

JobInProgress

[JobInProgress](#) // All information should be exported to monitor system.

Path jobFile;

Path localJobFile;

Path localJarFile;

~~LocalFileSystem localFs; // need higher level abstraction. decoupled from particular fs.~~

~~int restartCount; // not use.~~

long startTime;

~~long launchTime; // not use.~~

long finishTime;

int maxTaskAttempts;

String jobName;

BSPJobID jobId;

~~BSPMaster master; // only store job info.~~

[TaskInProgress](#) tasks[]; // will be replaced by o.a.h.b.v2.Task where stores a task to groom relation.

long superstepCounter;

Counters counters; // See [Counters In Action](#)

int numBSPTasks;

~~int clusterSize; // not use~~

~~String jobSplit; // dir where files are split.~~

String master; // newly added, indicating which master currently deals with this job. // job to master relation is many to one

Path inputPath; // newly added. input dir. should be the same as jobSplit.

[JobProfile](#)

String user

~~BSPJobID jobId // redundant.~~

~~String jobFile // redundant.~~

~~String jobName // redundant.~~

[JobStatus](#)

~~BSPJobID jobId;~~

long progress; // % of bsp()

long cleanupProgress; // % of cleanup()

long setupProgress; // % of setup()

State state; // runState in enum

~~int runState; // redundant.~~

~~long startTime; // redundant.~~

~~String schedulingInfo = "NA"; // not use~~

~~String user; // redundant.~~

~~long superstepCount; // redundant to superstepCounter.~~

~~String name; // not use~~

~~int tasks; // can be calculated.~~

~~long finishTime; // redundant.~~

~~Counters counter; // redundant.~~