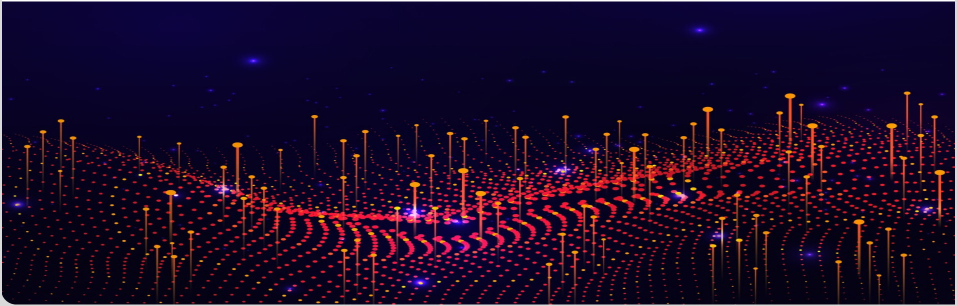
Karlsruhe Institute of Technology

Extended static model of user requests processing for a heterogeneous data aggregation platform

The XXIV International Scientific Conference of Young Scientists and Specialists (AYSS-2020)

Victoria Tokareva | 9-13 November 2020

INSTITUTE FOR ASTROPARTICLE PHYSICS (IAP)

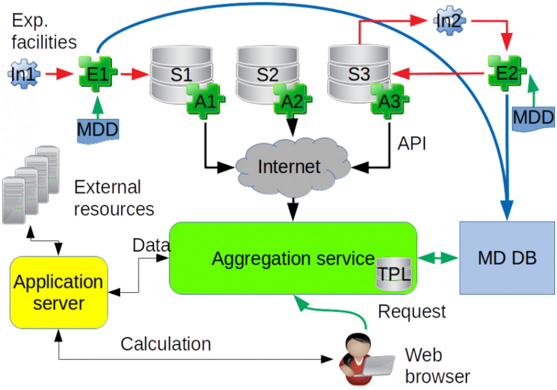


Advanced Planning Systems - APS

Karlsruhe Institute of Technology

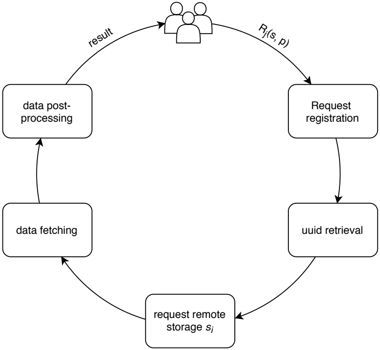
- In industry, APS are management tools, including scheduling modules and supporting environment. The goal is to achieve adaptive control on jobs processing
- APS are developed and customized for each application, taking into account the specific knowledge about the field
- When developing APS for multi-messenger astroparticle physics, one faces double specificity: the first one of astroparticle physics as the field in between particle physics and astronomy and the second one of multi-messenger astronomy, where different messengers determine diverse workflows
- Problems arising in such systems can be solved employing dynamic approaches offered by the queueing theory as well as scheduling (flow shop) techniques

Architecture of the GRADLCI data aggregation platform



- **Si** — local data storages;
- **Ini** — data sources of different types;
- **MDD** — metadata description;
- **Ei** — metadata extractors;
- **Ai** — adapters, provide API for data access;
- **TPL** — template library;
- **MDDB** — metadata database.

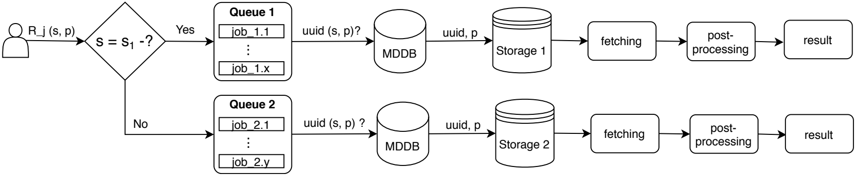
Life cycle of user requests processing



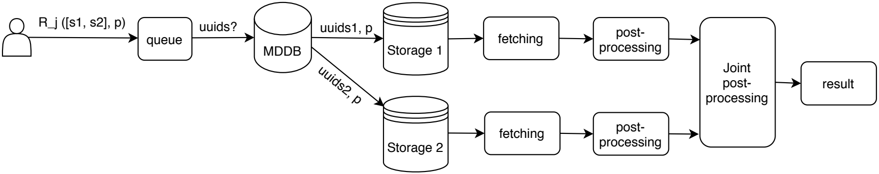
- $R_j([s_1, .., s_i, ...], p)$ are user requests,
- $j \in [0, ..., J]$ job identifier in the system,
- s_i - remote storage identifier, $i \in [1, ..., S]$
- $p \in \mathbb{R}^n$ - other request parameters
- n_{ij} is number of records requested, assuming it is a function $n_{ij}(p)$ of p.

