Utilization Driven Power-Aware Parallel Job Scheduling

Maja Etinski Julita Corbalan Jesus Labarta Mateo Valero

{maja.etinski,julita.corbalan,jesus.labarta,mateo.valero}@bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Motivation

- Performance increase has been followed by even higher increase in power consumption

Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.60
2	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU / 2010 Dawning	120640	1271.00	2984.30	
3	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband / 2009 IBM	122400	1042.00	1375.78	2345.50
4	National Institute for Computational Sciences/University of Tennessee United States	Kraken XT5 - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009 Cray Inc.	98928	831.70	1028.85	
5	Forschungszentrum Juelich (FZJ) Germany	JUGENE - Blue Gene/P Solution / 2009 IBM	294912	825.50	1002.70	2268.00

Top500

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	773.38	Forschungszentrum Juelich (FZJ)	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
1	773.38	Universitaet Regensburg	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
1	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
4	492.64	National Supercomputing Centre in Shenzhen (NSCS)	Dawning Nebulae, TC3600 blade CB60-G2 cluster, Intel Xeon 5650/ nVidia C2050, Infiniband	2580
5	458.33	DOE/NNSA/LANL	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Infiniband	276

Green500

Power reduction approaches in HPC

Power reduction approaches

Application level:

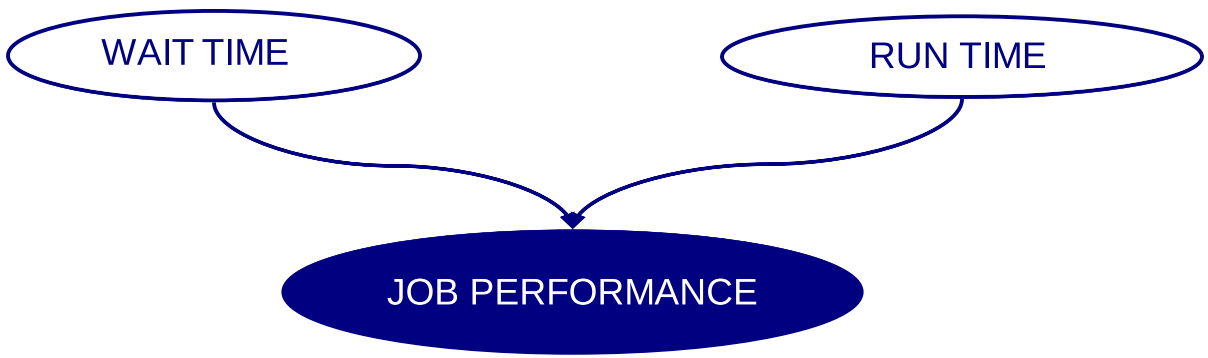
- Runtime systems:
  - exploit certain application characteristics (load imbalance, communication intensive regions)
  - based on very fine grain DVFS application

System level:

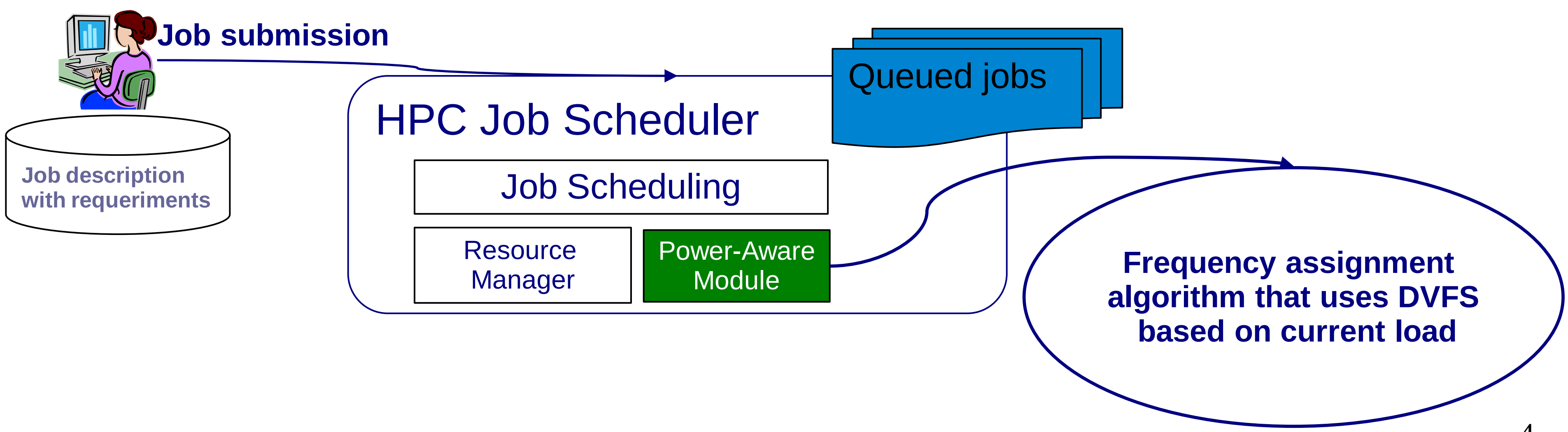
- Turning off idle nodes:
  - resource allocation such that there are more completely idle nodes
  - determining number of online nodes
- Operating system power management via DVFS:
  - linux governors – per core, unawareness of the rest of the system
- DVFS taking into the account entire system workload?

