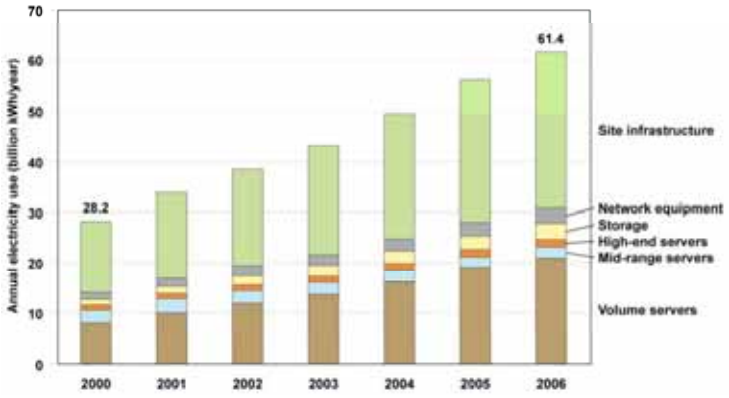
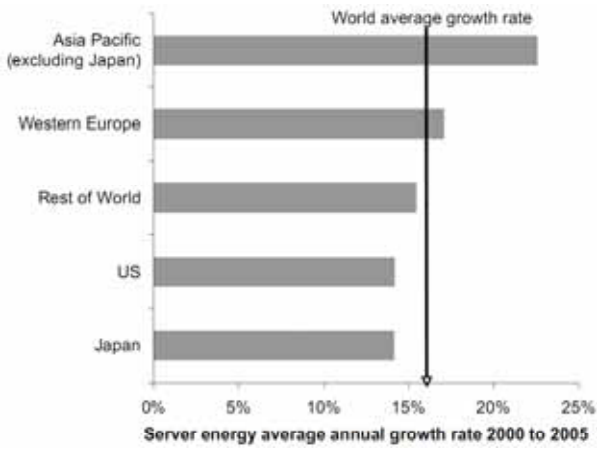
Energy-Efficient HPC: Complementing Green hardware with Green Software

The problem: Data Centre Energy Usage

US Data Centre Energy Use

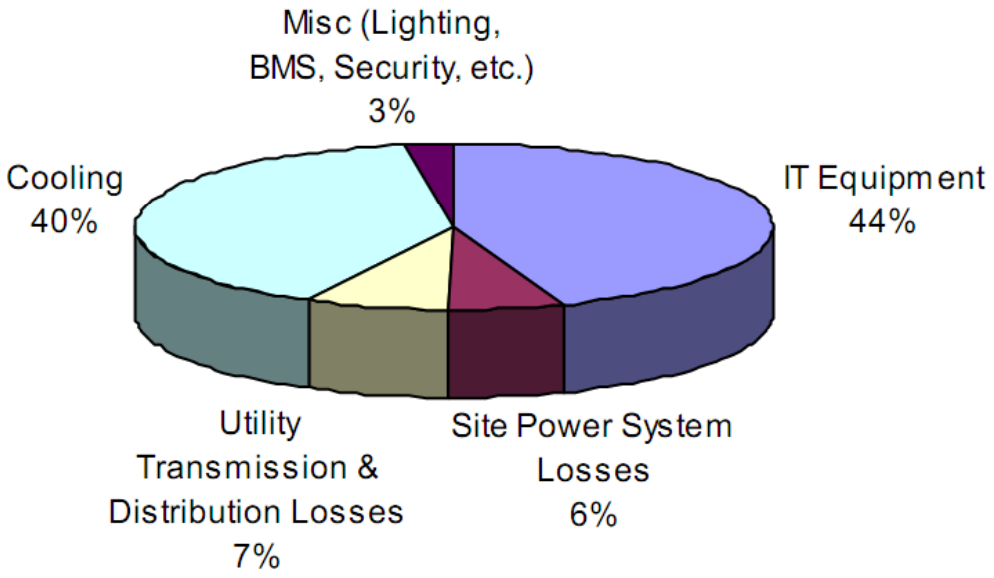


Worldwide Data Centre Energy Use Growth Rates



U.S. Environmental Protection Agency ,ENERGY STAR Program
Report to Congress on Server and Data Center Energy Efficiency, Public Law 109-431

