9,3], [1]]
read_bytes     : []
read_calls     : []
write_bytes    : []
write_calls    : []

length: 13
```



- Levenshtein
  - Number of operation (inserts, deletes, and changes) requires to convert one coding in another

Example

The distance between SATURDAY and SUNDAY is 3. To convert the words, we can (1) delete characters A, (2) delete T and (3) substitute R by N.

- Sliding window
  - Find maximum similarity between two codings
  - Slide shorter coding over a longer one an compute similarity of the corresponding parts

Example

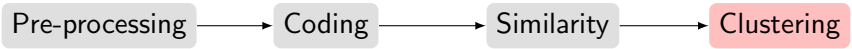
```
sim([2:2:2:2:8:2:2],[9:3:-:-:-:-:-]) = (2/9+2/3)/7 = 13%
sim([2:2:2:2:8:2:2],[-:9:3:-:-:-:-]) = (2/9+2/3)/7 = 13%
sim([2:2:2:2:8:2:2],[-:-:9:3:-:-:-]) = (2/9+2/3)/7 = 13%
sim([2:2:2:2:8:2:2],[-:-:-:9:3:-:-]) = (2/9+3/8)/7 = 8%
sim([2:2:2:2:8:2:2],[-:-:-:-:9:3:-]) = (8/9+2/3)/7 = 22%
sim([2:2:2:2:8:2:2],[-:-:-:-:-:9:3]) = (2/9+2/3)/7 = 13%
```

- Phases
  - Find best I/O phases combination of two jobs with a maximum similarity
  - Depends on phase extraction from data pre-processing and sliding window similarity
  - Preserve the order of the phases

Example

Best I/O phases combination between `[[9:3], [1]]` and `[[2:2:2:2:8:2:2], [1], [8:1:1]]` would be:

```
[[-:-:-:-:9:3:-], [1], [-:-:-]]
[[2:2:2:2:8:2:2], [1], [8:1:1]]
```



- ClusteringTree
  - Clustering and labeling 10,000 jobs with agglomerative clustering algorithm
    - Clustering more than 10,000 jobs $\rightarrow$ Scalability problems
    - Library: sklearn 0.22.1 in python 3.8.0
  - Training of a decision tree model with data from the previous step
  - Predict labels of 1,000,000 jobs with the trained decision tree model
- SimplifiedDensity
  - Just one parameter: $SIM \in [0, 1]$
  - Just one condition: Add job to cluster if similarity to cluster centroid is smaller than SIM
  - Centroids
    - becomes jobs that could not be assigned to a cluster
    - form new clusters

Explored clustering stacks

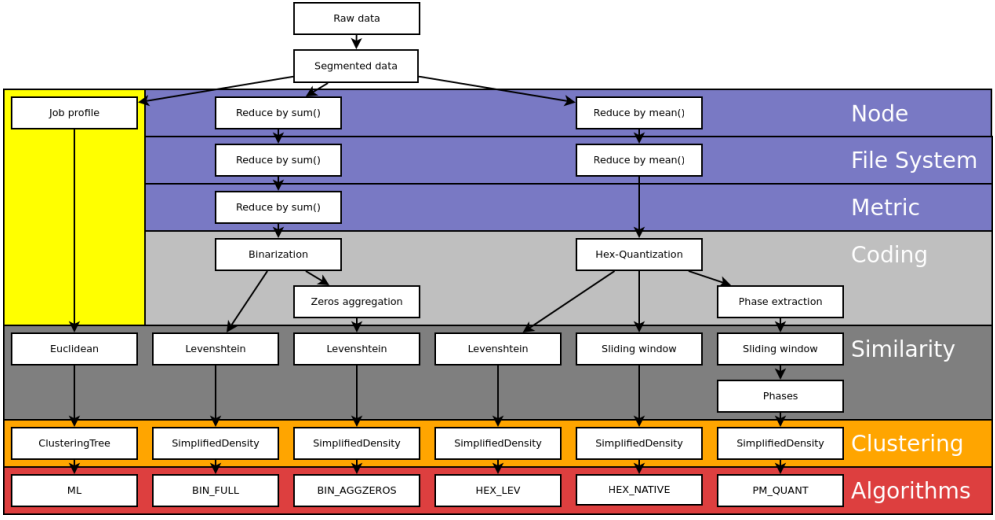


Table Of Contents

ML

Clustering Stack

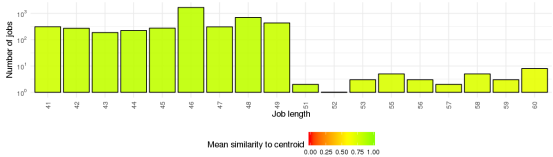
Object	Operation
All dimensions	Job profile
Similarity	Euclidean
Clustering	ClusteringTree

- Mostly polluted clusters, i.e., jobs with different I/O behaviour are in the same cluster
  - Example:[1], [0:0:14:0], [4:0], [0:0:4:0], [272:272:272]
- No good results achieved

BIN_ALL

Clustering Stack

Object	Operation
Node	Reduce by sum()
File System	Reduce by sum()
Metric	Reduce by sum()
Coding	Binarization
Similarity	Levenshtein
Clustering	SimplifedDensity



Clustering with SIM = 0.7 results in 7615 jobs and 3306 job types.