Power-Aware Parallel Job Scheduling

- Job performance in HPC center depends on two components:



- Job scheduler has a global view of the whole system:

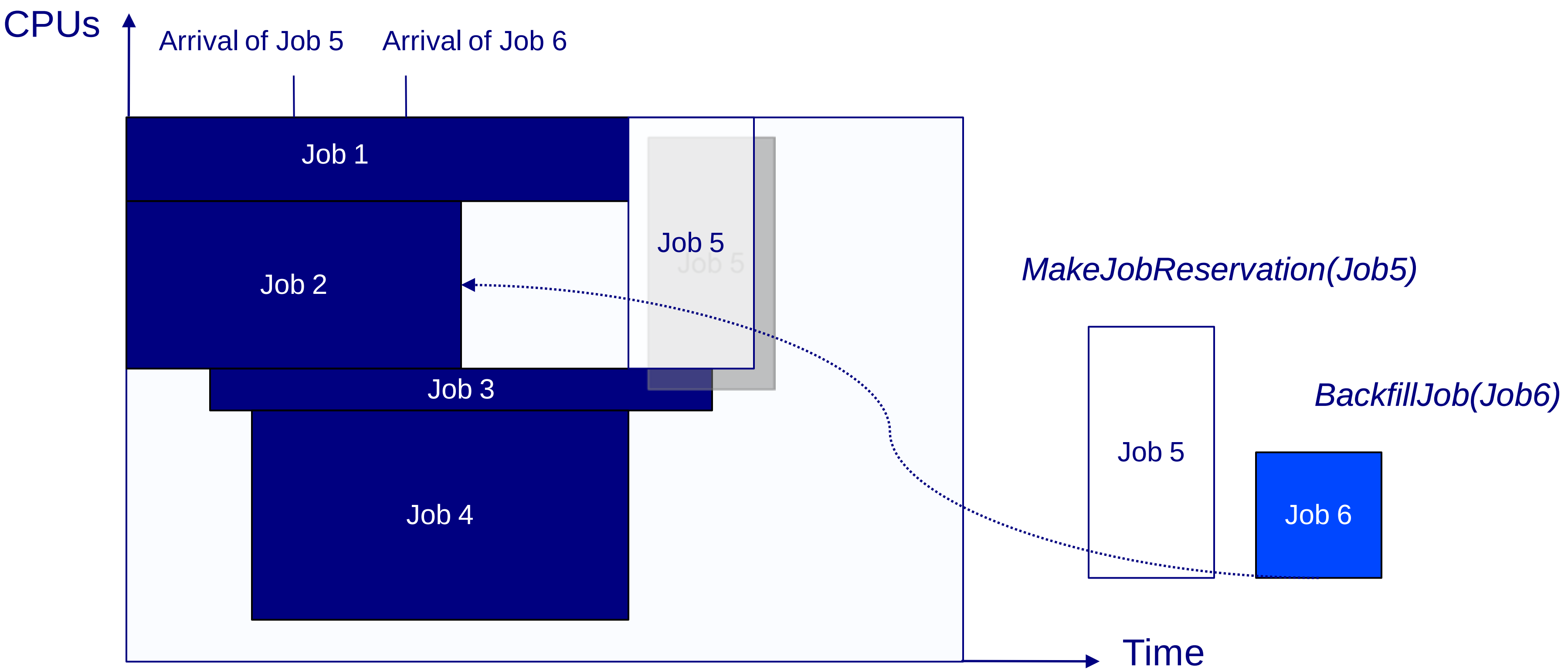


Outline

- Parallel job scheduling:
  - the EASY backfilling policy
  - **frequency assignment**
- Power and run time modeling:
  - how does frequency scaling affect power dissipation and execution time?
- Evaluation:
  - experimental methodology (simulator, workloads, policy parameters)
  - results
  - evaluation of system size increase

The EASY backfilling policy

- Jobs are executed in FCFS order except when the first job in the wait queue can not start
- Users have to submit an estimation of job's runtime – *requested time*
- When the first job in the WQ can not start, a reservation is made for it based on requested times of running jobs
- A job is executed before previously arrived ones only if it does not delay the first job in the queue



When to use DVFS? Which frequency?

