Energy Aware Scheduling in Cloud Datacenter

Jemal H. Abawajy, PhD, DSc., SMIEEE
Director, Distributed Computing and Security Research
Deakin University, Australia

Introduction

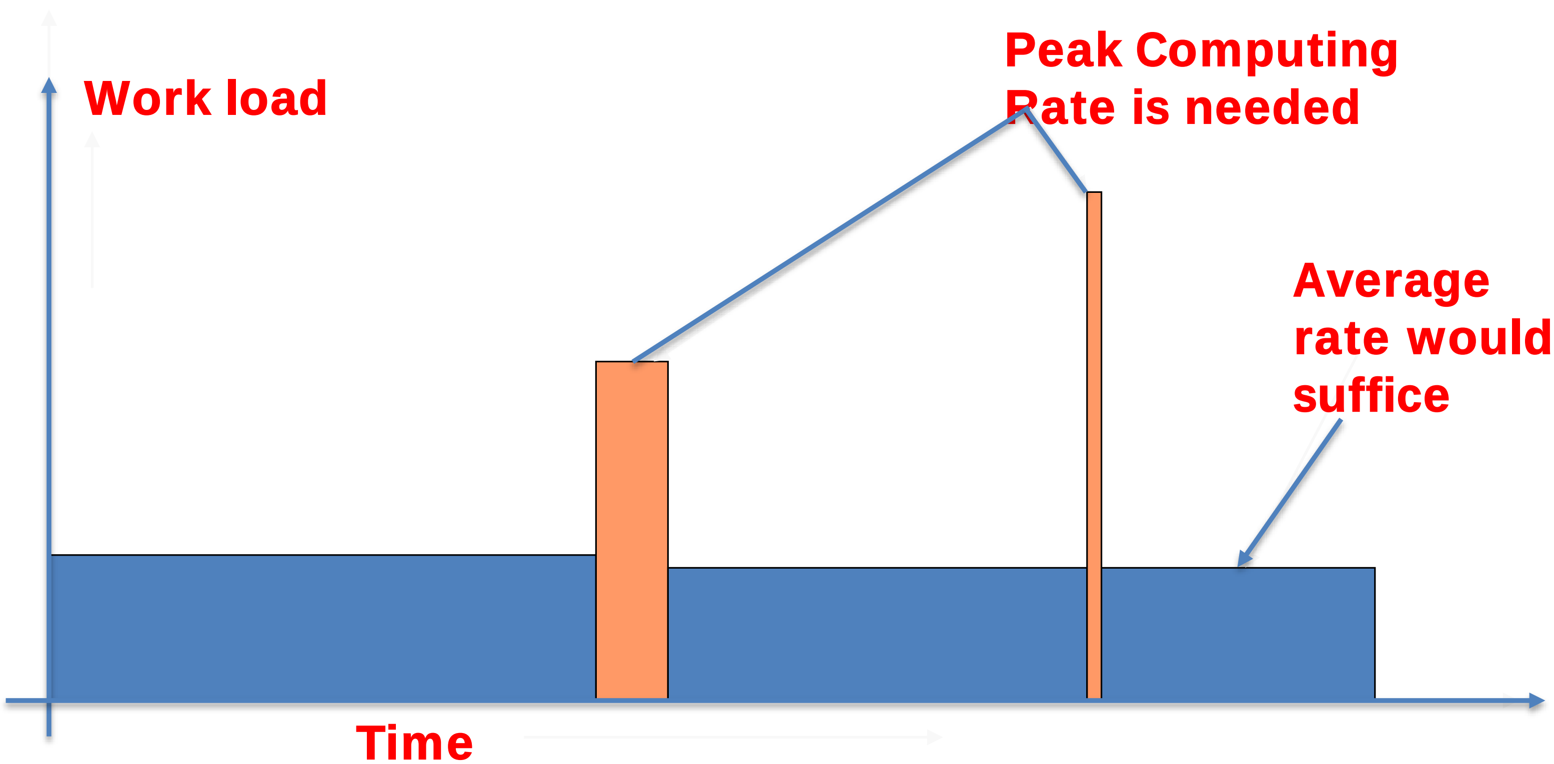
- *Cloud computing* is the delivery of *computing* as a service rather than a product
  - You pay only for what you used
  - Hosted in very large data centres
- Data centres consume high <u>energy costs</u> and <u>huge carbon footprints</u>.
  - **Financial Issues**: In excess of $11 billion in 2010 and cost doubles every five years.
  - **Reliability issues**: For every 10° increase in temperature, the failure rate of a system doubles.
  - **Environmental issues**: Closer to 2% when data center systems are factored into the equation.

