Western Norway
University of
Applied Sciences

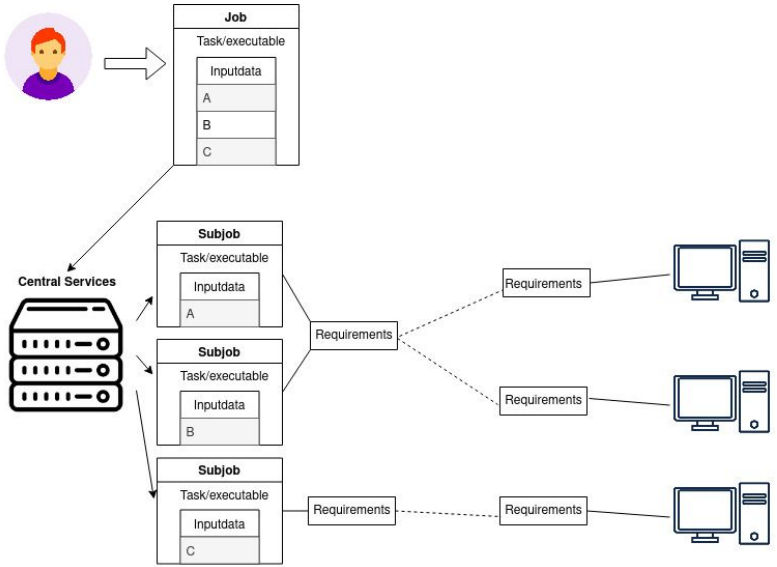
The jAliEn Job optimizer

Haakon André Reme-Ness

ALICE Tier-1/Tier-2 Workshop Seoul, 16.04.2024

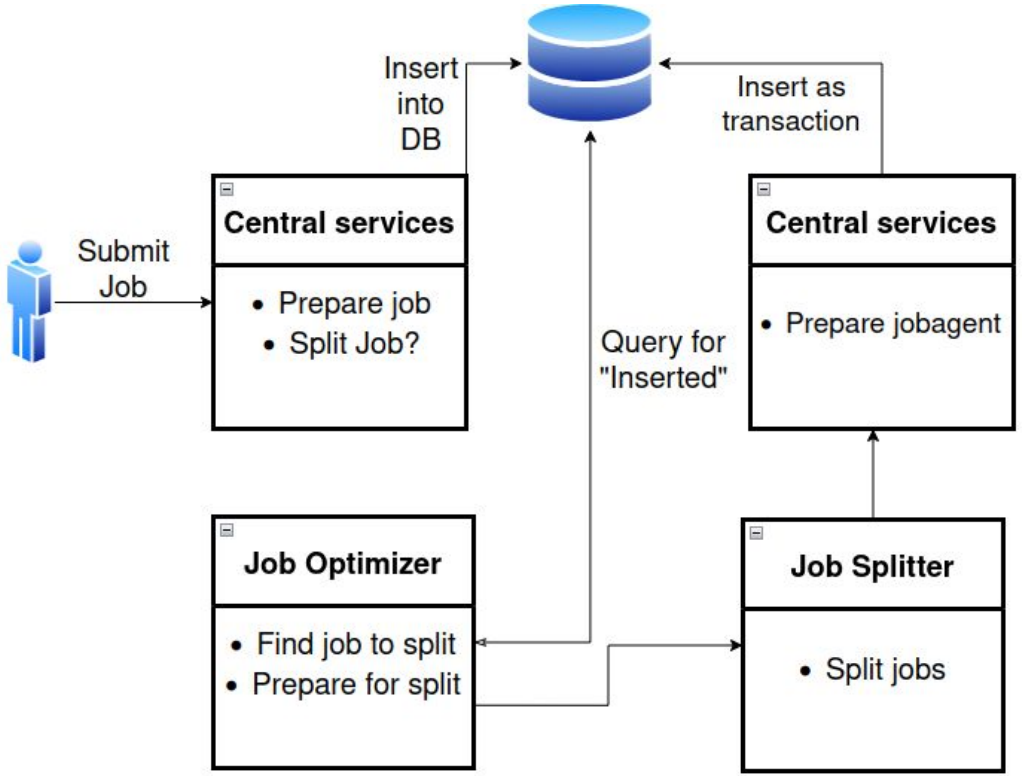
Quick Job Optimizer overview

- What is the Job Optimizer service?
  - Responsible for splitting jobs
  - Started together with other JCentral services
- Continuous service that pick up jobs waiting to be split from grid queue to split into smaller subjobs and insert into grid queue
  - Can run on any central machine
  - Can be turned off from parameter value
  - Split different jobs in parallel if more than one thread is defined



Workflow Job Splitter

- None split jobs are inserted directly to **Waiting**
- Jobs are split instantly if available threads on site
- Job splitter does a lot of upfront work before creating subjobs to use less resources (particularly pattern matching)



ALICE

Western Norway University of Applied Sciences

What is new?

- More checks upfront
  - Example: Unreasonable large XML collections compared to quotas is stopped earlier
- Can no longer cheat quotas
  - Had to change some quotas to adjust for this
- New split functionalities (more on this later)
- Changes to traces/logger
  - Hopefully better for debugging
- Increased performance
  - Focus on database

Database optimization

- Use of indexing as to not reach timeout (5 seconds)
- All inserting of subjobs (including JobAgent) done as a transaction
- Avoiding locks where possible
  - Small tweaks of isolation level for JobAgent inserts
- Batch insert of subjobs
  - Saw noticeable performance increase
  - Unfortunately if one fails to be inserted all fails, and batch must be repopulated
- Bulk lookup in Catalogue for splitting by Storage Element
- Subjob JDL optimization?

Subjob JDL's