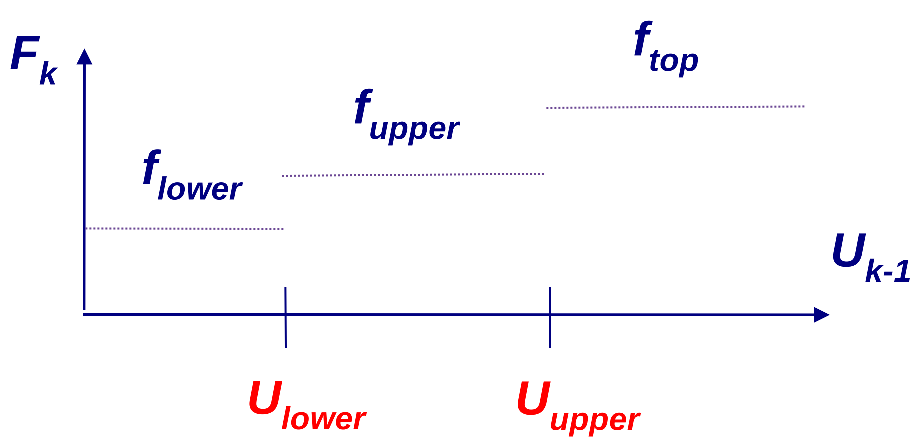
- Use of DVFS during period of low system activity
- When utilization is low, impact on performance is minimal (normally there are no queued jobs)
- Majority of workloads have average systems utilizations in range 45% - 75% (Parallel Workload Archive)
- Transient periods of low load (over night and holidays)
- Two levels of control:
  - system utilization
  - number of jobs in the wait queue
- Frequency assignment algorithm can be applied with any parallel job scheduling policy (Industrial strength schedulers are usually based on backfilling policies)

Frequency assignment

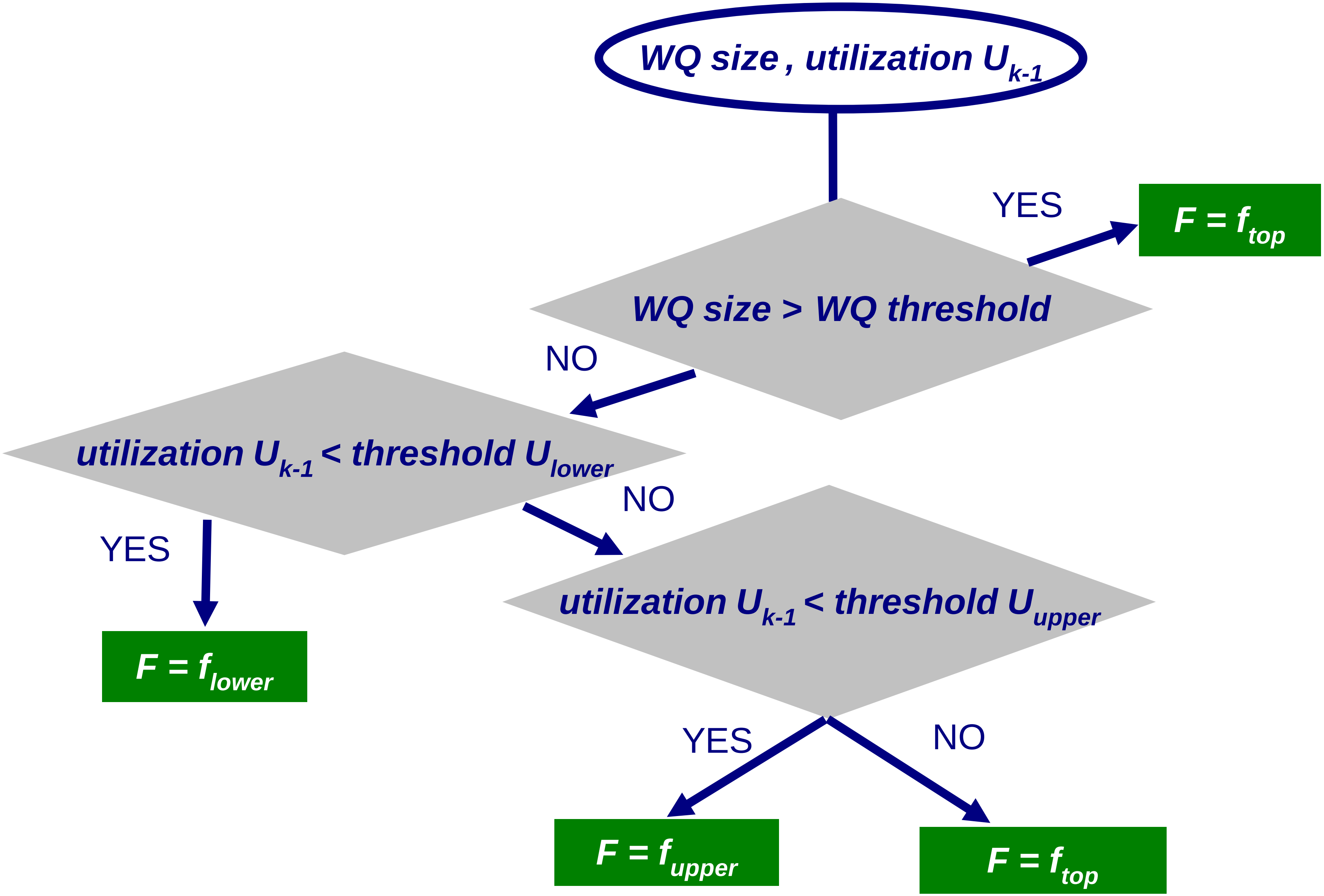
- Frequency assigned once (at jobs start time) for entire job execution
- Utilization is computed for each interval ***T***:

$$U_j = \frac{\sum_{k=1}^{N_{jobs}} Proc_k * RunTime_k}{N_{proc} * T}$$

- If there are more than $WQ_{threshold}$ jobs in the wait queue no frequency scaling will be applied
- Otherwise, job started during interval J_k runs at frequency ***F***



