Business from technology

# Energy-aware job scheduler for high-performance computing

**7.9.2011**

**Olli Mämmelä (VTT),** Mikko Majanen (VTT), Robert Basmadjian (University of Passau) , Hermann De Meer (University of Passau), André Giesler (Jülich Supercomputing Centre), Willi Homberg (Jülich Supercomputing Centre),
**olli.mammela@vtt.fi**

# Outline

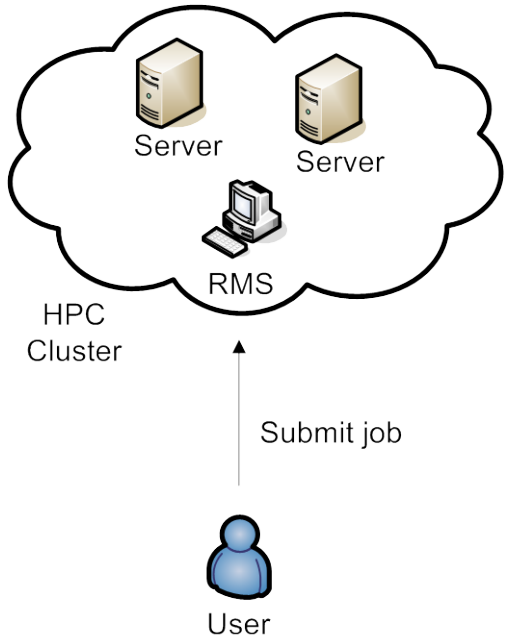
- Introduction
- HPC energy-aware scheduler
- Evaluation with simulation model
- Evaluation with real-world testbed
- Conclusions

# Introduction

- Energy-awareness has become a major topic nowadays
- ICT as a whole is estimated to cover 2% of world's carbon dioxide emissions
- HPC is no exception: growing demand for higher performance increases total power consumption
- Research in energy-aware HPC
  - Energy-efficient hardware
  - Dynamic Voltage and Frequency Scaling (DVFS) technique
  - Shutting down HW components at low system utilization
  - Power capping and thermal management
- This work presents an energy-aware job scheduler for HPC

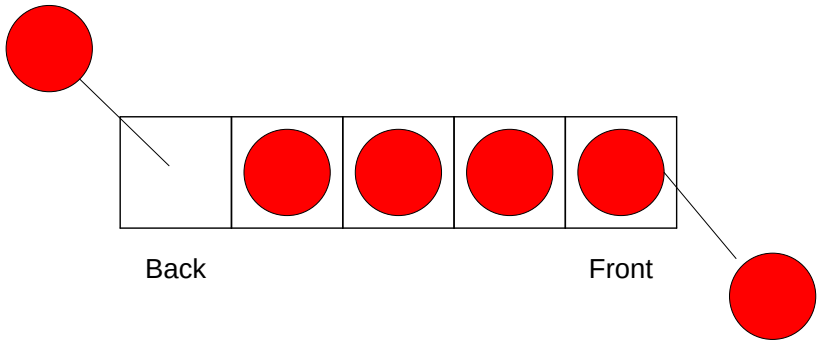
# HPC energy-aware scheduler

- HPC cluster consists of a resource management system (RMS) and several compute nodes
- Users submit jobs to the queue(s) inside the RMS
- Job scheduler is responsible for scheduling decisions
- Several algorithms available for job scheduling
- Energy-aware scheduler supports three commonly used scheduling algorithms with energy-saving features



# HPC energy-aware scheduler

- FIFO
  - When a job is completed, resources are checked for the first queue item
  - If not enough resources, all jobs have to wait
- Energy-aware FIFO (E-FIFO)
  - Go through the queue until the first job cannot be started
  - Check estimated start time of the 1st job in the queue based on the available resources and currently running jobs
  - If estimated start time is more than *T* seconds all idle nodes are powered off