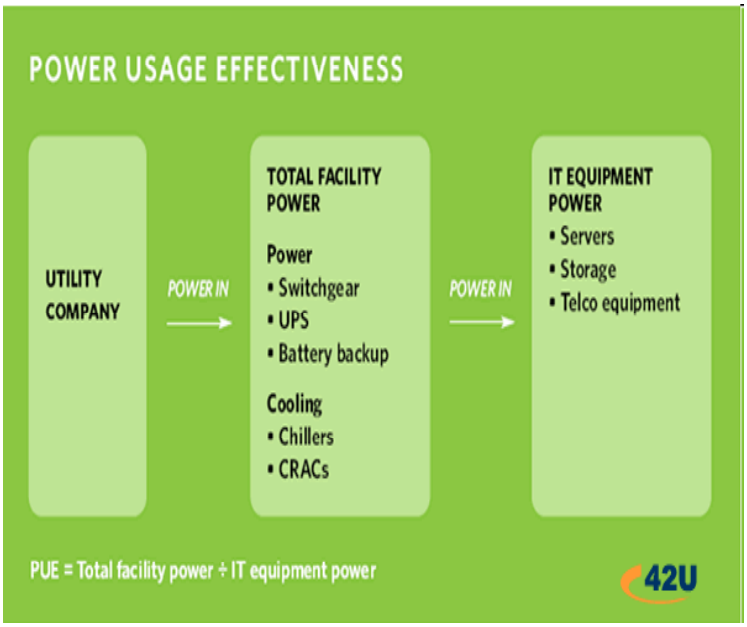
Breakdown of power usage



e.g. In an analysis of several Aust Govt Data Centres the energy used by the mechanical/cooling equipment (53%-77%) exceeded that used by the IT equipment (22%-46%).

http://www.energyrating.gov.au/library/details200905-data-centre-efficiency.html

Metrics: PUE (Power Usage Effectiveness) & DCiE (Data Centre Infrastructure Efficiency)



PUE = Total Facility Power / IT Equipment Power
DCiE = IT Equipment Power / Total Facility Power * 100%

Level of Efficiency

- **3.x Very inefficient**
- **2.5 Inefficient**
- **2.0 Average**
- **1.5 Efficient**
- **1.2 Very Efficient/Current Best Practice**

PUE Example:
Having a facility that uses 100,000 kW of total power of which 80,000 kW is used to power your IT equipment, would generate a PUE of 1.25.

DCiE Example:
Having that same facility that uses 100,000 kW of total power of which 80,000 kW is used to power your IT equipment, would generate a DCiE of 80%.

Source: Green Grid

Standards and Regulation

- USA
  - EPA ***Energy Star*** program
    - Servers
      - Current: Tier 1 requirements for standard/commodity servers - May 2009
      - Future: Tier 2 for Blade and >4-socket servers
    - Data Centres
      - Draft rating tool for Data Centres expected during 2009
- Australia
  - DEWHA developing national ICT energy efficiency strategy for PCs, Data Centres, and ICT peripherals. Minimum Energy Performance Standards (MEPS), based on ***Energy Star*** standards for computers & monitors, targeted for October 2010. Standards for Data Centres will follow.

Energy Efficiency in new Data Centre projects

- Location, Location, Location
  - Build in cool/cold climate – chiller-less
  - Build near power plant (e.g. Hydro)
  - Floating (Google & IDS proposals, etc)

Microsoft's chiller-less data centre in Dublin, Ireland.

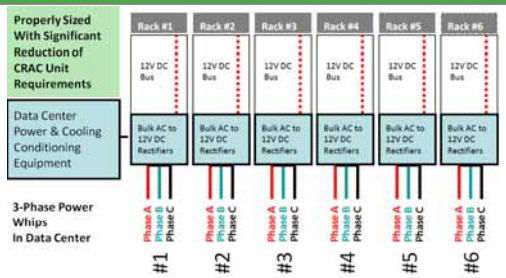
- Mobile, "containerised"

SGI 40-foot container holds 33,600 cores or 14.2PB, including cooling 80% reduction in cooling vs Data Centre

Microsoft's "Gen 4" Data Centre in Chicago will hold over 60 containers. Aiming for PUE of 1.125 by 2012