UPAS



WQ size , utilization U_{k-1}
WQ size > WQ threshold
YES
$F = f_{top}$
NO
utilization U_{k-1} < threshold U_{lower}
YES
$F = f_{lower}$
NO
utilization U_{k-1} < threshold U_{upper}
YES
NO
$F = f_{upper}$
$F = f_{top}$

Power Model

- *CPU power presents major portion of total system power*
- *It consists of dynamic and static power:*
  - $\mathbf{P_{cpu}} = P_{dynamic} + P_{static}$ $\mathbf{P_{dynamic}} = AcfV^2$ $\mathbf{P_{static}} = \alpha V$
- Fraction of static in total CPU power is a model parameter:
  - $P_{static}(V_{top}) = \mathbf{X}(P_{static}(V_{top}) + P_{dynamic}(f_{top}, V_{top}))$

  ($\mathbf{X}$ = 25% in our experiments)
- Two scenarios for idle CPUs:
  - idle processors do not consume power
  - idle CPUs are at the lowest frequency with low activity factor
- Average activity factor assumed to be same for all jobs
- Activity factor of idle processors 2.5 times lower than running activity
- *DVFS gear set :*

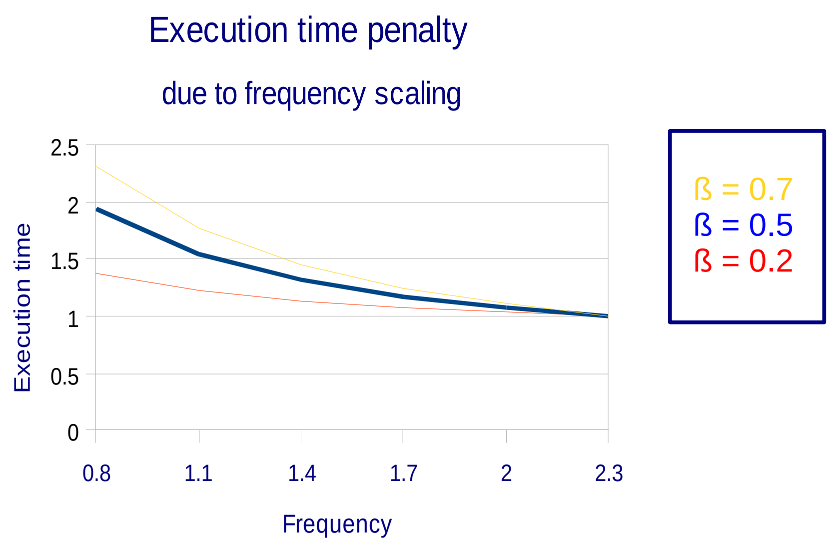
f	0.80	1.10	1.40	1.70	2.00	2.30
V	1.00	1.10	1.20	1.30	1.40	1.50
Norm(P)	0.28	0.38	0.49	0.63	0.80	1.00

Time Model

- Execution time dependence on frequency is captured by the following model:

$$F(f,\beta)=T(f) / T(f_{top}) = \beta(f_{top} / f - 1) + 1$$

[Hsu,Feng SC05: *A Power-Aware Run Time System for High-Performance Computing*]



- *ß* is assumed to have the following distributions:

Number of CPUs	Distribution
less or equal to 4	$\mathcal{N}(0.5,0.01)$
between 4 and 32	$\mathcal{N}(0.4,0.01)$
more than 32	$\mathcal{N}(0.3,0.0064)$

Evaluation

- C++ event driven parallel job scheduling simulator has been upgraded

Alvio simulator

- Policy parameters:

utilization thresholds: $\boldsymbol{U_{lower}}$ = 50% $\boldsymbol{U_{upper}}$ = 80%

reduced frequencies: $\boldsymbol{f_{lower}}$ = 1.4 GHz $\boldsymbol{f_{upper}}$ = 2.0 GHz

utilization computation interval: $\boldsymbol{T}$ = 10 min

wait queue length threshold: $\boldsymbol{WQ_{threshold}}$ = 0, 4, 16, NO

Policy parameters

- Metric of job performance – Bounded Slowdown

$$BSLD = max\left(\frac{WaitTime + RunTime}{max(RTthreshold, RunTime)}, 1\right)$$

