

opt_jr_doc

Generated by Doxygen 1.8.5

Sat Dec 9 2017 12:56:25

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Application Class Reference	5
3.2	Batch Class Reference	7
3.3	Bounds Class Reference	8
3.4	Candidate Class Reference	8
3.5	ObjFun Class Reference	9
3.6	optJrParameters Class Reference	9
3.7	sAlphaBetaManagement Class Reference	10
3.8	Search Class Reference	10
3.9	slastSimulatorRun Class Reference	10
4	File Documentation	11
4.1	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/main.cpp File Reference	11
4.1.1	Function Documentation	11
4.1.1.1	main	12

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Application	5
Batch	7
Bounds	8
Candidate	8
ObjFun	9
optJrParameters	9
sAlphaBetaManagement	10
Search	10
slastSimulatorRun	10

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

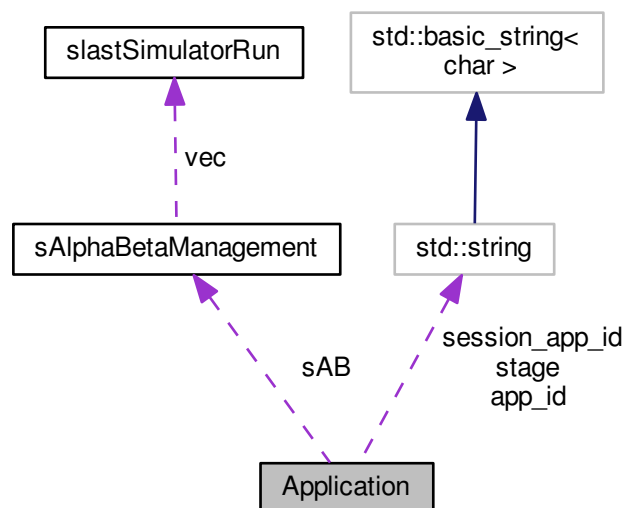
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/appByWeight.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/db.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/debugmessage.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor_helper.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/main.cpp	11
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParam_helper.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/read_app_file.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/readConfigurationFile.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.hh	??
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/utility.hh	??

Chapter 3

Class Documentation

3.1 Application Class Reference

Collaboration diagram for Application:



Public Member Functions

- **Application** (std::string session_app_id, std::string app_id, double w, double chi_0, double chi_C, double m, double M, double V, double v, double D, double csi, std::string St, int DatasetSize)
- double **ObjFunctionComponent** (sConfiguration &configuration, MYSQL *conn, [optJrParameters](#) &par)

Public Attributes

- int **mode**
- std::string **session_app_id**
- std::string **app_id**

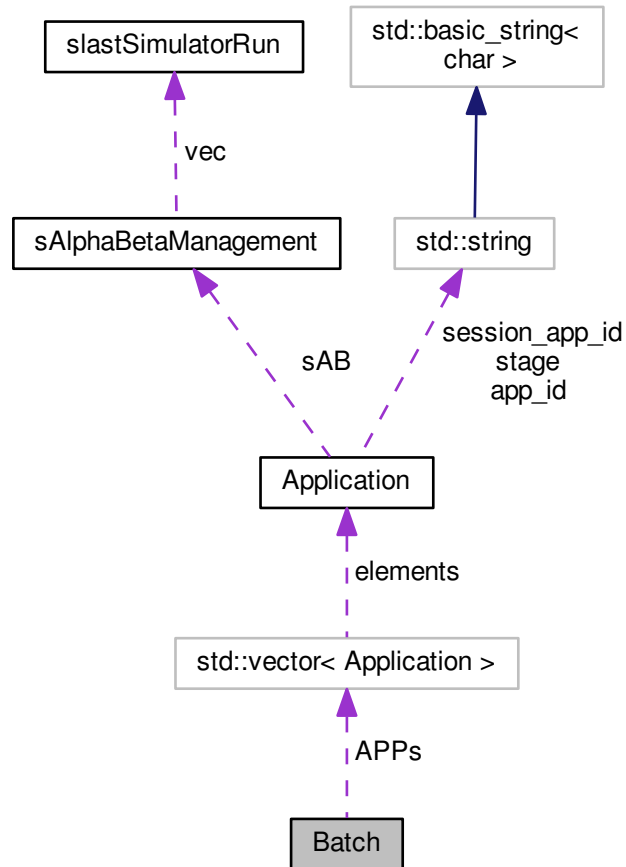
- double **w**
- double **term_i**
- double **chi_0**
- double **chi_C**
- double **m**
- double **M**
- double **V**
- double **v**
- double **Deadline_d**
- double **csi**
- std::string **stage**
- int **datasetSize**
- double **nu_d**
- int **currentCores_d**
- int **nCores_DB_d**
- int **bound**
- double **R_d**
- double **R_bound_d**
- double **baseFO**
- double **initialBaseFO**
- float **alpha**
- float **beta**
- [sAlphaBetaManagement](#) **sAB**
- int **boundIterations**
- int **vm**

