Energy-Aware Processor Management for Virtual Machines

Jan Stoess   **Christian Lang**   **Marcus Reinhardt**

System Architecture Group   University of Karlsruhe, Germany

APP APP APP APP
Guest OS
vCPU vCPU vRAM vNIC vDISK

APP APP APP APP
Guest OS
vCPU vCPU vRAM vNIC vDISK

Hypervisor
CPU CPU CPU CPU RAM NIC DISK

- Increasing Power
- Increasing Density

vCPU
Temperature Power Mgmt.

- Virtualization is currently energy-oblivious
- Some thermal characteristics are indivisible (and thus hard to virtualize)

Guest   vCPU vCPU
Host   CPU CPU

Virtualization introduces two layers of OS-directed resource management

VNIC interposing
Packet decoding
Virtual switch
NIC driver

Virtualization overhead dissipates "unaccounted" energy

APP APP
20W
APP APP
Standard Guest OS
40W vCPU

APP APP
30W
APP APP
Standard Guest OS
30W vCPU

APP APP
25W
Guest +Energy Mar
vCPU

APP APP
APP APP
Guest OS +Energy Management
35W vCPU 35W vCPU 25W

- VM scheduler pursues coarse-grain, low-frequency scheduling according to energy criteria
- Migration (balancing) and/or preemption of virtual processors
- Enforces limits for energy-unaware, defect, or malicious guests

- Energy-aware guest VMs pursue fine-grain, high-frequency migration and/or preemption of threads
- Respect application-level QoS
- Respect global energy criteria through energy-faithful virtualization

VM Scheduler

Hypervisor
60W CPU 60W CPU 60W CPU 60W CPU

- Extensible policies require extensible accounting
- Provide unified energy instrumentation framework
- Flexible user-space VM scheduler/accountant

Standard Guest OS
vCPU

Guest OS +Energy Management
vCPU vPerfCtr

- Provide virtual performance counters for energy-aware guests
- Factor out other guests' energy dissipation
- Enable reuse of existing approach to energy management

VM Scheduler

I/O Driver (Dom0)

virtualize   I/O request   I/O request   report

Hypervisor
CPU access CPU access CPU access
report

- Temperature is hard to virtualize
- Measure per-VM CPU energy dissipation
- Instrumented at context switches, reported in user-accessible logs
- Gauged by thermal temperature sensors

instrument   CPU   PerfCtr

- Framework respects I/O virtualization overhead
- Drivers instrument and report energy consumption per VM
- Hypervisor track per-driver CPU energy dissipation

APP APP
10W 25W 5W
Guest OS
20W vCPU 30W vCPU

APP APP
20W 15W 10W
Guest OS
30W vCPU 20W vCPU

10W   10W

VM Scheduler

- Inject balloon processes to create artificial energy
- Dynamically evens out processor energy imbalances

APP APP APP
Guest OS
°C vCPU vCPU

VM Scheduler

- VCPU hot-plugging as an "energy valve"
- Enables energy-aware self-balancing of guests

APP APP
Guest OS
vCPU

I/O Driver (Dom0)

- Para-virtual I/O interface enables precise accounting at application level

Virtual Machine Architecture powered by L4Ka
http://l4ka.org
http://l4ka.org/projects/virtualization/