- BSLD at frequency *f*

$$BSLD = max\left(\frac{WaitTime + NewRunTime(J, f)}{max(RTthreshold, RunTime)}, 1\right)$$

Metric of performance

Workloads

- Five workloads from production use have been simulated:

Cornell Theory Center
-large jobs with relatively low level of parallelism

San Diego Supercomputing Center
-less sequential jobs than CTC
-runtime distribution similar

San Diego Supercomputing Center
- no sequential job

Lawrence Livermore National Lab
- small to medium size jobs

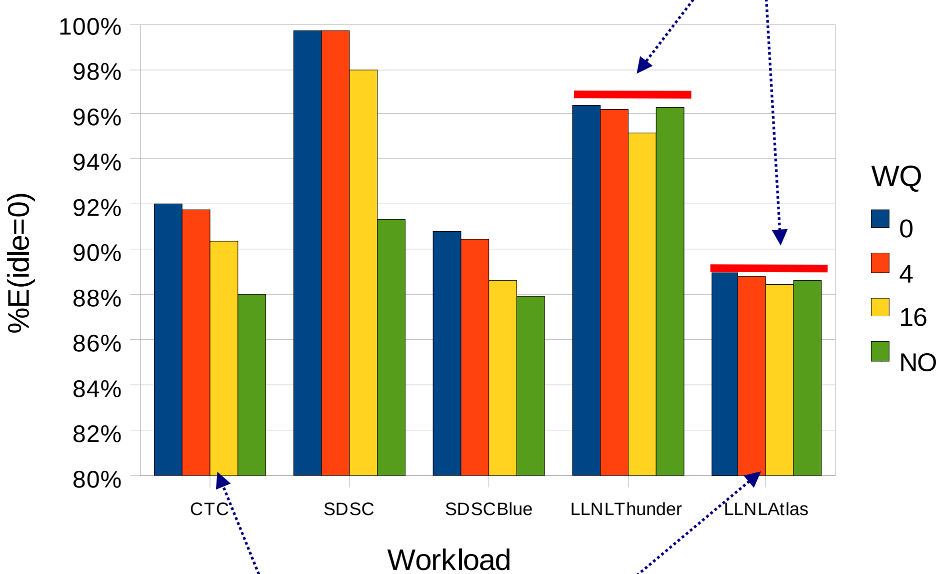
Lawrence Livermore National Lab
- large parallel jobs

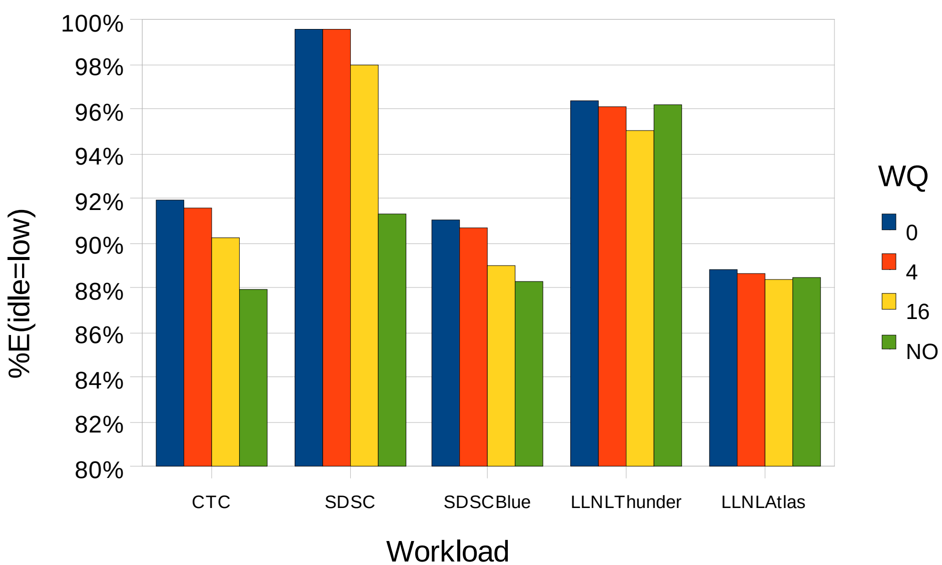
Workload - #CPUs	Avg Util	Avg LR	%T below U_{upper}	% T below U_{lower}
CTC - 430	70%	1.61	50%	28%
SDSC – 128	85%	8.17	26%	5%
SDSCBlue – 1152	69%	2.31	55%	26%
LLNLThunder – 4008	80%	0.80	29%	11%
LLNLAtlas – 9216	75%	0.94	26%	19%