Energy Efficiency – Server Infrastructure

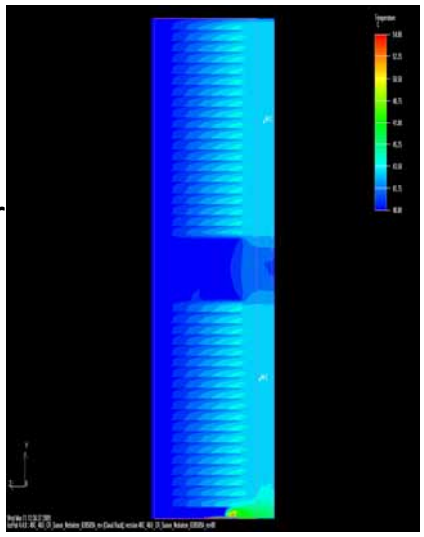
- Power Distribution
- Power Supplies
  - AC
    - 80 PLUS® certified - up to approx 90% efficient
  - DC
    - Rack-level ***and row-level*** AC->DC rectification
    - Used within racks, 99% efficient, 2M hour MTBF
    - 48vDC also supported
- Fans
  - Autonomic fans with dynamic monitoring & speed adjustment
  - Fewer, larger, variable-speed fans using only 4W per fan



Eliminates "stranded power" with near perfect (>95%) phase balance

Energy Efficiency – Server Infrastructure (cont)

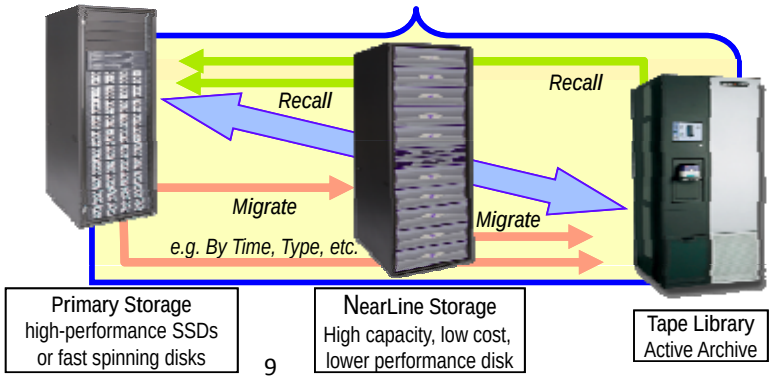
- Airflow
  - Uniform flow pattern throughout cabinet
    - Move air short distance: Front➔Rear
      - Low-power fans utilised
    - Hot components at rear of boards
      - Not blowing hot air over components
- Operate at higher temperatures
  - up to 40°C (104°F)
  - PUE down to 1.2 supported



Thermal Simulation At 40°C Ambient

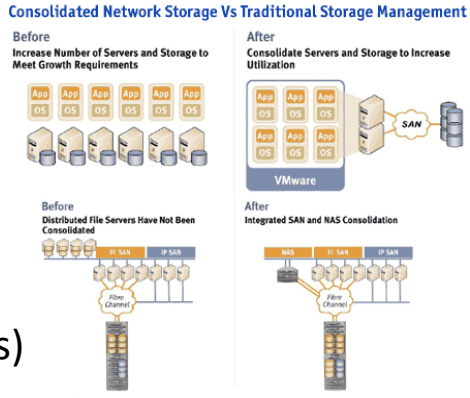
Energy Efficiency – Storage Infrastructure

- Technology – reduce constantly spinning disks
  - Tiered storage
    - SSDs can improve performance and save power
    - Use "greenest" disks for lower-performance direct-access
    - (D-)MAID stops disks spinning when not needed
    - ***HSM – transparently migrate data to tape*** (e.g. SGI's **DMF**)



Energy Efficiency – Storage Infrastructure (cont)

- Strategies
  - Storage Virtualisation & "Thin Provisioning"
    - Applications "see" more storage than is physically present
    - Defer storage acquisition until you really need it
  - Storage Consolidation
    - Consolidate available storage into larger pools – can be done physically (e.g. SAN) or logically (e.g. software-based pooling across multiple separate servers)

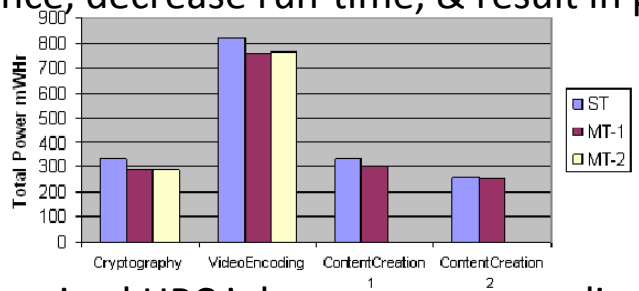


Energy Efficiency – Software Strategies

- General
  - Server Consolidation and Virtualisation (USA IT $ focus)
  - Remote Power Monitoring & Management (EU IT $ focus)
  - Kernel capabilities
    - Linux and Windows support:
      - Core-parking and CPU P-state management
      - SATA Aggressive Link Power Management (ALPM)
      - etc...
    - Linux also supports:
      - Dynamic interrupt request (IRQ) balancing (power-and-performance-aware)
      - Configurable options: *VM writeback period, disk I/O aggregation/period, optical device poll period, etc...*

