

Table of Contents

Using TOTEM Config Splitter Improved (TOTCSI).....	1
Table of contents.....	1
Prerequisites.....	1
Map-reduce approach.....	1
Fetching needed files.....	2
Configuring TOTCSI.....	3
.....	3
Running the reconstruction splitting.....	4
Step 1 - split.....	4
Step 2 - check map.....	5
Step 3 - check reduce.....	5
Step 4 - submit map.....	5
Step 5 - check map output.....	5
Step 6 - resubmit map.....	6
Step 7 - submit reduce.....	6
Step 8 - check reduce output.....	6
Step 9 - resubmit reduce.....	6
Running the simulation splitting.....	6
Step 1 - split.....	6
Step 2 - check map.....	7
Step 3 - check reduce.....	7
Step 4 - submit map.....	7
Step 5 - check map output.....	7
Step 6 - resubmit map.....	8
Step 7 - submit reduce.....	8
Step 8 - check reduce output.....	8
Step 9 - resubmit reduce.....	8

Using TOTEM Config Splitter Improved (TOTCSI)

The document focuses on using TOTCSI for easy splitting and submitting the jobs to computer cluster.

Table of contents

1. Prerequisites
2. Map-reduce approach
3. Fetching needed files
4. Configuring TOTCSI
 1. Mandatory changes in TOTCSI configuration
 2. Mandatory changes in job template
5. Important notice
6. Running the reconstruction splitting
 1. Step 1 - split
 2. Step 2 - check map
 3. Step 3 - check reduce
 4. Step 4 - submit map
 5. Step 5 - check map output
 6. Step 6 - resubmit map
 7. Step 7 - submit reduce
 8. Step 8 - check reduce output
 9. Step 9 - resubmit reduce
7. Running the simulation splitting
 1. Step 1 - split
 2. Step 2 - check map
 3. Step 3 - check reduce
 4. Step 4 - submit map
 5. Step 5 - check map output
 6. Step 6 - resubmit map
 7. Step 7 - submit reduce
 8. Step 8 - check reduce output
 9. Step 9 - resubmit reduce

Prerequisites

- sufficiently large directory on AFS (more than 700MB) for CMSSW and TOTCSI workspace
 - ◆ can be in TOTEM scratch space (/afs/cern.ch/exp/totem/scratch/)
 - ◆ user's work directory (i.e. /afs/cern.ch/work/l/lgrzanka) works fine aswell (it can have up to 100GB of space)
- jobs have to be submitted to lxbatch cluster from lxplus machines

Map-reduce approach

Map-reduce approach has two main parts:

- Map
 - ◆ apply-to-all function, performing a given operation on each element of an input list
 - ◆ often gives the possibility to parallelize computing
 - ◆ it is possible thanks to the splitting of the configurations

- ◆ example: for each job configuration (input) submit it to LFS and fetch (operation) the results (output)
- Reduce
 - ◆ fold function, performing a given combine operation on the input list (which is often the result of a Map function)
 - ◆ almost always sequential
 - ◆ at most times it is a merge operation
 - ◆ example: merge (operation) given .root files (input) into single one (output)

Picture below shows the map-reduce in simulation example.

Fetching needed files

- Fetching CMSSW into user work private directory

```
cd /afs/cern.ch/work/i/ijurkows/private
svn co svn+ssh://svn.cern.ch/repos/totem/trunk/offline/cmssw/src CMSSW_4_2_4/src
```

- Follow instructions here to compile CMSSW
- You can use TOTCSI as a service or copy it to your AFS directory
 - ◆ Using the TOTCSI as a service
 - ◇ All calls should be directed through the

```
/afs/cern.ch/exp/totem/soft/TOTCSI/totcsi
```

- ◆ Fetching TOTCSI into user work private directory (for developers)
 - ◇ For users:

```
cp -Lrf /afs/cern.ch/exp/totem/soft/TOTCSI config_splitter
```

◇ For developers:

· Project trunk:

```
svn co svn+ssh://svn.cern.ch/repos/totem/trunk/offline/cmssw/tools/conf
```