How to make cloud computing energy efficiency?

- Cloud hosts a variety of applications
  - Some run for a few seconds (e.g. serving requests of web applications such as e-commerce and social networks portals)
  - Others run for longer periods of time (e.g. simulations or large dataset processing).
- Consider energy as a resource
- Recent devices (CPU, disk, communication links, etc.) support multiple power modes.

Energy Consumption Management

- High performance is needed only for a small fraction of time, while for the rest of time, a low-performance, a low-power processor would suffice.



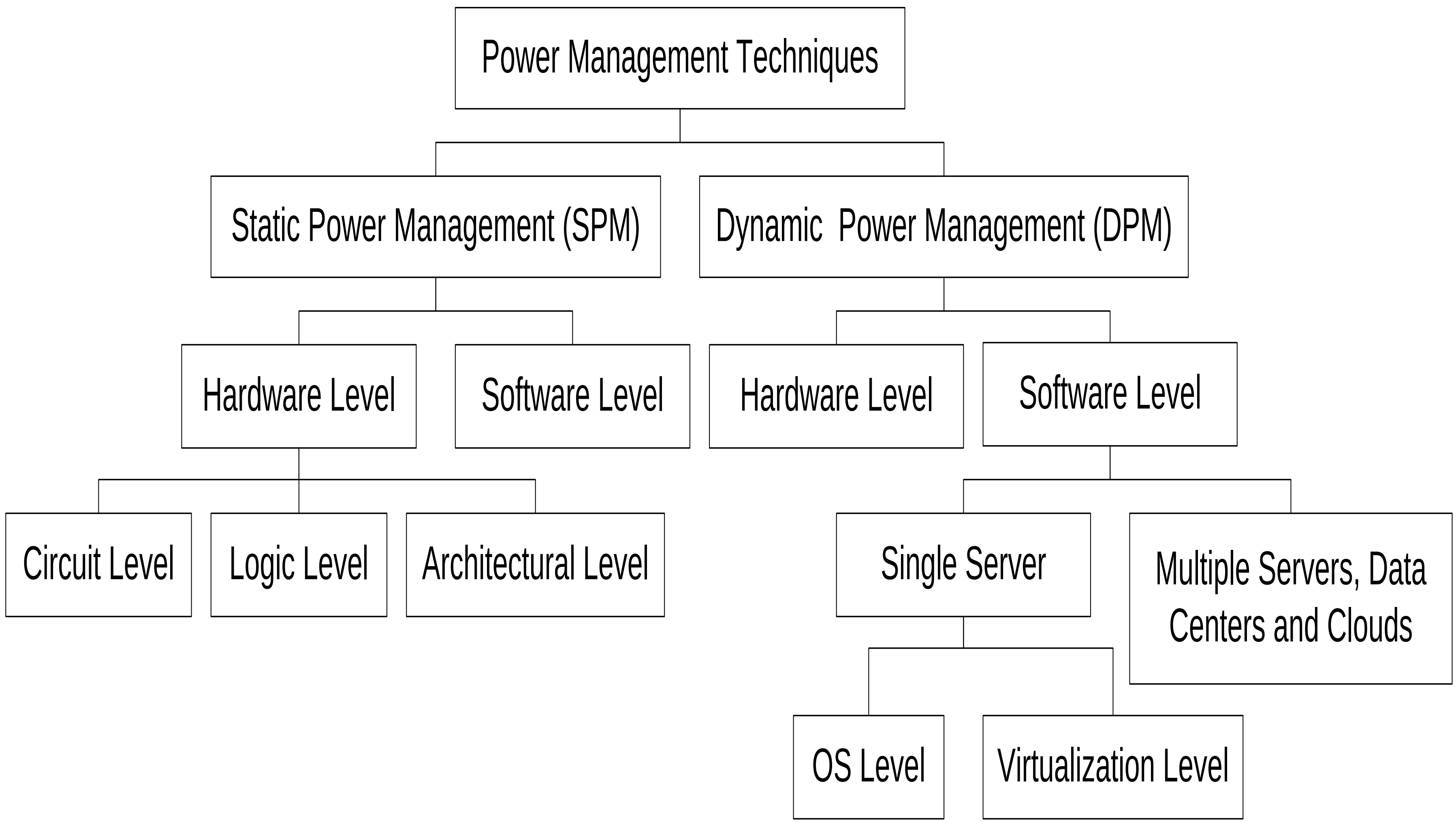
Resource Management for Cloud Computing

- Advanced scheduling to reduce energy consumption
  - Consider energy as a resource when provisioning computing resources to the applications
  - Exploit system level power management to reduce the power consumption.
    - Modern processors operate at multiple frequency levels.
    - The higher the frequency level the higher the energy consumption