The documentation for this class was generated from the following files:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.cpp

3.2 Batch Class Reference

Collaboration diagram for Batch:



Public Member Functions

- **Batch** (`std::vector< Application > apps`)
- void **calculate_nu** (`optJrParameters &par`)
- void **initialize** (`sConfiguration &configuration`, `MYSQL *conn`, `optJrParameters &par`)
- void **fixInitialSolution** (`optJrParameters &par`)
- sCandidates **approximatedLoop** (`int &iteration`, `optJrParameters &par`)

Public Attributes

- `std::vector< Application > APPs`

The documentation for this class was generated from the following files:

- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.hh`
- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.cpp`

3.3 Bounds Class Reference

Static Public Member Functions

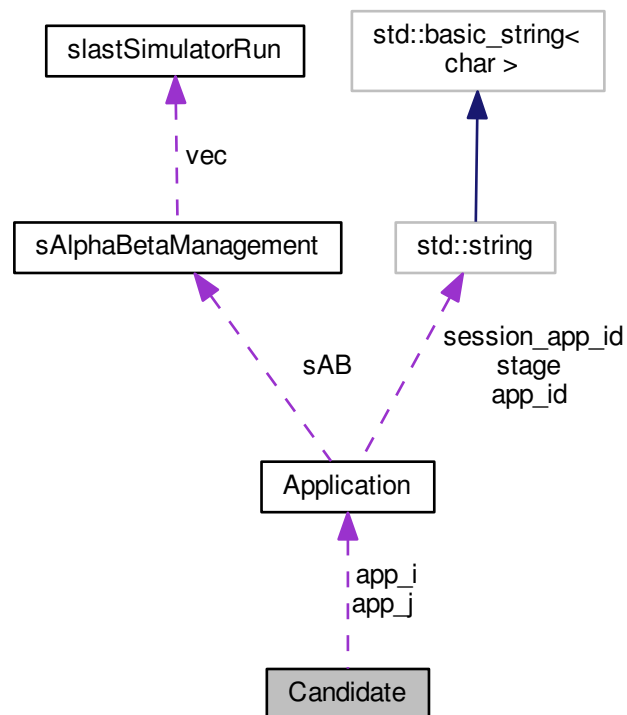
- static void **calculateBounds** ([Batch](#) &app_manager, int n_threads, sConfiguration &configuration, MYSQL *conn, [optJrParameters](#) &par)

The documentation for this class was generated from the following files:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.hh
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.cpp

3.4 Candidate Class Reference

Collaboration diagram for Candidate:



Public Member Functions

- Candidate** ([Application](#) *i, [Application](#) *j, int NCi, int NCj, double D_FO, int d_i, int d_j)

Public Attributes

- [Application](#) * **app_i**

- int **newCoreAssignment_i**
- int **delta_i**
- double **real_i**
- [Application](#) * **app_j**
- int **newCoreAssignment_j**
- int **delta_j**
- double **real_j**
- int **nodes_i**
- int **nodes_j**
- double **deltaFO**

The documentation for this class was generated from the following file:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.hh

3.5 ObjFun Class Reference

Static Public Member Functions

- static double **ObjFunctionComponent** (sConfiguration &configuration, MYSQL *conn, [Application](#) &app, [optJrParameters](#) &par)
- static double **ObjFunctionComponentApprox** ([Application](#) &App, [optJrParameters](#) &par)
- static double **ObjFunctionGlobal** (sConfiguration &configuration, MYSQL *conn, [Batch](#) &App_manager, [optJrParameters](#) &par)