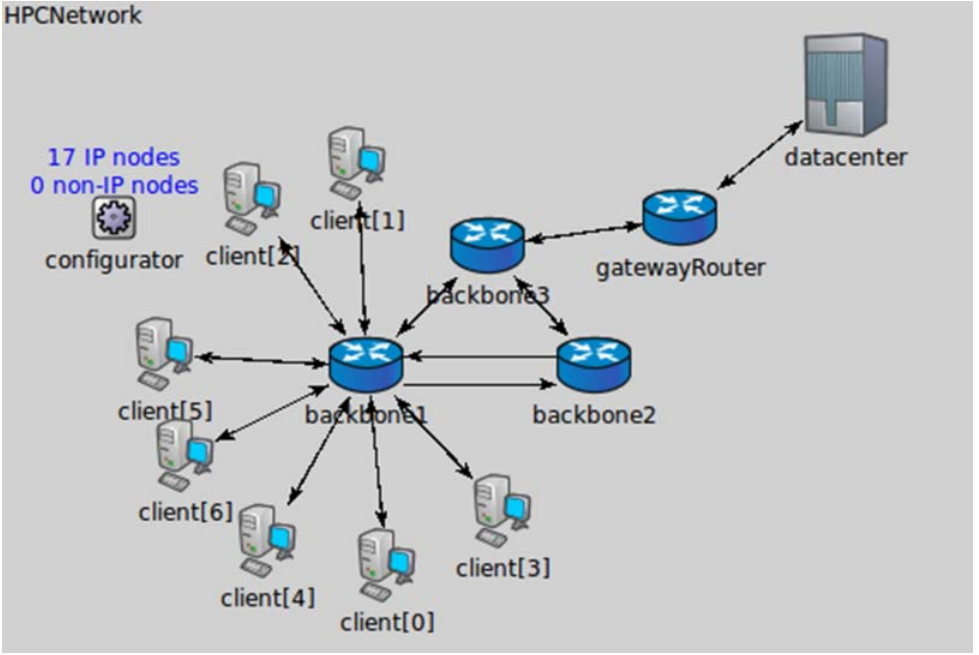


# HPC energy-aware scheduler

- Backfilling (first fit and best fit)
  - Functions like FIFO, but when there are not enough resources for the execution of the first job in the queue, the rest of the queue is checked for jobs that can be executed
  - Execution should not cause any delay for the first job
  - Backfill First Fit (BFF): first job that meets the resource and time constraints is chosen
  - Backfill Best Fit (BBF): all potential backfill jobs are searched and the selection is made based on certain criteria
  - In this work BBF uses these criteria to select the ”best” job
    1. Nodes
    2. Cores
    3. Memory
- Energy-aware backfilling (E-BFF and E-BBF)
  - Same methods for energy savings as in FIFO
  - Idle nodes are powered off if the estimated start time of the first job in the queue is more than *T* seconds
  - Backfilling has less opportunities to turn off idle nodes than FIFO

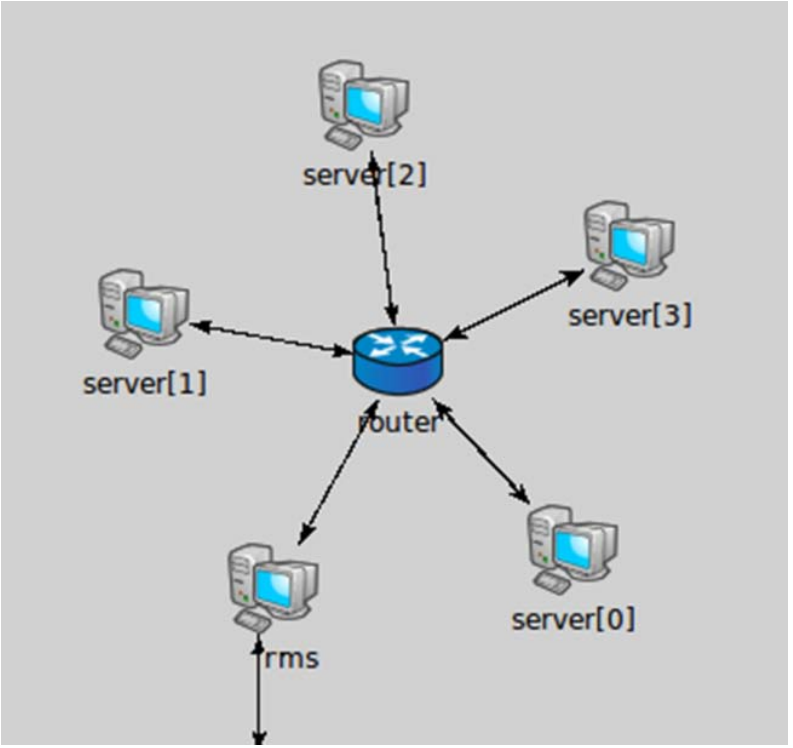
# Simulation model

- HPC simulation model implemented with OMNeT++ and the INET Framework
- Models for clients, data centre, servers, and the RMS
- Network topology consists of three backbone routers and a gateway router
- Clients send job requests to the data centre