- Tag for version TOTCSI 1.0:

```
svn co svn+ssh://svn.cern.ch/repos/totem/tags/TOTCSI/TOTCSI_1_0 config_
```

- Tag for version TOTCSI 1.1:

```
svn co svn+ssh://svn.cern.ch/repos/totem/tags/TOTCSI/TOTCSI_1_1 config_
```

Configuring TOTCSI

You can find the example configurations (basic usage) by going into:

```
cd config_splitter/examples/configurations
```

And the example templates for CMSSW by going into:

```
cd config_splitter/examples/templates
```

.

Some parts of configuration files (mostly paths) have to be configured by user so that TOTCSI can work properly.

- Type of task user wants to do (either "Reconstruction" or "Simulation")

```
config.task_type
```

- Path to compiled CMSSW should be set according to user's preferences (the path from example configurations won't work).

```
config.cmssw_dir
```

- Paths to directories for map output and reduce output.

```
config.map_output  
config.reduce_output
```

- Paths to map and reduce templates.

```
config.input_config.map_path  
config.input_config.reduce_path
```

- Path to main workspace directory.

```
config.workspace.root_dir
```

Some fragment of the job configuration file have to be changed by user in order to create a template that TOTCSI understand.

- `{{number_of_events}}` tag - will be replaced by the number declared in `config.simulation.number_of_events` (simulation) or `config.reconstruction.events_per_file_to_reconstruct` (reconstruction). One should change

```
process.maxEvents = cms.untracked.PSet(
  input = cms.untracked.int32(-1)
)
```

into

```
process.maxEvents = cms.untracked.PSet(
  input = cms.untracked.int32({{number_of_events}})
)
```

- **{{outputname("name_base")}}** (for standard output files) and **{{outputNtuple_name("name_base")}}** (for NTuple output files) - will be replaced by the output file names generated for each job. One should change

```
process.TotemNtuplizer.outputFileName = "ntuple_8372.root"
```

to

```
process.TotemNtuplizer.outputFileName = "{{outputNtuple_name("TotemNTuple")}}"
```

- **{{input}}** - will be replaced by the appends to process.source.fileNames. One should change

```
process.source.fileNames.append("/castor/cern.ch/user/r/rlazarz/TOTCSI/map/reco_8372.part1
```

to

```
{{input}}
```

- **{{skipped_events}}** - will be replaced by the number of events to be skipped (counting from the beginning of a file. Following line should be added in configuration template after cms.Source initialization:

```
process.source.skipEvents = cms.untracked.uint32({{skipped_events}})
```

One workspace (directory) can be used only for one reconstruction/simulation! Trying to use same workspace for different configurations at the same time (e.g. launching submit map for one configuration and right after the jobs are submitted doing the same for the second configuration) is very possible to cause errors in TOTCSI work!

If you want to use directory for subsequent reconstruction/simulation it is highly recommended to first clean the directory.

Running the reconstruction splitting

This is the step-by-step simple reconstruction procedure (first the map phase, then the reduce phase). We will be using the files from examples folder. For the more advanced functionality please read this .

Step 1 - split

Go into main config_splitter folder and split the configurations. As we use the example configuration we have to specify it's location by using -c option (or alternatively --config= option)

```
cd config_splitter
./totcsi split -c examples/configurations/totcsi_configuration_reconstruction.py
```

Alternatively use shorten version of **split** command: **sp**.

Step 2 - check map

Optional step!

This checks the integrity of splitted configurations by first importing them and then running local cmsRun for a lighter (only few events to reconstruct) configuration for map phase.

```
./totcsi check_map
```

Alternatively use the shorter command **cm**

Step 3 - check reduce

Optional step!

This should be only done after completing Step 2, otherwise the reduce (merge) phase will have no input files!

