

## Setting up the test

download the test jobs:

```
cd src/
mkdir Tests
cd Tests/
cvs co -d TestJobs UserCode/leo/Utilities/PerfToolkit/TestJobs/
cd TestJobs/
scram b
```

## Testing

### CRAB+CMSSW+SYS

Check that in your test dir the following files are available: `jobscript_cmssw.sh` `net.sh`

Then, create a crab cfg like:

```
[CRAB]
jobtype = cmssw
scheduler = glite
[CMSSW]
datasetpath=/RelValProdTTbar/JobRobot-MC_3XY_V24_JobRobot-v1/GEN-SIM-DIGI-RECO
pset=JPE.py
total_number_of_events=-1
events_per_job = 50000
output_file = cmssw_net.log, cmssw_vmstat.log, cmssw.xml, cmssw.stdout
[USER]
return_data = 1
ui_working_dir = Site.T2_CH_CSCS-Cfg.JPE-Dataset.RelValProdTTbarJobRobotMC_3XY_V24_JobRobotv1-Eve
b.50000-Sw.CMSSW_3_6_0_pre5-Date.201005041858
additional_input_files = net.sh
script_exe=jobscript_cmssw.sh
copy_data = 0
publish_data=0
publish_data_name = name_you_prefer
[GRID]
rb = CERN
se_black_list = T0,T1
```

If you want to use some scripts, use the `crab.template` file and the `crab_LaunchIOTestJobs.py` file, like:

```
crab_LaunchIOTestJobs.py T3_CH_PSI CMSSW_3_7_0_pre4_Brian2nd JPE.py 50000 /RelValProdTTbar/JobRob
```

**TODO:** update the script on CVS

### CMSSW+SYS

If you want to test single jobs (eg from a dedicated WN), then crab is not an option. Use, instead, the `jobscript_standalone_cmssw.sh` script, eg:

```
./jobscript_standalone_cmssw.sh JPE
```

please check the `EVENTS` and `SW` variables:

```
$ cat jobscript_standalone_cmssw.sh
```

```
#!/bin/bash

CFG=$1
SW=CMSSW_3_7_0_pre4_Brian2nd
LOG="cmssw"
EVENTS=50000

DIR=Site.T3_CH_PSI-Cfg.${CFG}-Dataset.RelValProdTbarJobRobotMC_3XY_V24_JobRobotv1-EventsJob.${EVENTS}
mkdir $DIR

#eval `scram ru -sh`

vmstat -nd 10 &> ${DIR}/${LOG}_vmstat_1.log &
PIDSTAT=$!
./net.sh ${DIR}/${LOG}_net_1.log &
PIDWATCH=$!
sleep 60
( /usr/bin/time cmsRun -j ${DIR}/${LOG}_1.xml ${CFG}.py ) &> ${DIR}/${LOG}_1.stdout
kill -9 $PIDSTAT $PIDWATCH
```

## Analyzing the results

First of all, download the proper scripts:

```
cvs co -d PerfToolKit UserCode/leo/Utilities/PerfToolkit/cpt_getJobInfo.py
cvs co -d PerfToolKit UserCode/leo/Utilities/PerfToolkit/cpt_getStats.py
cvs co -d PerfToolKit UserCode/leo/Utilities/PerfToolkit/cpt_utilities.py
cvs co -d PerfToolKit/plugins UserCode/leo/Utilities/PerfToolkit/plugins
```

### Example 1: CRAB+CMSSW+SYS

The first step is to create the rootfiles containing the needed information:

```
$ python PerfToolKit/cpt_getJobInfo.py --type=CMSSWCRAB Site.T2_CH_CSCS-Cfg.JPE-Dataset.RelValPro
```

--type identifies the workflow you've used. It can take the values:

- CRAB: a CMSSW job sent through CRAB
- CMSSW: a CMSSW job executed stand alone
- CMSSWCRAB: a stand-alone CMSSW jobs executed (through a script) with CRAB

Then, we need to create the tables and the graphs. The script is `cpt_getStats.py` and takes the following arguments:

- The list of rootfiles (separated by a space) which contains the information. \* and ? wildcards are supported
- = --save-png= : Saves created histos in png format
- --save-root: Saves created histos in a ROOT file. If enabled, these histos will be not drawn on screen
- --no-auto-bin: Disables automatic histo binning
- --binwidth-time=BINWIDTHTIME: Bin width of time histos in seconds
- --no-plots: Do not draw plots, only outputs the summary tables
- --label=LABEL: Label to be used in naming plots, etc
- --mode=MODE: Preconfigured modes for analysis: SiteMon, SiteMonExt, SiteCfrExt, Default (default value 😊). This drives which quantities are examined and the output style

For example, to perform a Site monitoring during time:

```
$ python PerfToolKit/cpt_getStats.py --mode=SiteMonExt *CSCS*.root
```

The behaviour of different "modes" can be configured in the `setCPTMode(mode)` function defined in `cpt_utilities.py`. **Warning:** some histograms may be not plotted when the contained values are too small (e.g. User time~ 10 secs). You can try setting a more fine-grained binwidth, e.g. `--binwidth-time=5`

-- LeonardoSala - 17-May-2010

---

This topic: Sandbox > LSCPTNew

Topic revision: r4 - 2010-06-21 - unknown



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)