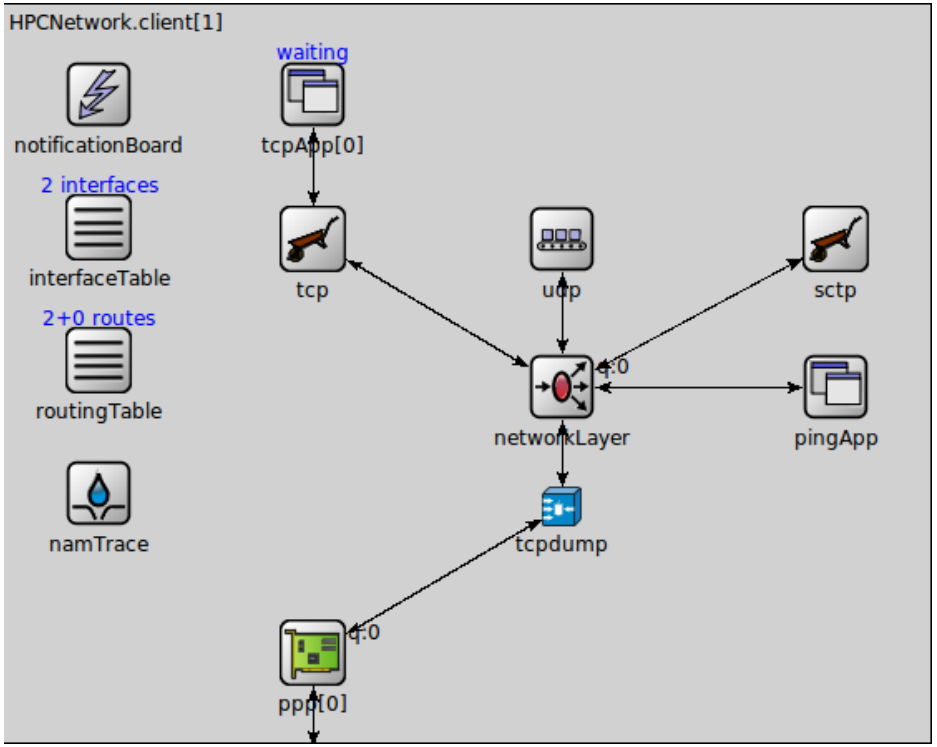
# Simulation model

- Data centre module consists of servers, the RMS and a router between them
- RMS handles incoming job requests and schedules the jobs to the servers
- RMS also sends power off / power on actions when needed
- Servers receive jobs from the RMS and execute the jobs



# Simulation model

- RMS, servers, and clients derived from StandardHost module of INET Framework
- Transport, network, physical layer protocols already available
- Functionalities developed as an application layer program



# Simulation model

- Application models also include models of the server components and their power consumption models
- Details of server CPUs, cores, memory, fans, etc. are defined
- Power consumption models
  - Processor, memory, hard disk, network interface card, mainboard, fan, power supply unit
  - Models were derived by performing various observations with physical equipment and specific benchmark programs