Developing Energy Efficient Software

- Speed matters
  - "Embarassingly parallel" tasks = no synchronisation
    - Power savings are achieved by finishing jobs faster and then allowing the CPU(s) to enter idle / power-saving mode (or, even better, power-down their server).
    - For multi-core, techniques such as multi-threading increase performance, decrease run-time, & result in power savings



  - Highly synchronised HPC jobs are more complicated...

Source: Intel, "Creating Energy-Efficient Software", Steigerwald, Chabukswar, Krishnan, De Vega

Tips for developing energy-efficient software

- *Underlying assumption: choose best algorithm(s)*
- Choose your language and compilers...
  - Worth paying for performance, support of new / current hardware instructions
    - Recent customer example: 45% performance improvement
- ...and libraries... (e.g. Intel IPP, MKL, etc)
  - Optimised for your CPU instruction set
- ...and tools (e.g. Intel VTune, TBB, Parallel Studio, etc)

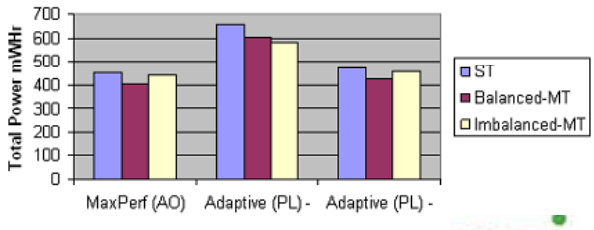
SGI prefers Intel compilers, libraries & tools

Tips for developing energy-efficient software (2)

- Leverage new Instruction Set features
  - e.g. SGI partner with Financial Services application for computing credit risk scenarios
    - Refactored code to “vectorise” computation
    - “Invasive” code change resulted in performance improvement of approximately 64x !!
    - A better way... “Non-invasive” vectorisation (C/C++)
      - “Through the use of templated expressions, we can take ‘garden variety’ pricing code and vectorise it by templating market-dependent variables and introducing an ‘if’ that handles branching over different market scenarios.”
      - Lets the compiler do all the hard work, and makes the code simpler and easier to maintain.

Tips for developing energy-efficient software (3)

- Should the work be distributed evenly amongst multiple CPU cores, or will dedicating CPU cores to threads improve overall performance?
  - Generally, the best performance and power savings are achieved by distributing work across cores, not binding ("affinitizing") threads to specific cores, and enabling all hardware performance-and-power-management features.
  - Affinity ***may*** be beneficial



  - ***Note***: Not optimal for real-time response

Source: Intel, "Creating Energy-Efficient Software", Steigerwald, Chabukswar, Krishnan, De Vega

Tips for developing energy-efficient software (4)

- Computational Efficiency – things to look for
  - Let application "idle" for as long as possible
    - Avoid high-rate periodic, repetitive timers that wake app
  - Loops
    - Minimise the use of tight loops – "unroll" them
      - Don't unroll too much code – may affect cache hit rate
    - Avoid "polling" loops – use event-driven code
      - If polling required, use longest poll interval possible
    - Eliminate busy-wait loops ("spinloops")
      - If spinning on an "atomic" memory instruction, use "pause" instruction – avoids memory order violation performance penalty

Source: Intel, "Creating Energy-Efficient Software", Steigerwald, Chabukswar, Krishnan, De Vega

Tips for developing energy-efficient software (5)

- "Use the cache, Luke..."
  - Prefer array (cache) stride = 1; avoid large power-of-2- strides and dimensions ➔ cache thrashing
  - If run-time initialisation is required (☹), perform the initialisation as close as possible to the time when the data will be used / accessed.
    - Have experienced efficiency gains of almost 50% for code segments that have initialised data resident in cache.
- Use "string" instructions on "Nehalem" CPUs
  - when possible...they run like the clappers...