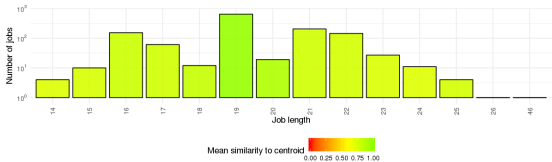
coding	type
0:0:0:0:0:0:294:0:0:0:0:32:0:0:0:0:0:0:0:0:0:0:0:0:32:0:0:0:0:0:0:0:0:0:0:0:32:0:0:0:0:0:0:0:0:0	centroid

coding	count
0:0:0:0:0:0:359:96:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0	95
0:0:0:0:0:0:295:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0	62
0:0:0:0:0:0:0:0:0:0:0:4:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0	47
0:0:0:0:0:0:359:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0	44
0:0:0:0:0:6:6:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0	40

BIN_AGGZEROS

Clustering Stack

Object	Operation
Node	Reduce by sum()
File System	Reduce by sum()
Metric	Reduce by sum()
Coding	Binarization
	Zeros aggregation
Similarity	Levenshtein
Clustering	SimplifedDensity



Clustering with SIM = 0.7 results in 1295 jobs and 336 job types.

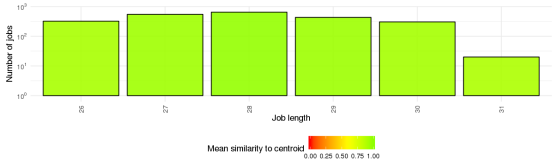
coding	type
272:272:272:272:272:272:278:286:272:272:272:272:272:272:272:272:272:272:272:272:272:272:272	centroid

coding	count
272:272:272:272:272:272:278:286:272:272:272:272:272:272:272:272:272:272:272:272:272:272:272	528
272:272:272:272:272:272:406:286:272:272:272:272:272:272:272:272:272:272	96
272:272:272:272:272:272:279:31:272:272:272:272:272:272:272:272:272:272:272:272:272:272:272:272	53
272:272:272:272:272:272:272:272:272:272:272:272:279:319:272:272:272:272:272:272:272:272:272	52
272:272:272:272:272:272:279:319:272:272:272:272:272:272:272:272:272:272:272:272:272:272:272	50

HEX_LEV

Clustering Stack

Object	Operation
Node	Reduce by mean()
File System	Reduce by mean()
Coding	Hex-Quantization
Similarity	Levenshtein
Clustering	SimplifedDensity



Clustering with SIM = 0.9 results in 5769 jobs and 2473 job types.

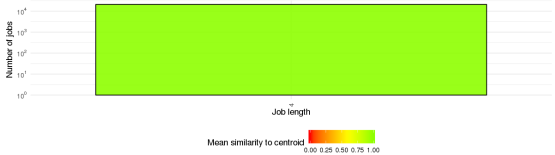
md_file_create	md_file_delete	md_mod	md_other	md_read	read_bytes	read_calls	write_bytes	write_calls	type
0:...:0	0:0:0:1:0:0:0:0:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	centroid

md_file_create	md_file_delete	md_mod	md_other	md_read	read_bytes	read_calls	write_bytes	write_calls	count
0:...:0	0:0:0:0:0:0:1:0:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	606
0:...:0	0:0:0:0:0:0:0:0:1	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	562
0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:0:0:0:0:0:1:0:0	0:...:0	0:...:0	429
0:...:0	0:0:0:1:0:0:0:0:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	185
0:0:0:0:0:0:1:1:1	0:...:0	0:0:0:0:0:0:1:0:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	75

HEX_NATIVE

Clustering Stack

Object	Operation
Node	Reduce by mean()
File System	Reduce by mean()
Coding	Hex-Quantization
Similarity	Sliding window
Clustering	SimplifedDensity



Clustering with SIM = 0.99 results in 20908 jobs and 11997 job types.