# Simulation parameters

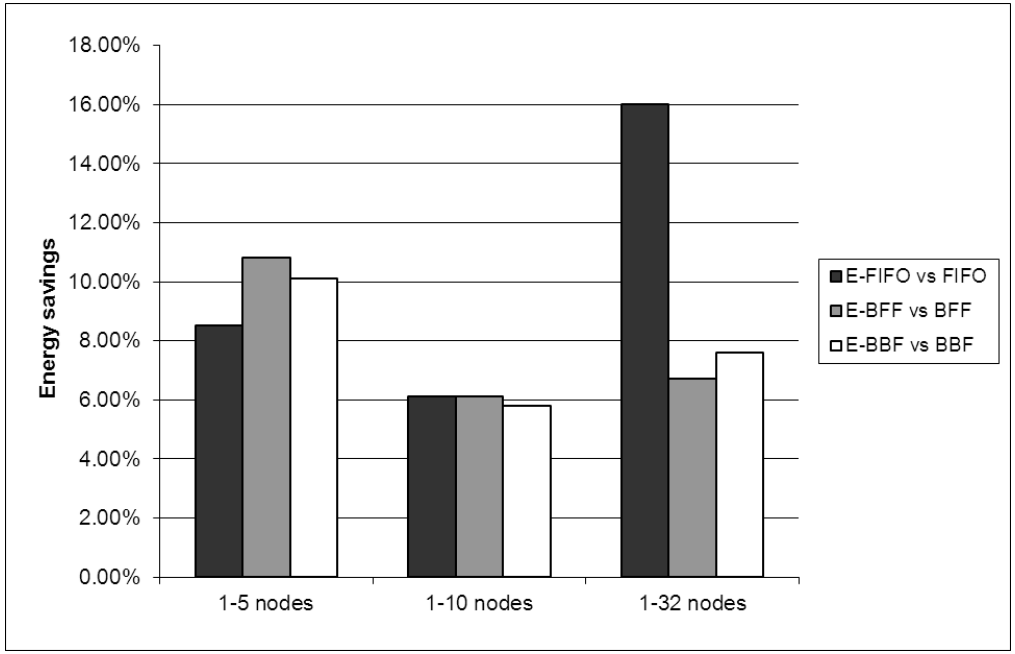
| Parameter | Value |
|---|---|
| Number of clients | 20 |
| Number of servers | 32 |
| Number of job requests | 20 * 20 = 400 |
| Job cores | 1, 2 or 4 |
| Job core load | uniform(30, 99) |
| Job memory | uniform(100 MB, 2 GB) |
| Job wall time | uniform(600 s, 86400 s) |
| Job nodes | uniform(1, 5), uniform(1, 10) and uniform(1, 32) |
| Number of simulation runs | 10 |

# Server parameters

| Parameter | Value |
|---|---|
| Number of CPUs | 2 |
| Cores per CPU | 2 |
| Core frequency | 2.4 GHz |
| RAM size | 4 * 2 GB = 8GB |
| RAM vendor | Kingston |
| RAM type | DDR2 800 MHz, unbuffered |

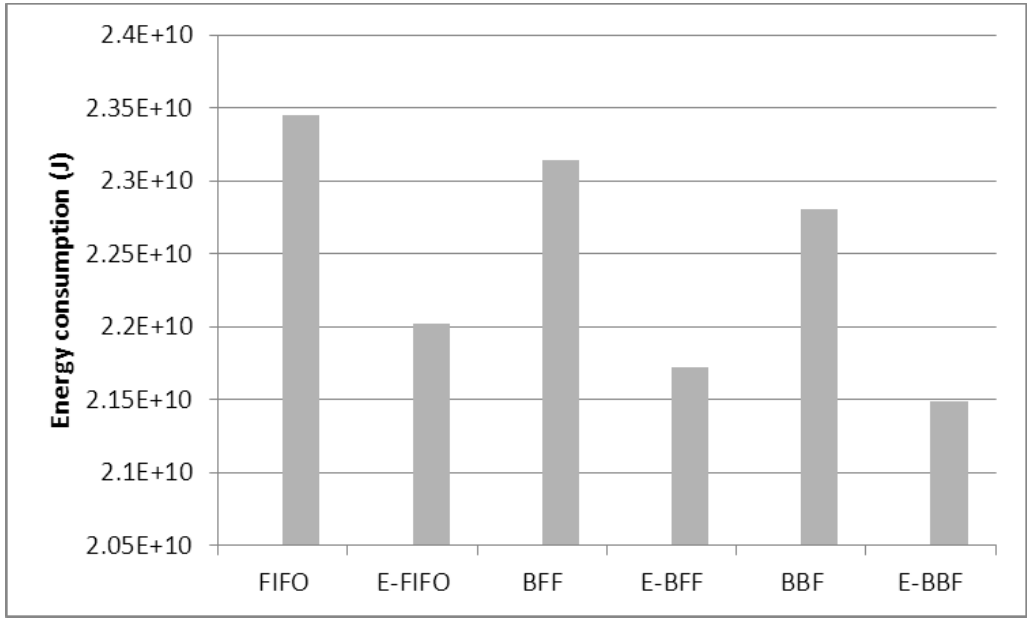
# Energy savings

- Comparing standard scheduling algorithms to their energy-aware versions
- Highest energy saving of 16 % with E-FIFO (1-32 nodes)
- Other savings approx. 6-10 %
- Savings are highly dependent on system utilization