Tips for developing energy-efficient software (6)

- Data Efficiency
  - Minimise data movement (e.g. try to avoid copying)
  - Disk I/O
    - Read/Write files in large blocks (XFS helps with this!)
      - SGI "Flexible File I/O" (FFIO) provides user-defined buffering
    - Implement a buffering strategy to reduce disk I/O
    - Utilise "intelligent" controllers & disks
      - If SATA disks, ensure NCQ enabled - reduces seeks & energy usage
    - Avoid multiple thread competing for disk I/O
      - Consolidate/synchronise I/O, else thrashing may result

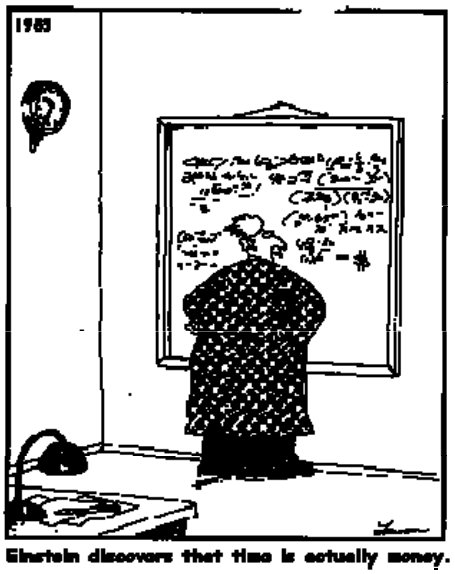
Source: Intel, "Creating Energy-Efficient Software", Steigerwald, Chabukswar, Krishnan, De Vega

Tips for developing energy-efficient software (7)

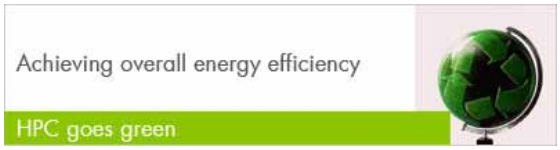
- Data Efficiency
  - Alignment
    - Align data structures so that every data element is aligned to a natural operand size address boundary
      - Current processors won't "fault" on non-aligned data, but accessing aligned data is often faster (than non-aligned data).
    - Arrange data structures to facilitate sequential access
      - Sequential access patterns enable a hardware pre-fetcher to improve performance.
  - To allow best use of Intel vector/SIMD instructions
    - Arrange data in "Structures of Arrays" (SoA), rather than traditional "Arrays of Structures" (AoS) format.
      - AoS data can be "swizzled" into SoA format before use

Developing Energy Efficient HPC Software

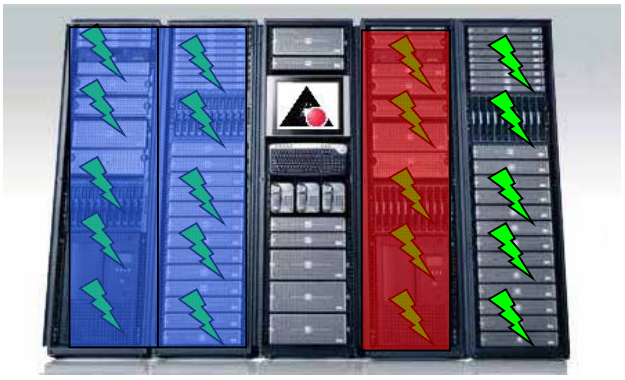
- Why bother?
  - Individual apps matter.
    - An individual app can drive an HPC system for an extended period of time. The behaviour of the app influences the power usage over this time.
  - Systems becoming power-constrained
    - We're now seeing requirements where an HPC solution must run in a limited power envelope.
      - Example: 200kW power limit for major HPC LOB application.
      - Data Centres charging by the kW, not by floor space, etc.



HPC Clusters: “Green Provisioning”



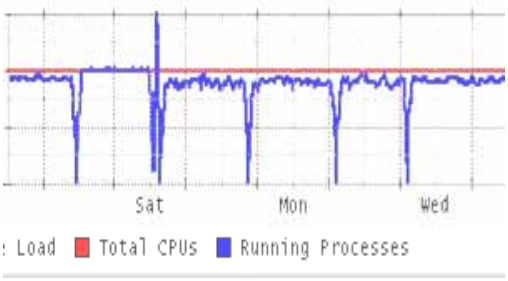
- Green Provisioning™ is a software solution from Altair to reduce the power consumption of HPC systems.
- This solution is built on top of the HPC workload manager from Altair: PBS Professional®
- No special hardware requirements except that Network booting of machines should be enabled (which is typically the configuration for HPC machines anyway)
- The solution uses a popular “Wake-On-Lan” open-source tool, along with additional Altair software.