This checks the integrity of splitted configurations by first importing them and then running local cmsRun for a lighter (only few events to reconstruct) configuration for reduce phase.

```
./totcsi check_reduce
```

Alternatively use the shorter command **cr**

Step 4 - submit map

After the splitting is done you can run the **submit_map** command (alternatively use shorten version of command: **sum**).

```
./totcsi submit_map
```

Check if jobs were sent to computing cluster with **bjobs** command.

```
bjobs
```

Step 5 - check map output

Optional step!

Wait for the jobs to finish (you can check if there any jobs left by **bjobs** command). Go to directories specified in the `./examples/configurations/totcsi_configuration_reconstruction.py` List output files with **ls** command and see if all files that should be produced, were produced.

For automatic checking of the output use **check_output** command (alternatively use shorten version of command: **co**).

```
./totcsi check_output
```

This should inform if any files were not produced and gives TOTCSI a list of jobs to be resubmitted (for the **resubmit** command).

Step 6 - resubmit map

Optional step!

This can be only done after completing Step 5

If there were some errors and some files weren't produced you can resubmit the jobs using **resubmit** command (alternatively use shorten version of command: **res**).

```
./totcsi resubmit
```

Step 7 - submit reduce

Wait for the jobs to finish and then submit the reduce phase.

```
./totcsi submit_reduce
```

Alternatively use shorten version of **submit_reduce** command: **sur**.

This should produce reduce job on the cluster.

Step 8 - check reduce output

Proceed with same commands as in step 5.

Step 9 - resubmit reduce

Proceed with same commands as in step 6.

Running the simulation splitting

This is the step-by-step simple simulation procedure (first the map phase, then the reduce phase). We will be using the files from examples folder.

It is almost the same as reconstruction procedure, there are only few minor changes.

Step 1 - split

Go into main config_splitter folder and split the configurations. As we use the example configuration we have to specify it's location by using **-c** option (or alternatively **--config=** option)

```
cd config_splitter
chmod 755 totcsi
./totcsi split -c examples/configurations/totcsi_configuration_simulation.py
```

Alternatively use shorten version of **split** command: **sp**.

Step 2 - check map

Optional step!

This checks the integrity of splitted configurations by first importing them and then running local cmsRun for a lighter (only few events to reconstruct) configuration for map phase.

```
./totcsi check_map
```

Step 3 - check reduce

Optional step!

This should be only done after completing Step 2, otherwise the reduce (merge) phase will have no input files!

This checks the integrity of splitted configurations by first importing them and then running local cmsRun for a lighter (only few events to reconstruct) configuration for reduce phase.

```
./totcsi check_reduce
```

Alternatively use the shorter command **cr**

Step 4 - submit map

After the splitting is done you can run the **submit_map** command (alternatively use shorten version of command: **sum**).

```
./totcsi submit_map
```

Check if jobs were sent to computing cluster with **bjobs** command.

```
bjobs
```

Step 5 - check map output

Optional step!

Wait for the jobs to finish (you can check if there any jobs left by **bjobs** command). Go to directories specified in the `./examples/configurations/totcsi_configuration_reconstruction.py` List output files with **ls** command and see if all files that should be produced, were produced.

For automatic checking of the output use **check_output** command (alternatively use shorten version of command: **co**).

```
./totcsi check_output
```


This should inform if any files were not produced and gives TOTCSI a list of jobs to be resubmitted (for the **resubmit** command).

Step 6 - resubmit map

Optional step!

This can be only done after completing Step 5

If there were some errors and some files weren't produced you can resubmit the jobs using **resubmit** command (alternatively use shorten version of command: **res**).

```
./totcsi resubmit
```

Step 7 - submit reduce

Wait for the jobs to finish. After all jobs are finished you can submit the reduce phase.

```
./totcsi submit_reduce
```

Alternatively use shorten version of **submit_reduce** command: **sur**.

This should produce reduce job on the cluster.

Step 8 - check reduce output

Proceed with same commands as in step 5.

Step 9 - resubmit reduce

Proceed with same commands as in step 6

This topic: TOTEM > CompOfflineTOTCSlv1TechnicalDocumentation

Topic revision: r2 - 2013-02-11 - IgorJurkowski



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