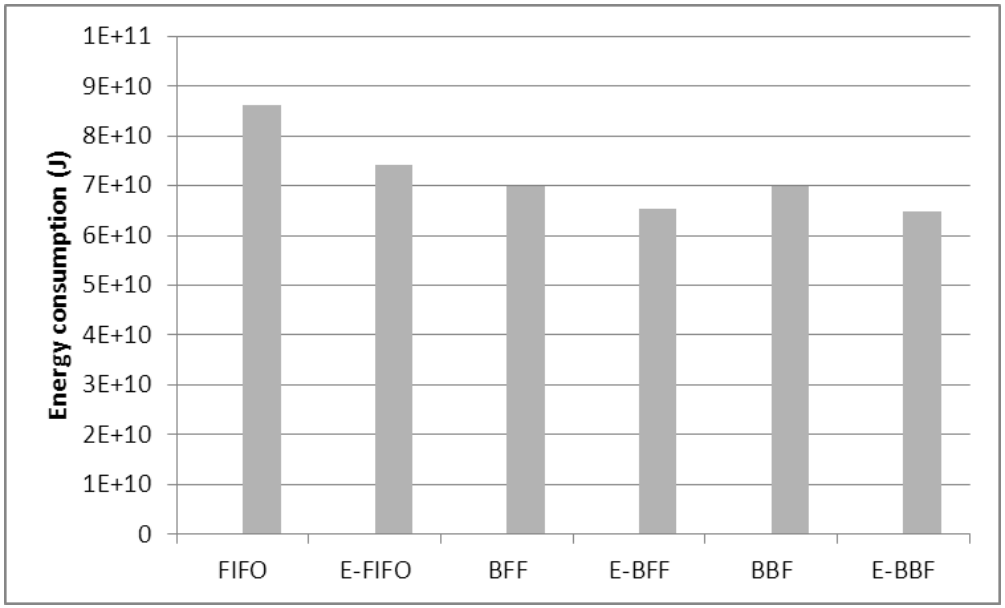
# Energy consumption (J), 1- 10 nodes

- FIFO is the most energy consuming
- Backfilling itself can decrease energy consumption
  - 1.3 % BFF vs FIFO
  - 2.8 % BBF vs FIFO
- Energy-aware backfill best fit (E-BBF) consumes least amount of energy
- E-BBF saves 9.1 % energy compared to FIFO