- Describe subjobs using delta changes from original JDL
- Implementation is done, but currently in testing
- Is it worth it? Done dynamically?
  - See a decrease in storage used but…
  - Adds a database calls to an already congested table

```
User = "jditzel";
JobTag = {
 "comment:Automatically generated analysis JDL"};
Packages = {
 "VO_ALICE@AliPhysics::vAN-20191202_ROOT6-1",
 "VO_ALICE@APISCONFIG::V1.1x"};
Executable = "/alice/cern.ch/user/h/haakon/LHC18r/297193/myTask.sh";
InputFile = {
 "LF:/alice/cern.ch/user/j/jditzel/LHC18r/297193//myAnalysis.C",
 "LF:/alice/cern.ch/user/j/jditzel/LHC18r/297193//myTask.root",
 "LF:/alice/cern.ch/user/j/jditzel/LHC18r/297193//AliAnalysisTaskDoubleHypNucTree.cxx",
 "LF:/alice/cern.ch/user/j/jditzel/LHC18r/297193//AliAnalysisTaskDoubleHypNucTree.h"};
InputDataList = "wn.xml";
InputDataListFormat = "xml-single";
InputDataCollection = {
 "LF:/alice/cern.ch/user/j/jditzel/LHC18r/297193//000297193.xml,nodownload" };
Split = "se";
SplitMaxInputFileNumber = "15";
JDLPath = "/alice/cern.ch/user/j/jditzel/LHC18r/297193/myOutputDir/myTask.jdl";
JDLArguments = "000297193.xml 000";
ValidationCommand = "/alice/cern.ch/user/j/jditzel/LHC18r/297193/myTask_validation.sh";
OutputDir =
"/alice/cern.ch/user/j/jditzel//LHC18r/297193//myOutputDir/000/#alien_counter_03i#";
Output = {
 "log_archive.zip:std*@disk=1",
 "root_archive.zip:EventStat_temp.root,AnalysisResults.root,*.stat@disk=2" };
Requirements = (
member(other.Packages,"VO_ALICE@AliPhysics::vAN-20191202_ROOT6-1") ) && (
member(other.Packages,"VO_ALICE@APISCONFIG::V1.1x") ) && ( other.TTL > 70000 ) && (
other.Price <= 1 );
TTL = 70000;
Price = 1.0;
MemorySize = "8GB";
 WorkDirectorySize = {
  "5000MB" };

 JDLVariables = {
  "Packages",
  "OutputDir",
  "CPUCores" };
CPUCores = "1";
Type = "Job";
```