Green Provisioning™ feature of PBS auto-shuts down the machines not in use. Only required resources are powered on.

HPC Clusters: “Green Provisioning”

- PBS Pro “Green Provisioning”
  - Shutdown service
    - runs in each PBS job monitoring and execution daemon (MOM)
    - checks the load average of a cluster node periodically. If the load average is below a (configurable) threshold value), it starts a (configurable) timer and shuts down the node upon timer expiry.



HPC Clusters: “Green Provisioning”

- PBS Pro “Green Provisioning”
  - Wakeup service
    - Pre-requisite: Nodes are “network bootable” with WOL
    - Runs at configurable intervals. e.g. 30 mins
    - Checks for Queued jobs
    - Acts when enough queued work (waiting jobs)
    - Checks for nodes that are “asleep” (i.e. shutdown)
    - Sends a Wake-On-LAN packet to the required node(s)
      - throttled to avoid “storms”
    - The PBS server submits jobs to the appropriate nodes after the completion of the current scheduler iteration

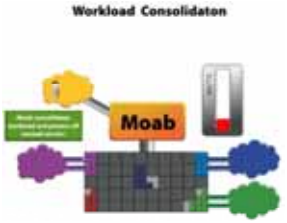
HPC Clusters: “Green Provisioning”

- PBS Pro “Green Provisioning”
  - Result
    - Significant savings
      - Regional example:

        Crest Animation Studios
        - » Operational savings of approx 20% on electricity
        - » Overall ROI: 6 months to break-even on PBS GridWorks

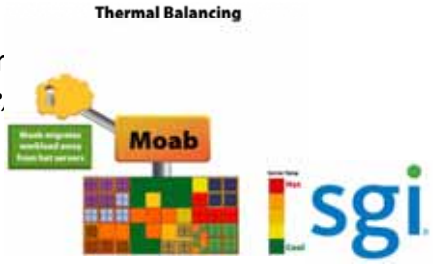
HPC Clusters: Moab Energy Saving

- **Moab:** a "meta-manager" – a "manager of managers"
  - maximizes server workload to boost performance per watt
    Leveraging both traditional workload packing and virtualization technologies, Moab can consolidate workloads from underutilized servers onto fewer servers.



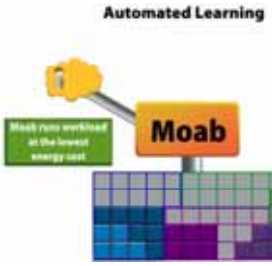
  - puts idle servers into standby, sleep, or power-off
    Using Operating System or hardware facilities (e.g. IPMI 2).
  - allows hot servers to cool down
    With temperature-aware workload scheduling, Moab directs workload away from overheated servers so they can cool down faster, thereby reducing the demand on cooling systems.

  - routes workload to the most energy-efficient server
    Using IPMI 2 and other tools to gather server temperature, utilization, and watts-used statistics, Moab can route workloads to the most energy-efficient resources, based on current and historic node temperatures and cost-per-watt data.



HPC Clusters: Moab Energy Saving

- **Moab:**
  - learns behaviour of individual jobs and adjust job management accordingly
    Moab can learn the behaviour and resource utilisation of individual jobs and use this information to optimally schedule future job instances.



  - takes advantage of off-peak hours
    Moab can automatically schedule lower-priority workloads for processing during off-peak hours, when energy costs are lower, while still ensuring that quality-of-service guarantees are met.
  - sends workload to locations with the lowest energy rates
    Moab can send workload to other IT locations which have lower energy costs, taking into account such factors as availability of resources, workload start-time constraints, data-transmission times, and required service levels.

  - monitors and reports on system energy use
    Moab's tracking, monitoring, and reporting capabilities enable you to manage and document your organization's energy efficiency and potentially provide the ability to track carbon credits or other statistics for chargeback and reporting purposes.