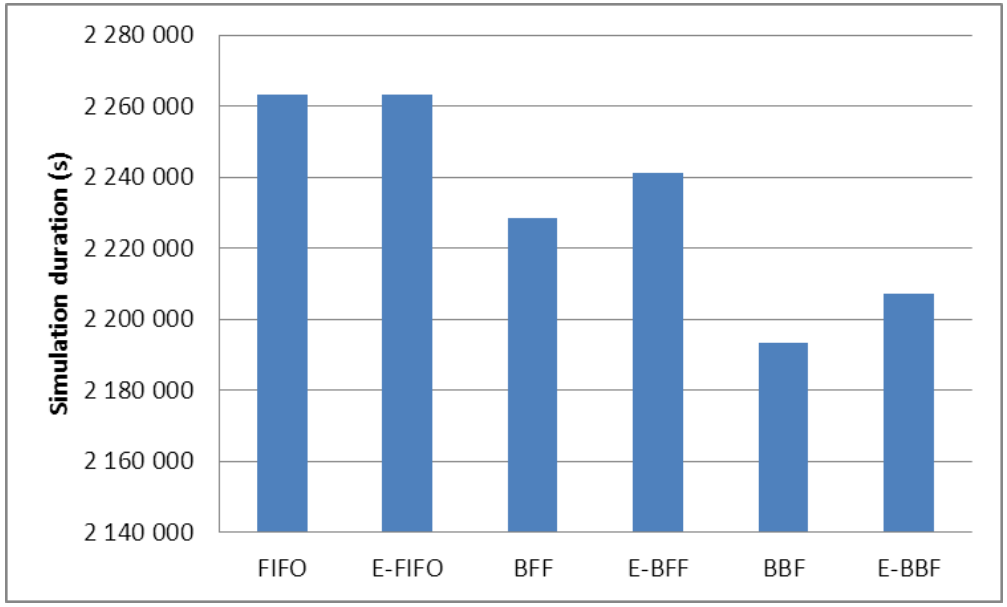


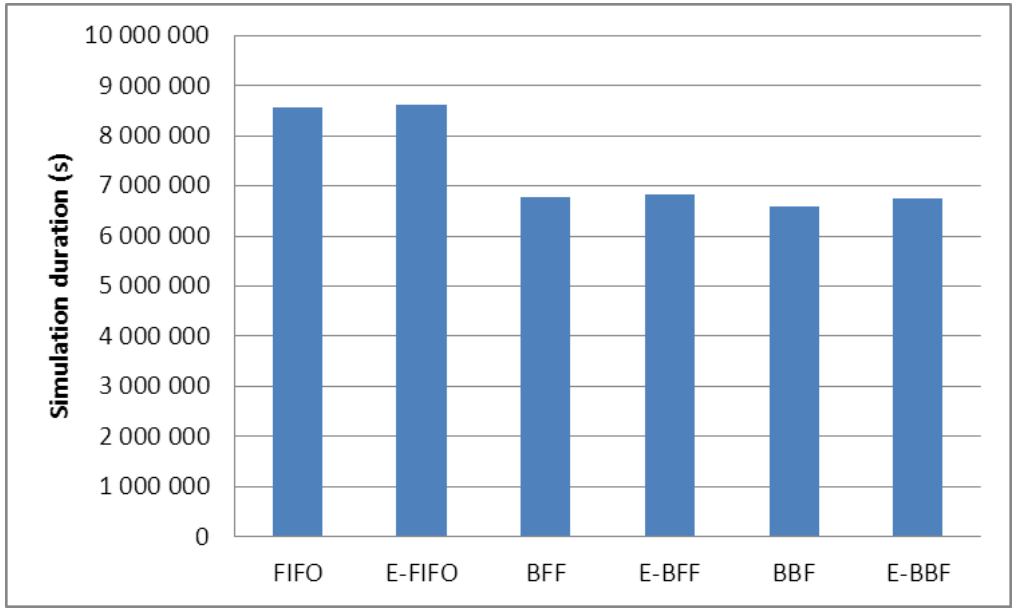
# Energy consumption (J), 1-32 nodes

- FIFO consumes again most energy
- Compared to FIFO, E-BBF can reduce energy consumption by 33 %
- Savings by standard backfilling is approximately 23 % compared to FIFO