Current status

- Currently running on a single machine doing all splitting
  - Ironed out a few bugs and issues, still expecting more…
- Started off too slow, easily tweakable to increase pickup rate of jobs to be split
  - Currently 5 threads splitting simultaneously with 3 seconds cooldown
- Increased Performance
  - Especially for storage element splits

```
Sep 20 10:14:20 [state ]: Job state transition from INSERTING to SPLITTING
Sep 20 10:14:20 [trace ]: [Testing Job Optimizer] Trying to split using new joboptimizer!
Sep 20 10:14:20 [trace ]: Job inserted by alimonitor.cern.ch, request came from alimonitor.cern.ch
Sep 20 10:14:20 [state ]: Job state transition to INSERTING
Sep 20 10:14:20 [trace ]: Using the inputcollection LF:/alice/data/2023/LHC23zs/539649/raw/0310/CTFs.xml,nodownload
Sep 20 10:14:22 [trace ]: Inserted 52 subjobs.
Sep 20 10:14:22 [state ]: Job state transition from SPLITTING to SPLIT
Sep 20 10:14:22 [trace ]: [Testing Job Optimizer] Done splitting and inserting using new joboptimizer!
Sep 20 10:14:23 [trace ]: There are 2574 files in the collection /alice/data/2023/LHC23zs/539649/raw/0310/CTFs.xml (se split job)
Sep 20 10:14:45 [state ]: Job state transition to SPLITTING (user alidaq - 52 baskets)
Sep 20 10:14:45 [trace ]: Starting submission of subjobs...(52 keys)
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926235986
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926235988
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926235990
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236010
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236012
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236013
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236015
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236017
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236020
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236022
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236023
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236024
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236025
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236026
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236027
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236028
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236029
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236030
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236031
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236032
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236033
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236034
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236035
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236037
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236039
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236042
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236044
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236046
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236048
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236050
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236052
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236054
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236056
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236059
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236061
Sep 20 10:14:45 [submit ]: Subjob submitted: 2926236063
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236066
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236118
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236120
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236123
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236125
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236127
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236129
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236131
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236133
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236135
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236137
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236139
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236141
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236143
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236145
Sep 20 10:14:46 [submit ]: Subjob submitted: 2926236147
Sep 20 10:14:46 [state ]: Job state transition from SPLITTING to SPLIT (1)
Sep 21 07:27:33 [state ]: Job state transition from SPLIT to DONE
Sep 21 07:27:33 [state ]: Job token deletion query done for: 2926234836
```

How to use the job splitter?

JDL Split Fields

- JDL field **Split** describes if a job is to be split and the strategy on how to do the splitting
- Split strategies:
  - production
  - file, directory, parent directory
  - se (Storage Element)
  - af (Analysis facilities)
- **#alien#** pattern to be replaced by value during splitting
- **SplitArguments** field is now redundant, but still function as usual
- Other optional or mandatory fields connected to specific split strategies

#alien# argument

- Replace **#alien#** in JDL with corresponding value
  - Based on what type of **#alien#** and subjob, counter most used
- In AliEn **#alien#** was limited to a few fields in JDL, Split Arguments, Outputdir, Output…
- No longer the case, now checks all fields for it!
  - Doing a lot of matches to find and replace **#alien#** arguments in each subjob can be expensive
    - Solution: Do matching only on masterjob -> replace with lambda function
    - Run lambda function with correct input when building subjob JDL
- Technically opens up the possibility of subjobs having different executables

#alien#

How to use:

- **#alien_counter_03i#** —> 001, 002, 003… (**03i** defines how many numbers, this case 3)
  - #alien_counter# —> 1,2,3…
- **#aliendir#** —> /alice/cern.ch/user/h/**hremenes**/LHC22f3.xml
- **#alienfulldir#** —> **/alice/cern.ch/user/h/hremenes/LHC22f3.xml**
- **#alienfilename/oldvalue/newvalue/#**
  - **#alienfilename/.xml/.new/# –>** /alice/cern.ch/user/h/hremenes/**LHC22f3.xml** -> /alice/cern.ch/user/h/hremenes/**LHC22f3.new**
- Use **first**, **last** or **all** in front to choose which inputdata file for the subjobs to use for the options above (Example: **#alienfirstdir#**)
  - **First** is default, uses first inputdata file in subjob, **last** the opposite
  - **All** uses all inputdata joined together with a "**,**" as delimiter

Production strategy

- Duplicate job a number of times equal to the range given
  - Monte carlo simulations
- Subjobs remain mostly the same (**#alien_counter#** will give small differences)
  - #alien_counter# now correctly follows range provided
    - Example: **production:20-30** will ensure the counter starts at 20

How to use:

- **Split = “production:1-100”**

File strategy

- Inputdata files are split based on their full LFN (Logical File Name)
  - Since LFN's are unique, this ensures that there is exactly one inputdata file per subjob

<u>How to use:</u>

- **Split = "file"**

directory strategy

- Inputdata files are split based on directory LFN (Logical File Name)
  - **/alice/cern.ch/user/h/hremenes/**LHC22f3.xml

How to use:

- **Split = “directory”**

Optional

- **SplitMaxInputFileNumber ->** Maximum number of inputdata files per subjob
- **SplitMaxInputFileSize ->** Maximum size of all inputdata files per subjob

parentdirectory strategy

- Inputdata files are split based on parent directory LFN (Logical File Name)
  - **/alice/cern.ch/user/h/**hremenes/LHC22f3.xml

How to use:

- **Split = “parentdirectory”**

Optional

- **SplitMaxInputFileNumber ->** Maximum number of inputdata files per subjob
- **SplitMaxInputFileSize ->** Maximum size of all inputdata files per subjob

SE strategy

- Group inputdata based on locality
  - Grouping is done with inputdata files sharing all SE's
- Bulk lookup improved this significantly
- Introduced merging of smaller subjobs
  - Some subjobs had 1 inputdata file, while others had in the hundreds
  - Default value on when subjobs should be merged, but can be set manually

SE strategy

How to use:

- **Split = “se”**
- **SplitMaxInputFileNumber ->** Maximum number of inputdata files per subjob (must be set)

Optional

- **SplitMinInputFileNumber ->** Minimum number of inputdata files per subjob (default is ⅕ of max)

AF strategy

- Mainly for analysis use
- Two different ways to use it
  - Default is just setting a maximum number of input files or size, and will then split accordingly
    - Together **CloseSE** requirement run jobs on AF that are evenly split
  - Other option is setting **ForceOnlySEInput**, ensuring only inputdata files that are available on site is used (not found files are ignored)
    - Can also set a threshold for percentage of missing files before this job fails (default is 10%)
    - Currently looking at only **CloseSE** in requirements to use for forcing SE
      - In the future will also look at **CloseCE,** currently being implemented
- Already in production, let me know if it is not working or other use cases not covered

AF strategy

How to use:

- **Split = “af”**
- **SplitMaxInputFileNumber ->** Maximum number of inputdata files per subjob

or

- **SplitMaxInputFileSize ->** Maximum size of all inputdata files per subjob

Optional

- **ForceOnlySEInput ->** Ignore inputdata files not found on SE
- **MaxInputMissingThreshold ->** Set percentage of ignored inputdata files before job fails (default 10%)

Going forward

- Push everything, have several machines running the splitter
- This will also include improvements regarding original masterjob submission that were mentioned earlier
- Finish testing for subjobs JDL delta change and maybe push it
- Finish AF splitting, improve on it based on feedback

Email: haakon.andre.reme-ness@cern.ch