Parallel workload archive
http://www.cs.huji.ac.il/labs/parallel/workload

Results: Energy - Original System Size

short wait queues





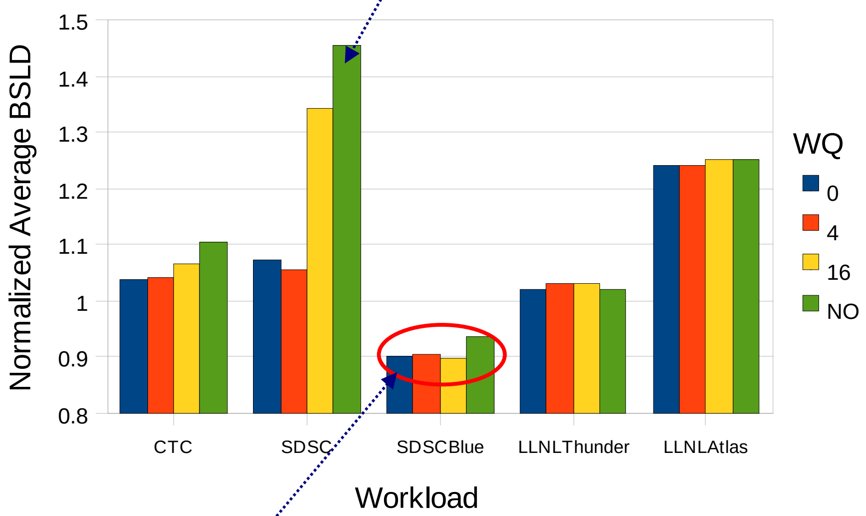
very similar results for both energy scenarios

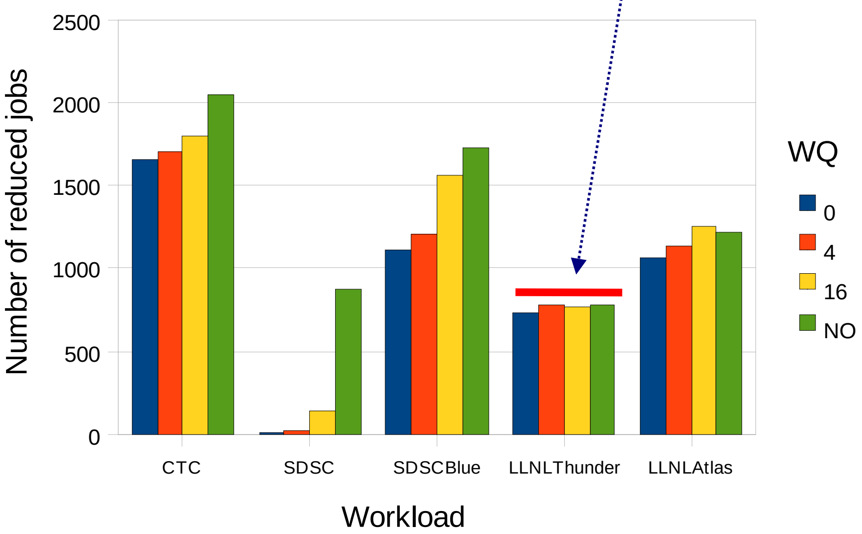
savings of not highly loaded workloads up to 12%

