



ToPoS: a Token Pool Server

Pieter van Beek
SARA Computing and Networking Services
High Performance Computing and Visualization
e-Science Support

What is Jan trying to do?

- ▶ Run KCSmart on various datasets
- ▶ with various parameter values:
 - ▶ 5 parameters (AKA “dimensions”);
 - ▶ non-linear parameter sweeps.
- ▶ Each run is ± 20 minutes.
- ▶ This week: 6240 “runs” (± 2000 hours) in 1½ day

The EGEE/GLite Way

- ▶ The WMS allows simple parameter sweeps
 - ▶ but only with constant parameter value increments
 - ▶ and cubic parameter spaces.

- ▶ Solutions:
 - ▶ Create a “parameter file” with n lines:

```
1.0 1.0 0.02 0 20000 20
1.5 1.0 0.02 0 20000 20
1.5 1.5 0.02 0 20000 20
2.0 1.0 0.02 0 20000 20
:
:
```

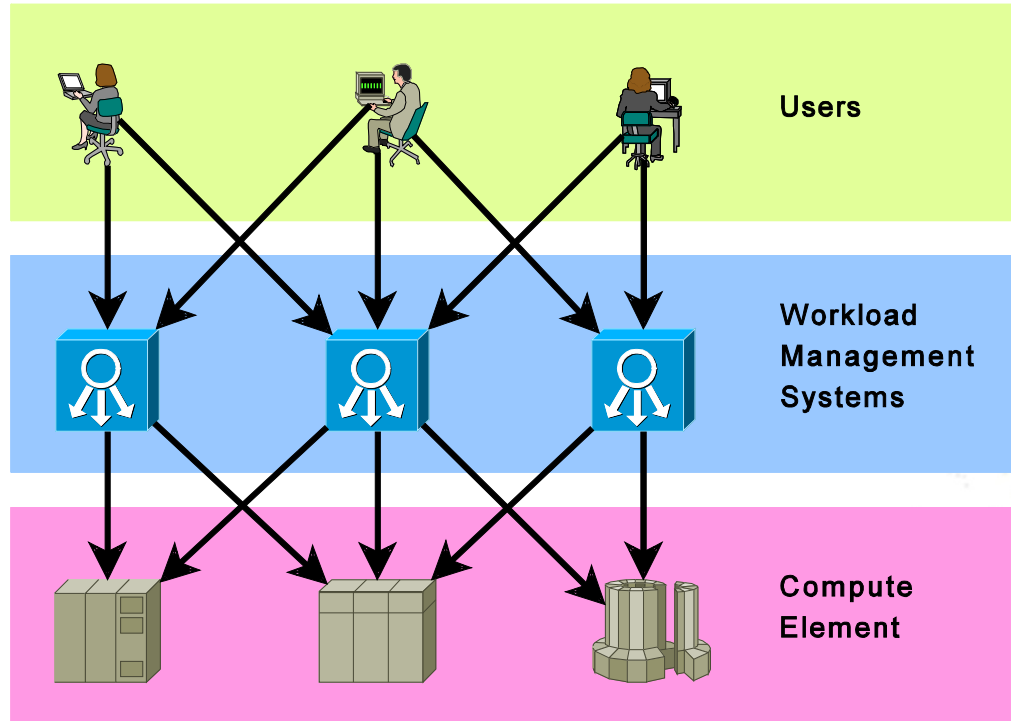
and have one parameter *line_number* $\in 1..n$
 - ▶ Create n different .jdl files.

- ▶ and then: just submit! Uhm... well, no ;-(

Users experiences with gLite

- ▶ **Overhead for starting jobs is considerable**
- ▶ **Determining the best *chunk size* is difficult.**
 - ▶ Too small -> large overhead
 - ▶ Too large -> timeouts and throughput problems.
- ▶ **Resource brokering is far from optimal**
- ▶ **Jobs often fail**

Resource Brokering is Broken



Failing Jobs (1)



Common experiences:

- Sorry, an Incomprehensible Error occurred: 0x3F1D
 - Your VOMS Credential has expired
 - What Job?
 - Success! (but there's no output)
 - Failure! (but it ran just fine)
 - Out of Wall-time (but no CPU-time?)
-
- A lot of “monitoring and resubmission” software is created again and again by many users.