Requests use-cases

Simple request (for case of two storages):

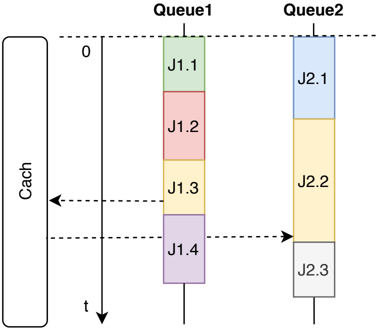


Aggregated request (for case of two storages):

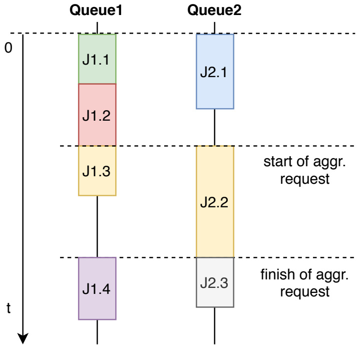


Strategies of request processing (for a case of two remote data storages)

Strategy 1



Strategy 2



Strategy 3: combined

Execution time estimation

- **No estimation**
- **Exact count:** precise number of events, corresponding to parameter p can be requested from metadata storage using $count()$ method. Time t_c required to perform this request is considered to be uniformly distributed in $[0, T_c]$, $T_c \in \mathbb{R}$
- **Approximation by quantiles:**
  can be stored on the aggregation side of arrays with parameter distribution quantiles for events in remote locations and estimate the number of events requested by them
  - \+ instant estimation
  - \- approximation

Requests processing model

Execution time of the request $R_j([s_1, .., s_i, ...], p)$:

$$T_j = min_i(t_c^{ji} + t_q^{ji}) + t_{in} + (\nu + (\tau)^i) \cdot \sum_i n_{ij} + max_i n_{ji} \cdot (max(\mu_i, \tau)) \quad (1)$$

where

1. t_c^{ji} is processing time estimation,
2. $t_q^{ij} = \sum_{k=1}^{j-1} T_k, j = \overline{2, J},\ T_1 \in \mathbb{R}$ is waiting (queueing) time
3. $t_{in} \in unif(0, \Theta_{in})$ is MDDB query initialization time
4. request processing by MDDB is $t_s(n_{ji}) = \nu \cdot n_{ij}, \nu \in \mathbb{R}, i \in \overline{1, N}$
5. fetiching time $t_f(n_{ij}) = \mu_i \cdot n_{ij}, \mu = (\mu_1, ..., \mu_s) \in \mathbb{R}^s$
6. individual post-processing time $t_a(n_{ij}) = \tau \cdot n_{ij}, \tau \in \mathbb{R}$
7. $(\tau)^i$ is joint processing time (for aggregated requests only)

Scheduling algorithms

Exact:

- Branch & bound
- Linear programming
- Dynamic programming
- etc. ...

\+ Proved to find optimal solution for any given problem

\- Most of problems are NP hard: the optimal solution is not guaranteed to be given in reasonable time

\- Scaling issues

Approximate:

- Local search
- Genetic algorithms
- Priority Dispatching Rules
- Decomposition heuristics

\+ Faster

\- Optimal solution is not guaranteed

\- Some algorithms from this group are shown to have scaling issues

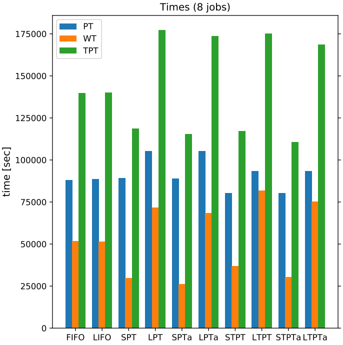
Static Priority Dispatching Rules

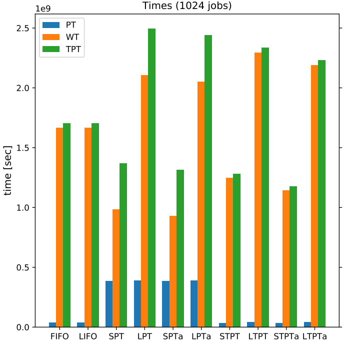
- First In First Out (FIFO)
- Last In First Out (LIFO)
- Shortest Processing Time (SPT)
- Longest Processing Time (LPT)
- Shortest Total Processing Time (STPT)
- Longest Total Processing Time (LTPT)

Criteria:

- Waiting Time (WT) of a job i on machine j queue
- Processing Time (PT): the required time to operate job i on machine j
- Total Processing Time (TPT): total processing time required to achieve job i

Simulation results





65% of generated jobs request only one storage, 35% request two storages The results are averaged by 1000 runs

Outlook

- The mathematical model of jobs processing for distributed heterogeneous data aggregation platform was formulated in terms of flow shop scheduling approach
- Possible solution techniques were under consideration
- For the chosen techniques simulation and comparative analysis were performed
- The best results according to combination of observed criteria is shown by priority dispatching rules STPT(a).

Future plans

- Comparative analysis of dynamic PDRs
- Elaboration of dynamic model based on queue theory
- Development of open APS for astroparticle physics data centers

Thank you for your attention!

victoria.tokareva@kit.edu