Results: Performance – Original System Size

high penalty in the least conservative case for highly loaded workload

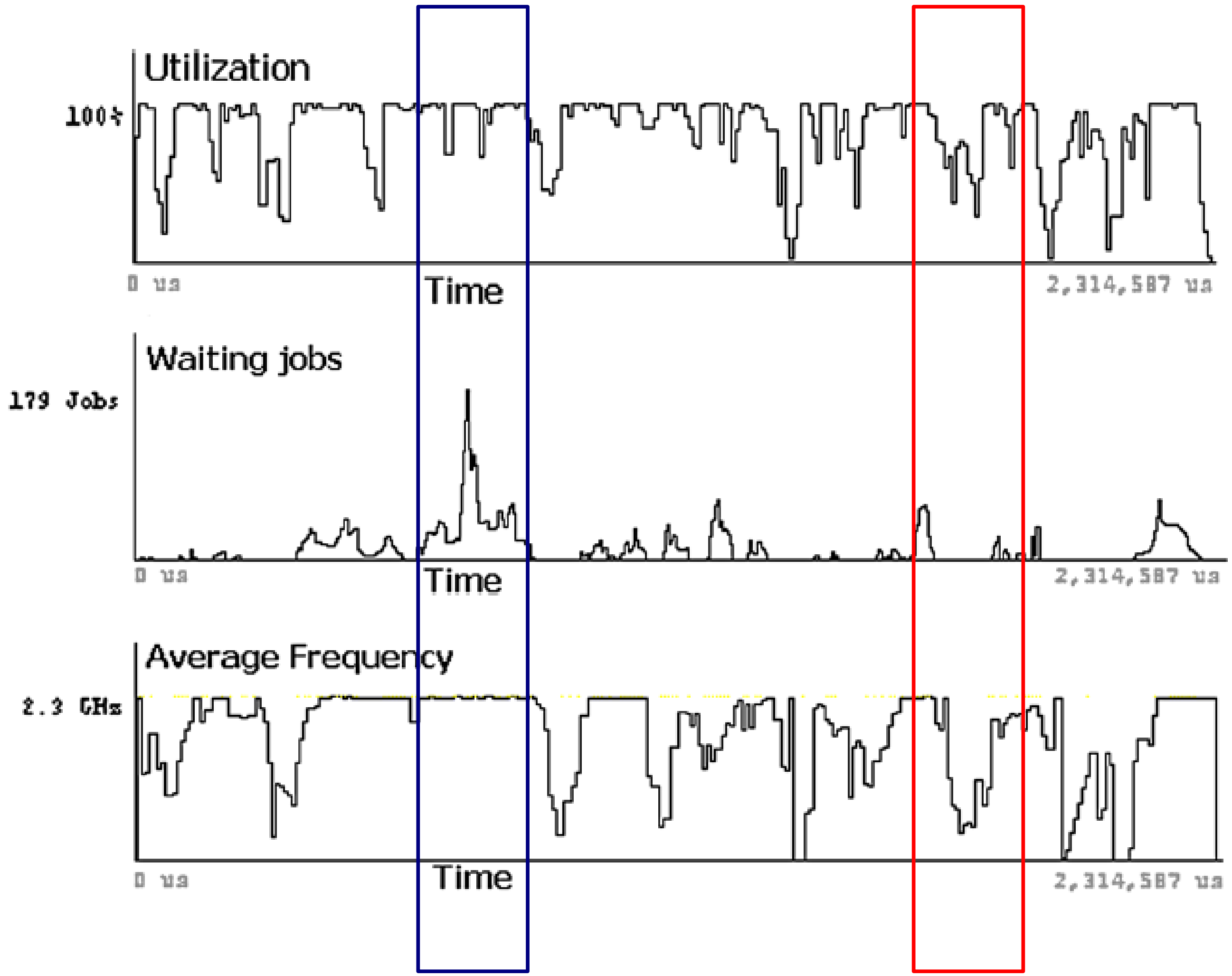
WQ threshold has almost no impact





an increase in number of backfilled jobs

Average frequency - SDSCBlue



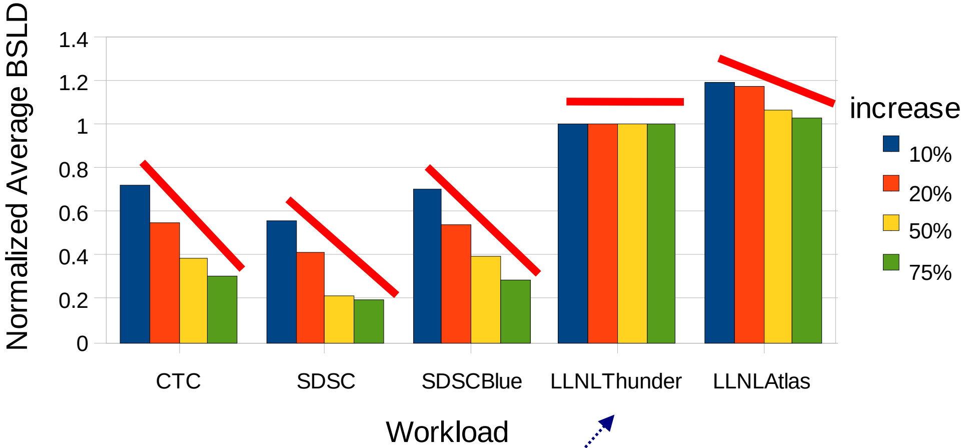
DVFS system size

- Frequency scaling is applied when load/utilization is low
- More CPUs -> lower load/utilization -> more opportunities for DFVS application
- DVFS scaling leads to lower power
- More CPUs -> lower CPU energy
- More CPUs -> better job performance due to lower wait times
- ***Is it possible to achieve both? (lower energy and higher performance)***
- Following system sizes have been considered in the evaluation process:
  - 10%, 20%, 50% and 75% bigger systems