md_file_create	md_file_delete	md_mod	md_other	md_read	read_bytes	read_calls	write_bytes	write_calls	type
0:...:0	0.04:0.04:0.07:0	0.04:0.04:0.14:0	0:0:0.25:0	0.04:0:0:0	0:0.04:0.04:0	0:...:0	0:0.04:0.04:0	0.04:0:0:0	centroid

md_file_create	md_file_delete	md_mod	md_other	md_read	read_bytes	read_calls	write_bytes	write_calls	count
0:...:0	0:0:0.04:0	0:0:0.11:0	0:0:0.29:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	1344
0:...:0	0:0:0.33:0	0:0:0.08:0	0:0:0.04:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	458
0:0:0:0.04	0:0:0.04:0	0:0:0.11:0	0:0:0.29:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	384
0:...:0	0:0:0.04:0	0:0:0.07:0	0:0:0.29:0	0:...:0	0:...:0	0:...:0	0:0:0.04:0	0:...:0	356
0:...:0	0:0:0.04:0	0:0:0.11:0	0:0:0.5:0	0:...:0	0:...:0	0:...:0	0:...:0	0:...:0	278

PM_QUANT

Clustering Stack

Object	Operation
Node	Reduce by mean()
File System	Reduce by mean()
Coding	Hex-Quantization
	Phase extraction
Similarity	Sliding window
	Phases
Clustering	SimplifedDensity

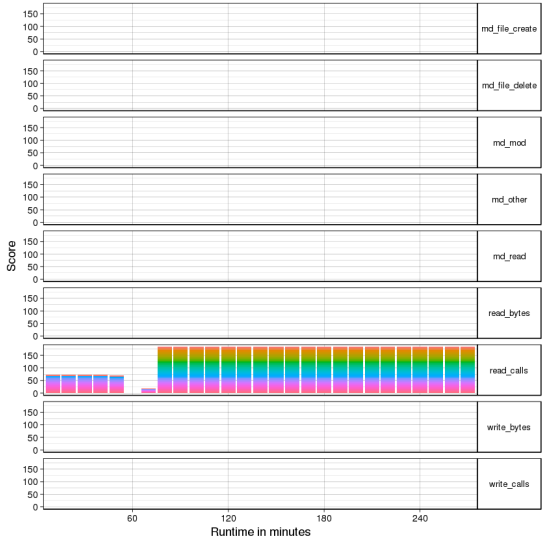
Clustering with SIM = 0.7 results in 5175 jobs and 154 job types

...	md_file_delete	md_mod	...	...	...	...	...	...	type
0: ... :0	4:0:0:0:0:0	4:0:0:0:0:0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	centroid

...	md_file_delete	md_mod	...	...	...	...	...	...	count
0: ... :0	4	4	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	2329
0: ... :0	4:0:0:0:0:0	4:0:0:0:0:0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	1773
0: ... :0	4:0	4:0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	224
0: ... :0	2	4	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	65
0: ... :0	0:4	0:4	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	57

Table Of Contents

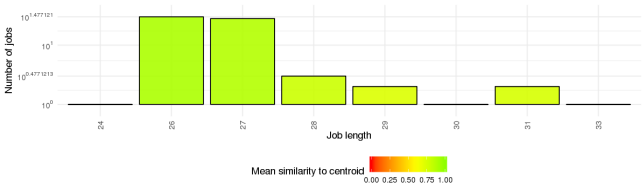
Use case: I/O intensive job



PM_QUANT

Clustering Stack

Object	Operation
Node	Reduce by mean()
File System	Reduce by mean()
Coding	Hex-Quantization
	Phase extraction
Similarity	Sliding window
	Phases
Clustering	SimplifedDensity



Clustering with SIM = 0.7 results in 68 jobs and 59 job types

...	...	...	...	...	...	read_calls	...	...	type
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	3:3:8:8:8:5:6:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	job
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	centroid

...	...	...	...	...	...	read_calls	...	...	count
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	8:8:8:8:8:2:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	4
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	8:8:8:8:8:3:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	3
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	7:7:7:7:7:2:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	2
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	8:8:8:8:8:5:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	2
0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	0: ... :0	8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8:8	0: ... :0	0: ... :0	2

Table Of Contents

Summary

- Applied methods
  - Data pre-processing / dimensionality reduction
  - Coding
  - Similarity
  - Clustering
- ML, BIN_ALL, BIN_AGGZEROS, HEX_LEV, HEX_NATIVE, PM_QUANT
- Lack of tools for clustering quality assessment of time series

Thank you for your attention!