NHR FAU

Friedrich-Alexander-Universität Erlangen-Nürnberg

Specifications and Standards

18.07.2022

Jan Eitzinger, NHR@FAU

What is this all about?

Standards and interfaces to ease interoperability

- Generic data structure descriptions (JSON Schema) – used for
  - Data structures in applications
  - Payload in APIs
  - File format
  - SQL table schema
- Job Archive: Directory layout and file naming specification
- Influx Data Line protocol specification
- API Schemas
  - REST (OpenAPI 3.0)
  - GraphQL

https://github.com/ClusterCockpit/cc-specification

Generic Job Meta data structure

Name	Type	Description
jobId	integer	The unique identifier of a job
user	string	The unique identifier of a user
project	string	The unique identifier of a project
cluster	string	The unique identifier of a cluster
subCluster	string	The unique identifier of a sub cluster
partition	string	The Slurm partition the job was submitted to
arrayJobId	integer	The unique identifier of an array job
numNodes	integer	Number of nodes used
numHwthreads	integer	Number of HWThreads used
numAcc	integer	Number of accelerators used

Generic Job Meta data structure cont.

Name	Type	Description
exclusive	integer	Specifies how nodes are shared. 0 - Shared among multiple jobs of multiple users 1 - Job exclusive 2 - Shared among multiple jobs of same user
monitoringStatus	integer	State of monitoring system during job run 0: Monitoring disabled 1: Monitoring enabled, job not yet archived or ongoing archiving 2: Monitoring enabled und successfully archived 3: Monitoring enabled but archiving failed
smt	integer	SMT threads used per core
walltime	integer	Requested walltime of job in seconds

Generic Job Meta data structure cont.

Name	Type	Description
jobState	string	Final state of job • completed • failed • cancelled • stopped • out_of_memory • timeout
startTime	integer	Start epoch time stamp in seconds
stopTime	integer	Stop epoch time stamp in seconds
duration	integer	Duration of job in seconds

Generic Job Meta data structure metadata

`metadata` object

- Additional information about the job
- Persisted as JSON column type in SQL
- Additional metadata can be added

Name	Type	Description
jobScript	string	The batch script of the job
jobName	string	Slurm Job name
slurmInfo	string	Additional slurm infos as show by scontrol show job

Generic Job Meta data structure tags

`tags` array

- Tag type to structure tags
- Many-to-many relation in SQL

Name	Type	Description
name	string	
type	string	

Generic Job Meta data structure resources

`resources` array

- Resource used by the job
- Used to query the metric database

Name	Type	Description
hostname	string	
hwthreads	int array	List of OS processor ids
accelerators	string array	List of of accelerator device ids
configuration	string	The configuration options of the node

Generic Job Meta data structure statistics

`statistics` object

- List of named metrics
- Job performance footprint
- Required: **`mem_used`**, **`cpu_load`**, **`flops_any`**, **`mem_bw`**, **`net_bw`**, **`file_bw`**

Name	Type	Description
unit	enum	Metric unit
avg	number	Job: metric average
min	number	Job:: metric minimum
max	number	Job metric maximum

Generic cluster data structure

- Specifies cluster topology and metric thresholds
- Used as configuration for cc-backend and other components

Name	Type	Description
name	integer	
metricDataRepository	object	Type of the metric data repository for this cluster
metricConfig	object	Metric specifications
subClusters	object	Array of cluster hardware partition metric thresholds

Generic cluster data structure subClusters

`subClusters` array of objects

- Array of cluster hardware partitions

Name	Type	Description
name	string	Hardware partition name
processorType	string	Processor type
socketsPerNode	integer	Native measurement
coresPerSocket	integer	Frequency of timeseries points
threadsPerCore	string	How the metric is aggregated (sum, avg, null)
flopRateScalar	integer	name, peak, normal, caution, alert
flopRateSimd	integer	
memoryBandwidth	integer	Node peak memory bandwidth in GB/s
nodes	string	Node list expression

Generic cluster data structure metricConfig

`metricConfig` array of objects

Name	Type	Description
name	string	Metric name
unit	string	Metric unit
scope	string	Native measurement resolution (hwthread, core, memoryDomain, socket, node, accelerator)
timestep	integer	Frequency of timeseries points
aggregation	string	How the metric is aggregated (sum, avg, null)
subClusters	array of objects	name, peak, normal, caution, alert

Generic cluster data structure subcluster:topology

`topology` object within subcluster

- Topology with hwthread domains
- Optional: List of accelerators

Name	Type	Description
node	integer array	HwTread lists of node
socket	array of array of integer	HwTread lists of sockets
memoryDomain	array of array of integer	HwTread lists of memory domains
die	array of array of integer	HwTread lists of dies
core	array of array of integer	HwTread lists of cores
accelerators	array of objects	id, type, model

ClusterCockpit line protocol subset

<measurement>,<tag set> <field set> <timestamp>

- **<tag set>** and **<field set>** comma-separated lists of key=value entries
- **<timestamp>** is Unix epoch time in seconds or nanoseconds

ClusterCockpit specification:

- **<measurement>** : metric name
- Mandatory tags
  - **hostname**
  - **type** (one of node, socket, die, memoryDomain, core, hwthread, accelerator)
  - **type-id** like CPU socket, PCI-E device id, or HW Thread identifier
- Mandatory field: **value** metric value

Job Archive

Directory tree layout

<cluster name>/<level 1>/<level 2>/<start time>

- **<level 1>** jobID/1000
- **<level 2>** jobID%1000
- In directory **<cluster name>** must be one file named **cluster.json**
- In directory **<start time>** the job data consists of two files
  - **meta.json**: Contains job meta information and job statistics
  - **data.json**: Contains complete job data with time series

- Example

For job ID **1034871** the directory path is **./1034/871/<timestamp>/**

REST API endpoints

API Endpoints

- **`[GET] /api/jobs/`** Get list of jobs. Filters applied using query parameters.
- **`[POST] /api/jobs/start_job/`** Add a newly started job
- **`[POST] /api/jobs/stop_job/`** Mark a job as stopped. Which job to stop is specified by the request body.
- **`[POST] /api/jobs/stop_job/{id}`** Mark a job as stopped.
- **`[POST] /api/jobs/tag_job/{id}`** Add a tag to a job
- **`[POST] /api/jobs/import/`** Imports a job and its metric data

https://github.com/ClusterCockpit/cc-specifications/blob/master/interfaces/rest/openapi.yaml

REST API: /api/jobs/start_job/

```
{
"jobId": 12323134,
"user": "user34",
"project": "dfg1390",
"cluster": "titan",
"subCluster": "icelake",
"partition": "singlenode",
"exclusive": 1,
"numNodes": 1,
"startTime": 1624624175,
"resources": [
{ "hostname": "e1245" }]
}
```

Example with curl:

```
$ curl -X 'POST' \
'<URL>/api/jobs/start_job/' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer 9ecf0d4a' \
-d '{<PAYLOAD>}'
```

Return object on success (201):

```
{ "id":  34234567  }
```

REST API: /api/jobs/stop_job/

```
{
"jobId": 12323134,
"cluster": "titan",
"startTime": 1624624175,
"stopTime": 1624624175,
"jobState": "completed"
}
```

Example with curl:

```
$ curl -X 'POST' \
'<URL>/api/jobs/stop_job/' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer 9ecf0d4a' \
-d '{<PAYLOAD>}'
```

Return object on success (201):

```
{ <JOB> }
```