HPC Clusters: Windows HPC Server 2008

- Both PBS Pro and Moab can manage, and interoperate with, an HPC cluster running Windows HPC Server 2008 (R2).
- The native Windows HPC Server 2008 Job Scheduler cannot presently consider a node's "custom resources" (such as power, heat, etc) when scheduling jobs – but such capabilities are on the product roadmap...

sgi

Energy Efficient HPC Strategies

- Understand the application(s)!!
  - Degree of parallelism possible (Amdahl's law)
  - Synchronisation requirements (e.g. between threads)
    - "Embarassingly Parallel" – no synchronisation
    - Highly synchronised – number and frequency of barriers
  - I/O demands – size and frequency of I/O transfers
    - Ensure adequate memory: paging kills performance
- Select an appropriate HPC system architecture
  - Shared Memory (SMP/NUMA)
  - Cluster
    - *Interconnect technology (IB, GbE, etc) & topology*
    - *Disk-full and/or Disk-less nodes*
    - *"Fat-nodes" and/or "Thin-nodes"*
    - *Hybrid (e.g. GPU-equipped) nodes*

sgi

Energy Efficient HPC Strategies (cont)

- Are application accelerators worthwhile?
  - GPUs
    - Typically consume as much power as the rest of a server
    - Can be worthwhile when:
      - You have, or can develop, a GPU-aware version of your app(s) that save multiple servers worth of power and execution time
      - Your app(s) predominantly use Single Precision Floating Point

    (this is changing over time to include integer & DPFP. e.g. NVIDIA's recently announced Fermi architecture)
  - FPGAs
    - Typically consume small amounts of power
    - Can yield large acceleration for integer/parallel apps
    - COTS implementations available for selected apps
    - Need ***significant*** expertise & time to create & modify

"On the energy efficiency of graphics processing units for scientific computing" Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on by: S. Huang, S. Xiao, W. Feng In Fifth Workshop on High-Performance, Power-Aware Computing (HPPAC'09) (2009)

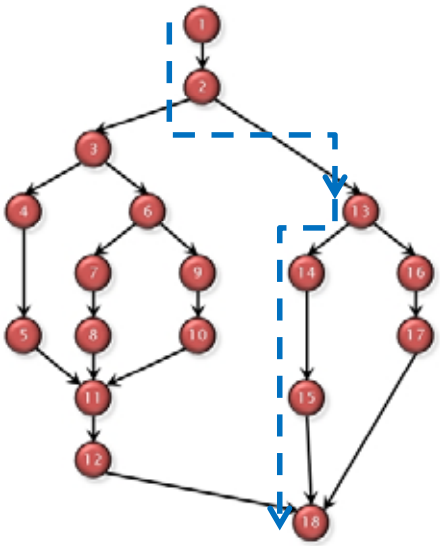
Tips for maximising HPC job efficiency

- "Embarrasingly Parallel" – no synchronisation

  - Start by "oversubscribing" all servers & CPUs with work
  - Let the HPC job manager and server operating systems schedule the work as best they can

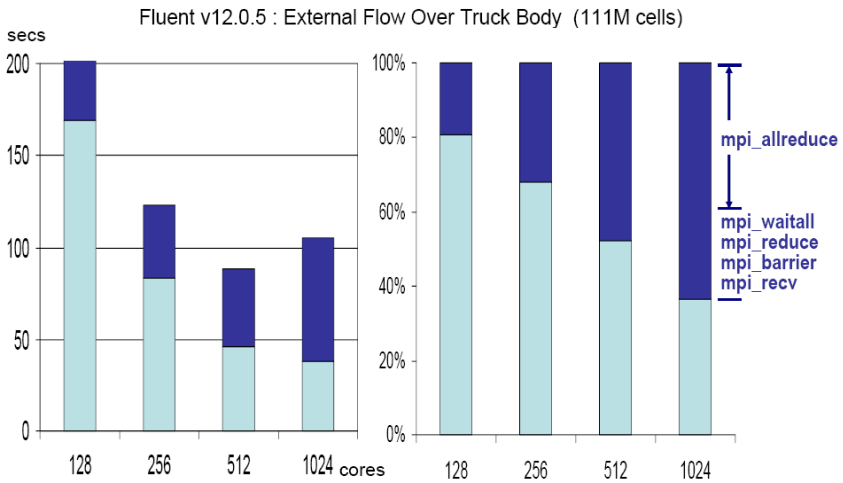
Tips for maximising HPC job efficiency (cont)

- Highly synchronised parallel applications
  - Workload not evenly distributed
  - Some processes (CPUs/servers) finish faster than others and have to wait
  - Longest-executing processes form the **critical path** for a job
  - "Domain decomposition" tends to yield more evenly distributed workloads than "functional decomposition"