# Average simulation duration (s)

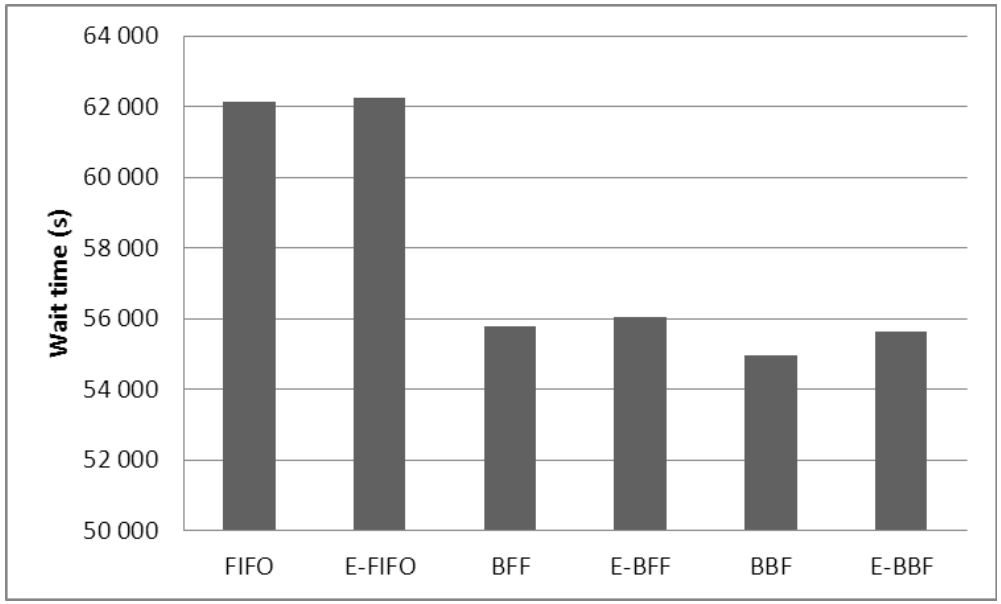


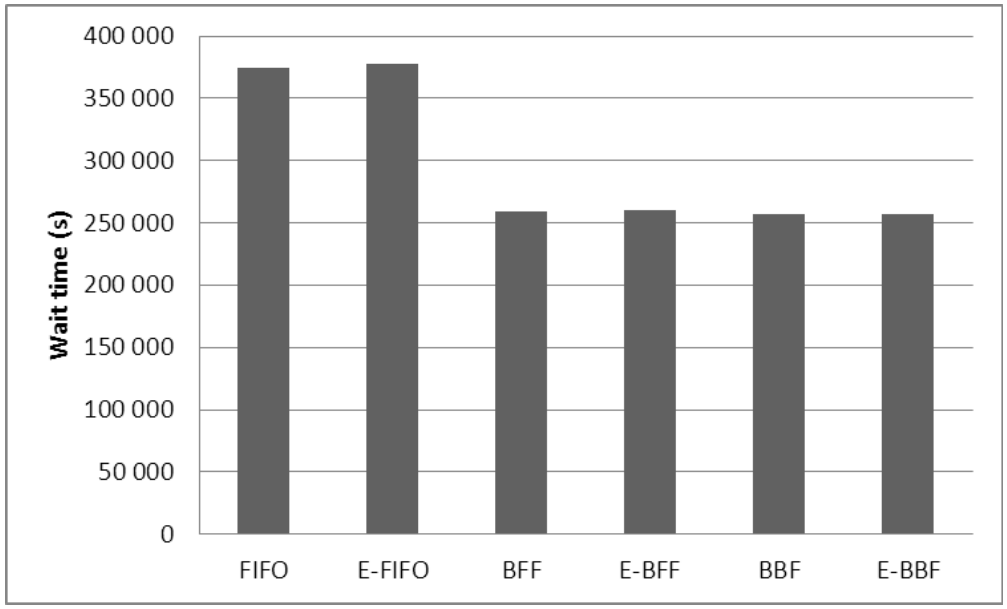




- 1-10 nodes requirements
- At highest 0.62 % increase (BBF vs E-BBF)

- 1-32 nodes requirements
- 2.32 % increase at highest (BFF vs E-BFF)

# Average wait time (s)




- 1-10 nodes
- 1.2 % increase at highest (BBF vs E-BBF)

- 1-32 nodes
- 0.81 % increase at highest (BBF vs E-BBF)

# Testbed configuration

- Energy-aware scheduler was also implemented in Juggle cluster at Jülich Supercomputing Centre
- Testing environment simulated typical usage of a supercomputer by using a workload generator
- Several benchmarks programs were used in the tests, more details in the paper
- Default scheduler of the testbed was Torque RMS
- Power was measured by Raritan device in intervals of three seconds
- Strategy for power savings was to place idle nodes in low-power standby state if no jobs which could make use of them
- Standby mode consumes 50 W less power than idle state/mode

# Juggle testbed parameters

| Parameter | Value |
|---|---|
| Number of nodes | 4 |
| CPUs per node | 2 |
| Cores per CPU | 2 |
| Core frequency | 2.4 GHz |
| CPU architecture | AMD Opteron F2216 |
| Operating System | Linux |
| CPU Idle Power | 95 W |
| RAM size | 4 * 8 * 1 GB = 32 GB |
| RAM vendor | Kingston |
| RAM type | DDR2 667 MHz, unbuffered |

# Testbed results

| | Torque Scheduler | E-BFF |
|---|---|---|
| Elapsed time | 2049 s | 2062 s |
| Energy consumed | 1600 kJ | 1500 kJ |
| Avg. power consumed | 781 W | 729 W |

- An energy saving of 6.3% was achieved
- Elapsed time increases 0.63%

# Conclusions

- Developed energy-aware scheduler can be applied to HPC data centres without any changes any hardware
- With the simulation energy savings of 6-16 % were achieved with energy-aware scheduling strategies compared to standard scheduling algorithms
- Choice of a job scheduling algorithm can have an effect on the energy consumption
- Testbed experiments also showed energy savings without a large increase in completion time
- Simulation and testbed experiments showed similar results, which means that the simulation is able to model real-world environment accurately

# Future work

- Apply DVFS technique when appropriate
- Explore different variaties of the backfill best fit algorithm with regards to energy
- Try out different low power states, such as standby or hybernated
- Expand the work to include multiple data centres in a federated site scenario

# More information


- olli.mammela@vtt.fi
- This work was supported by the EU FP7 project FIT4Green
- www.fit4green.eu

# VTT creates business from technology