System Oversizing: Performance

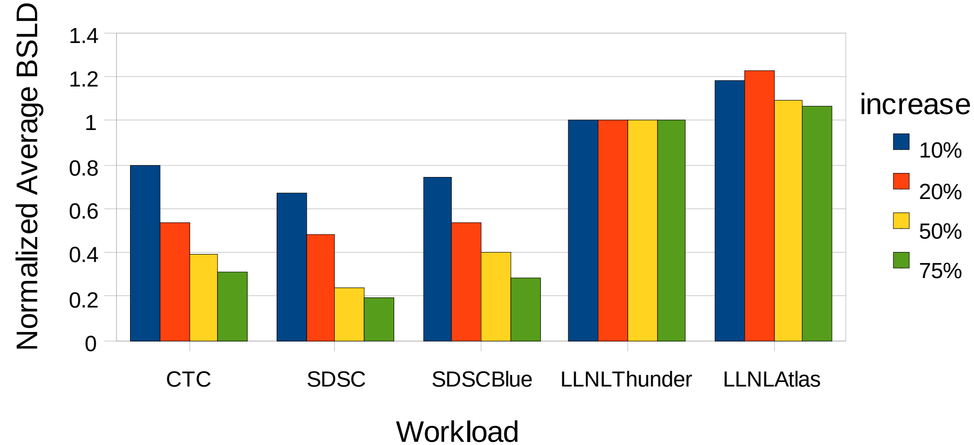
shorter wait time -> higher performance

WQ$_{threshold}$ = 0



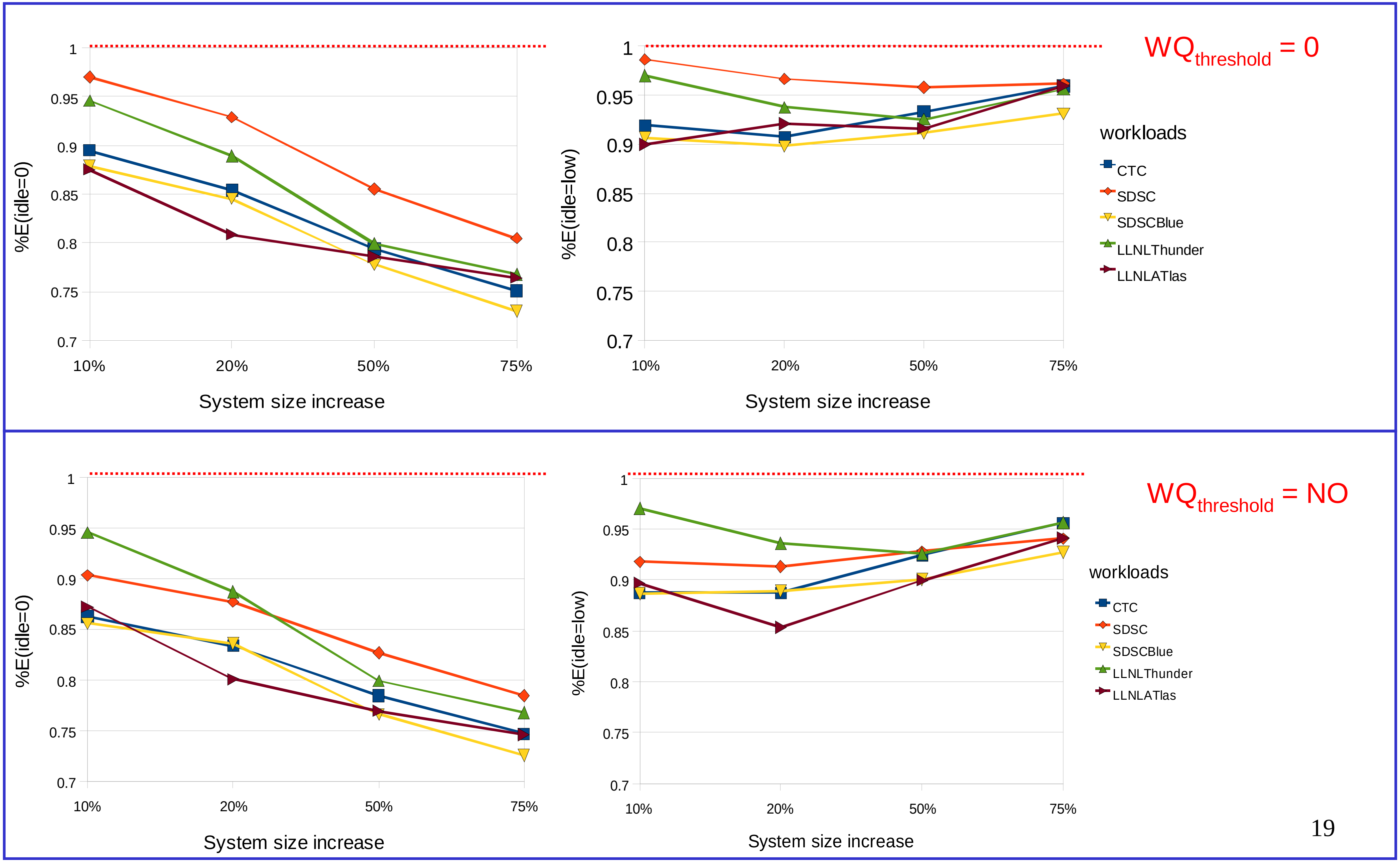
LLNLThunder has perfect BSLD

WQ$_{threshold}$ = NO



CTC, SDSC, SDSCBlue achieve performance better than original for only 10% increase in system size

System Oversizing: Energy



Conclusions

- Use of DVFS at the level of parallel job scheduling has been proposed
- A power-aware parallel job scheduling policy based on system utilization has been evaluated
- Trade-off between job performance and energy
- For less loaded workloads it is possible to save up to 12% of energy without affecting average BSLD significantly
- Modest energy savings in highly loaded workloads result in high performance penalty
- An analysis of system dimension has been performed showing that bigger DVFS systems can results in lower CPU energy consumption and higher job performance

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Utilization Driven Power-Aware Parallel Job Scheduling

Thank you for your attention!