Failing Jobs (2)



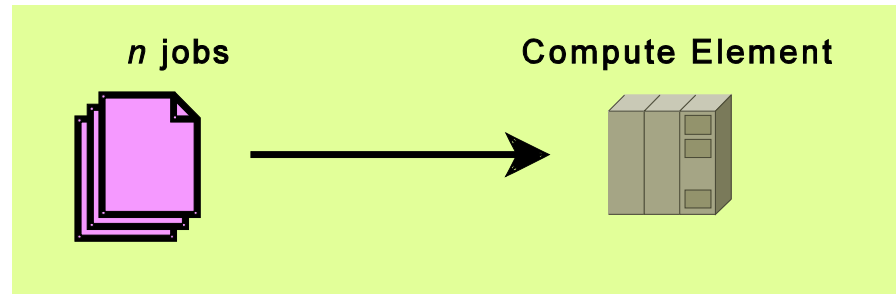
- **A real world example:**
 - 27,000 jobs
 - duration: approx. 4 hrs
 - approx. 280 WNs

- **Theoretical duration: 16 days**

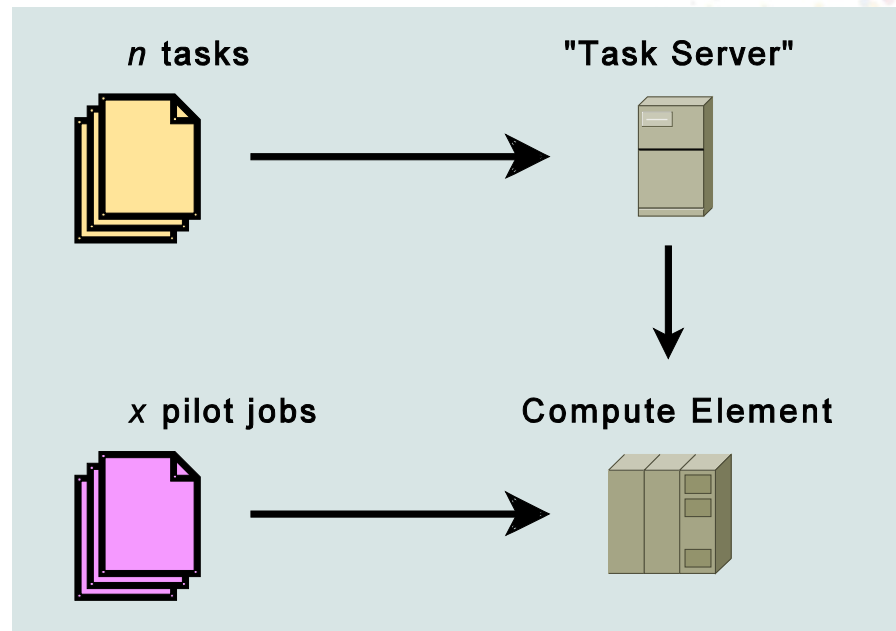
- **But with a success rate of 70% ...**
 - Approx. 9 resubmissions
 - “Practical” duration: >2 months

Pilot Jobs

▶ “Normal” jobs



▶ Pilot jobs



ToPoS: Trivialest of Possible Solutions

- A simple REST web service
 - accessible through HTTP
 - even just with a browser!

- “Use cases”:
 - User: create a directory to upload *task files*
 - User: upload task files, e.g.
 - ▶ data files (keep it small, please!)
 - ▶ parameter files (preferred!)
 - Job: ask the server for new files to process
 - Job: delete a file on the server



Jan's Challenge I

- Putting Jan's "parameter file" on the server, one file per line:

```
#!/usr/bin/perl -w
use strict;

my $counter = 0;
my $URL = 'https://topos.grid.sara.nl/3/realms/jan_bot/pools/todo/nextToken';
while (<>) {
    chomp;
    system("echo -n '$_' > param_$counter.txt");
    $counter++;
}
```

- Upload all these files:

```
#!/bin/bash
URL=https://topos.grid.sara.nl/3/realms/jan_bot/pools/todo/nextToken
for filename in param_*.txt; do
    curl -T $filename -H 'Content-Type: text/plain' $URL
done
```

Jan's Challenge II

The “nextToken” URL

■ What does ToPoS' nextToken resource return?

```
%> curl -i -L $NEXT_TOKEN
HTTP/1.1 303 See Other
Location: https://topos.grid.sara.nl/3/realms/jan_bot/pools/done3/tokens/19158
Content-Length: 730
Content-Type: application/xhtml+xml; charset=UTF-8
...

HTTP/1.1 200 OK
Content-Length: 23
Content-Type: text/plain; charset=US-ASCII
...

4.0 3.0 0.02 0 20000 20
```

Jan's Challenge III

■ In the job's shell script, put a loop around the processor:

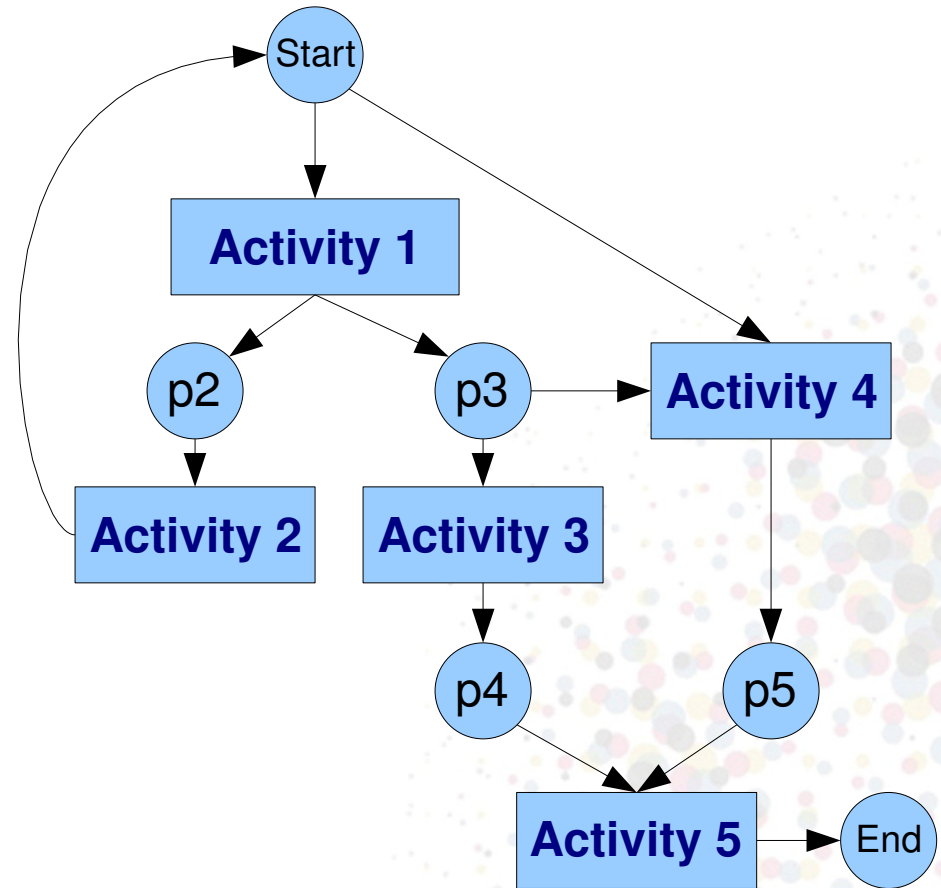
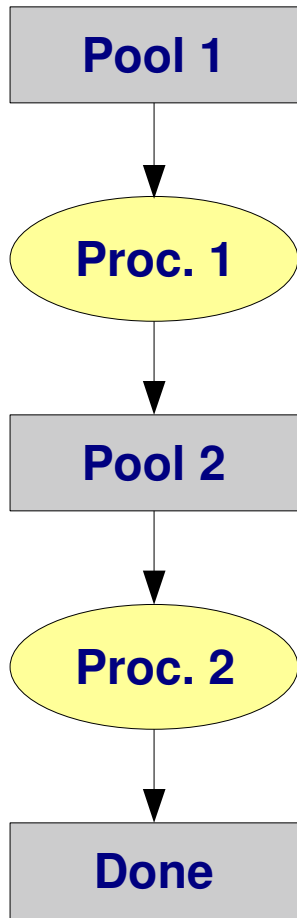
```
#!/bin/bash
TODO=https://topos.grid.sara.nl/3/realms/jan_bot/pools/todo/nextToken
DONE=https://topos.grid.sara.nl/3/realms/jan_bot/pools/done/nextToken

while true; do
  curl --dump-header headers.txt -L $TODO > params.txt
  TASKURL=$( grep '^Location: ' headers.txt | awk '{ print $2 }' )
  [ -n "$TASKURL" ] || break

  #
  # Do the processing...
  #

  #curl -X DELETE $TASKURL
  curl -T params.txt $DONE?delete=$( basename $TASKURL )
done
exit 0
```

ToPoS and Workflow



Portfolio

■ SciaGrid

- Collaboration between SRON, KNMI, NIKHEF and SARA
- Website where users can select
 - ▶ satellite data (Sciamachy)
 - ▶ data processors

■ Arnold Kuzniar and Jack Leunissen (WUR)

- BLAST protein sequence alignment

■ Bas Dutilh (CMBI)

- HAMMER sequence alignment (?)

■ Jan Bot (TUD)

- KCSmart

Future development

- ▶ **API's for Bash, Perl, Python, Java, Ruby, Lua, D, C#, Lisp, Scheme, Tcl, ...**
- ▶ **ATOM/RSS instead of / next to HTTP**
- ▶ **Uploading parameter files**
- ▶ **File integrity checksum**
- ▶ **??? Any wishes ???**