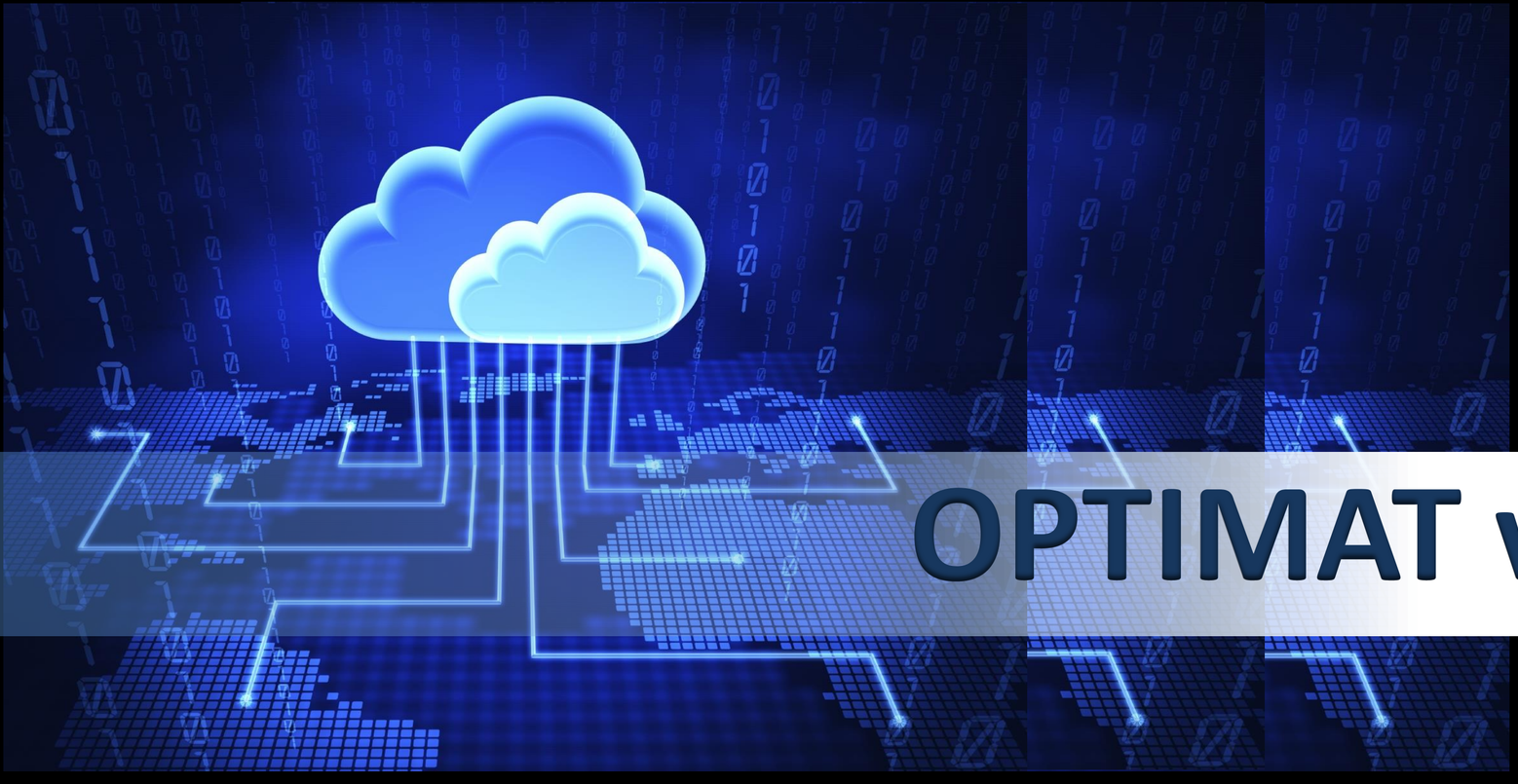

UNIVERSITY OF Southampton

Engineering and the Environment

Aeronautics, Astronautics and Computational Engineering

OPTIMAT v2: Job Manager

Dr. Ivan Voutchkov,

UTC for Computational Engineering, *AACE, Faculty of Engineering and the Environment*

Many engineering optimization and simulation analysis require large number of design trials that can be run in parallel. Various job management solutions exist, however most of them require active communication between master and slave nodes. The master is always in control of slave nodes, which is effective for dedicated clusters, but can be a significant inconvenience for utilisation of available desktop power. Often engineers have a powerful machines in front of them (12+ cores) which are sitting idle 95% of the time.

The current job manager allows each team member to dynamically contribute their CPU resources to one or more job queues. They can start, stop or adjust their contribution at will. If they decide to leave whilst executing a job, it will be taken automatically by another available contributor.

Jack

Kate

Paul

Jack, Kate (thermal) and Paul (structural) need resources to solve their parallel problems

submits jobs by creating JCFG files – Isight, batch script, etc.

collects results

JackJob1.jcfg, **JackJob2.jcfg**, **JackJob3.jcfg**, **KateJob1.jcfg**, **KateJob2.jcfg**

KateJob3.jcfg

```
executable = program1.exe arguments
stdout = stdout.txt
stderr = stderr.txt
runtime = 60 // minutes
repeat_on_fail = 3 // times to retry
numcpus = 4 // required cores
```

PaulJob1.jcfg, **PaulJob2.jcfg**

PaulJob3.jcfg

```
executable = program2.exe arguments
stdout = stdout.txt
stderr = stderr.txt
runtime = 120 // minutes
repeat_on_fail = 3 // times to retry
numcpus = 4 // required cores
```

\\sharedStorage\Thermal

\\sharedStorage\Struct

8 | 4 | 8 | 4 | 2

Peter
`jobRunner 8 \\sharedStorage\Thermal`

John
`jobRunner 4 \\sharedStorage\Thermal`

Mike
`jobRunner 8 \\sharedStorage\Thermal`
`jobRunner 4 \\sharedStorage\Struct`

David
`jobRunner 2 \\sharedStorage\Struct`

Team members with spare cores can volunteer to contribute resources to any QUEUE. They are in full control of how many cores to contribute and when exactly to start or end their contribution. Each member runs a 'jobRunner' command on their machine:

```
jobRunner <numcpus> <QUEUE>
```


OPTIMATv2 has been developed under the Strategic Investment in Low-Carbon Engine Technology (SILOET) project, RD6, WP2.6, Task 2.6.3.1

http://www.southampton.ac.uk/engineering/research/groups/CED/posters.page | email: iiv@soton.ac.uk
Computational Engineering & Design Group, University of Southampton, SO16 7QF, U.K.