Tips for maximising HPC job efficiency (cont)

- Highly synchronised parallel applications
  - Process synchronisation introduces delays:
    - Aim: Minimise the time required to communicate with "remote" CPUs, memory, I/O



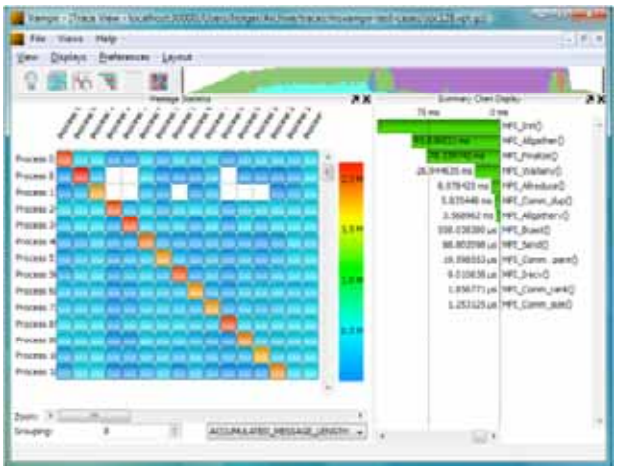
Tips for maximising HPC job efficiency (cont)

- Highly synchronised parallel applications
  - Approach to minimising execution time: Tuning loop:
    - Run and profile application
    - Change code
    - [Change compiler options]
    - [Change platform e.g. CPUs, memory, I/O, interconnect]
  - Small changes can significantly affect the critical path
  - Can adapt the platform to the application
    - e.g. Hybrid cluster with "application accelerators" to overcome unbalanced workloads

sgi

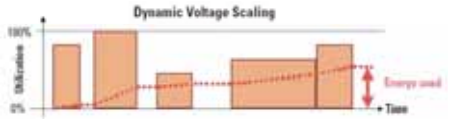
Tips for maximising HPC job efficiency (cont)

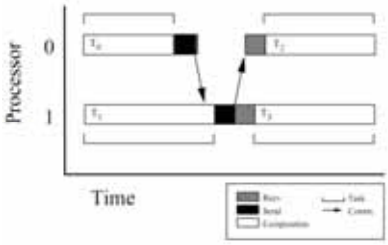
- Highly synchronised parallel applications
  - Primary goal: Minimise synchronisation traffic
    - Send fewer messages containing more data
      - Rather than many messages containing small data
    - Send fewer "collective" messages rather than many point-to-point messages
    - Combine multiple MPI operations
      - e.g. use MPI_Allreduce() rather than MPI_Reduce() + MPI_Bcast()

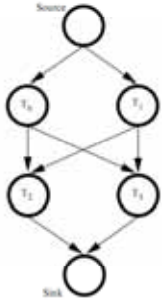


HPC Systems: Next Steps in Energy Saving

- DVS-aware Job Schedulers
  - Dynamic Voltage Scaling (DVS)



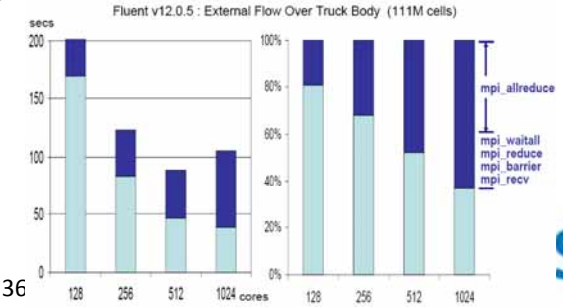




  - New classes of DVS-aware schedulers being developed that utilise their knowledge of parallel (esp. MPI-based) applications and processor DVS capabilities, to ensure timely job completion with minimal energy resources.

Energy Efficient HPC: Conclusion

- HPC energy efficiency is a total package
  - Hardware: CPUs, chipsets, power, racks, fans/cooling, packaging
  - Software: operating systems, job management, applications
- SGI's HPC energy efficiency focus continues on:
  - Motherboard & system design, power & cooling, packaging
  - DMF (i.e. HSM and tiered storage management)
  - ***HPC Application expertise becoming more important***
    - **Have qualified staff who are recognised experts in several HPC-related fields (e.g. climate modelling, computational chemistry, etc)**
  - Minimising synchronisation & communication time for HPC applications



(BTW: Check out SGI's costs-&-carbon-calculator)

www.sgi.com/ecomonitor

sgi® accelerating results™

Thank you