The documentation for this class was generated from the following files:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.hh
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.cpp

3.6 optJrParameters Class Reference

Public Member Functions

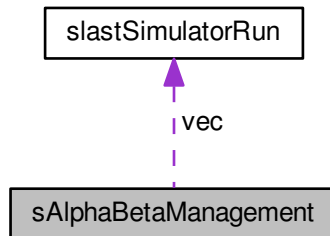
- **optJrParameters** (char **args, int argc)
- const std::string **get_filename** ()
- const int **get_debug** ()
- const int **get_cache** ()
- const int **get_globalFOcalculation** ()
- const int **get_K** ()
- const int **get_simulator** ()
- const int **get_number** ()
- const int **get_maxIteration** ()

The documentation for this class was generated from the following files:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.hh
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrparameters.cpp

3.7 sAlphaBetaManagement Class Reference

Collaboration diagram for sAlphaBetaManagement:



Public Attributes

- [slastSimulatorRun](#) **vec** [HYP_INTERPOLATION_POINTS]
- int **index**

The documentation for this class was generated from the following file:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh

3.8 Search Class Reference

Static Public Member Functions

- static void **localSearch** (sConfiguration &configuration, MYSQL *conn, [Batch](#) &App_manager, [optJr-Parameters](#) &par)

The documentation for this class was generated from the following files:

- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.hh
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.cpp

3.9 slastSimulatorRun Class Reference

Public Attributes

- int **nCores**
- double **R**

The documentation for this class was generated from the following file:

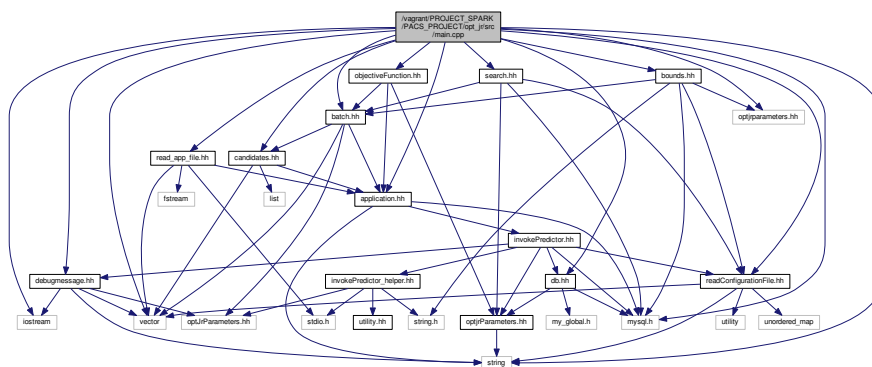
- /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh

File Documentation

4.1 /vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/main.cpp File Reference

```
#include <iostream>
#include <string>
#include <mysql.h>
#include <vector>
#include "optjrparameters.hh"
#include "readConfigurationFile.hh"
#include "debugmessage.hh"
#include "db.hh"
#include "application.hh"
#include "read_app_file.hh"
#include "batch.hh"
#include "bounds.hh"
#include "search.hh"
#include "objectiveFunction.hh"
#include "candidates.hh"
```

Include dependency graph for main.cpp:



Functions

- int **main** (int argc, char **argv)

4.1.1 Function Documentation

4.1.1.1 `int main (int argc, char ** argv)`

- 1) Read parameters from command line and save them in an "optJrParameters" object
- 2) read informations from "wsi_config.xml" file and save it in a "sConfiguration" object (which is unordered_map(string,string))
- 3) Connect to the Database
- 4) Open *.csv file with Applications data, and save it in a "Batch" object
- 5) Calculate bounds for each application loaded (with the calculateBounds method of [Bounds](#) class)
- 6) Calculate nu indices for each application (with the calculate_nu method of [Batch](#) class)
- 7) Fix initial solution (with the fixInitialSolution method of [Batch](#) class)
- 8) Initialize Objective Function evaluation for each application (with the initialize method of [Batch](#) class)
- 9) Find an "optimal" solution invoking "localSearch" method (of "Search" class)