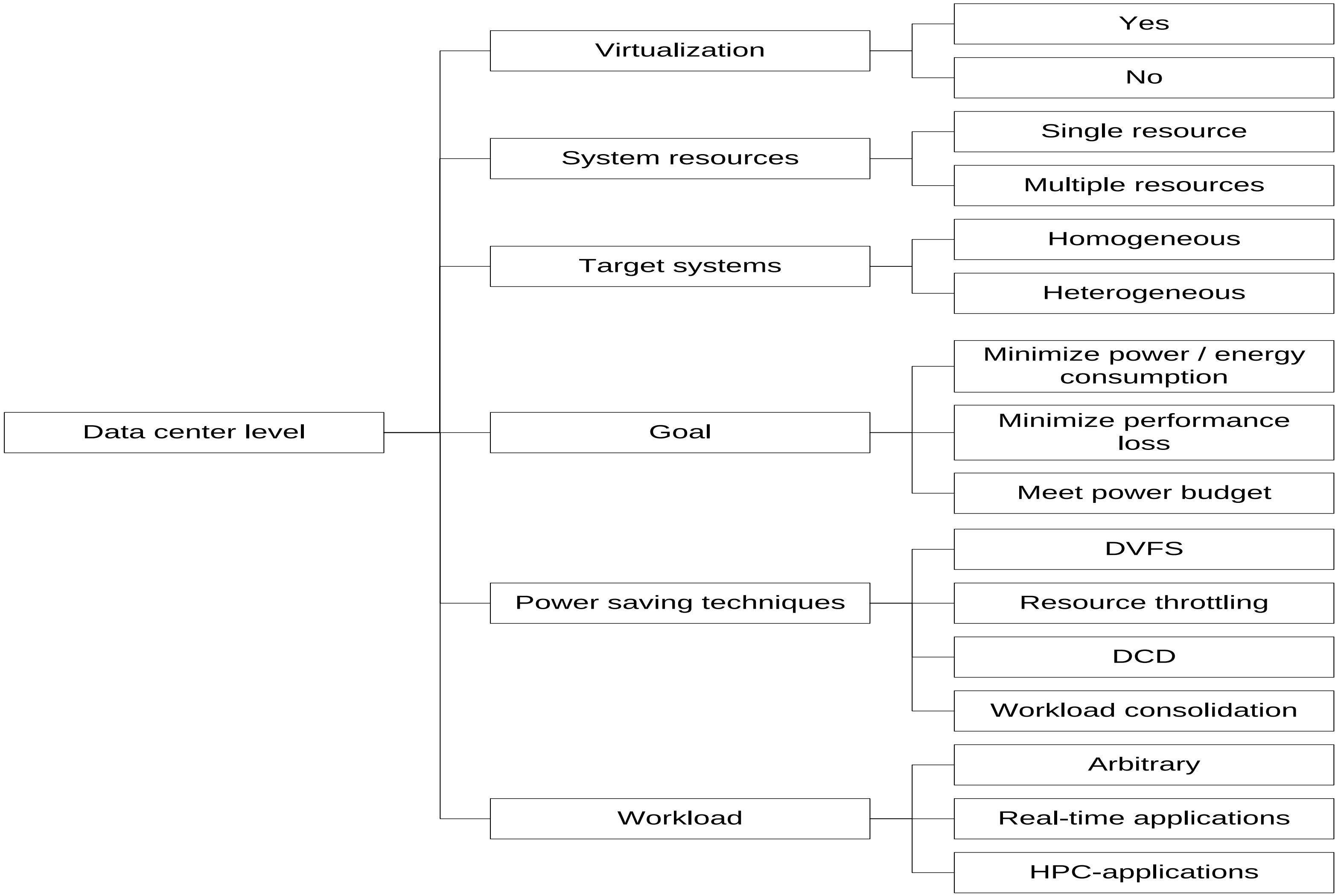
Energy Aware Scheduling in Cloud Datacenter

- Two conflicting objectives of the scheduler:
  - Minimizing energy consumption of the data center
  - Meet the quality of servers (e.g., deadlines)
- Why they are conflicting objectives?
  - Energy reduction is system centric
  - QoS is user centric

Taxonomy of Power Management Techniques



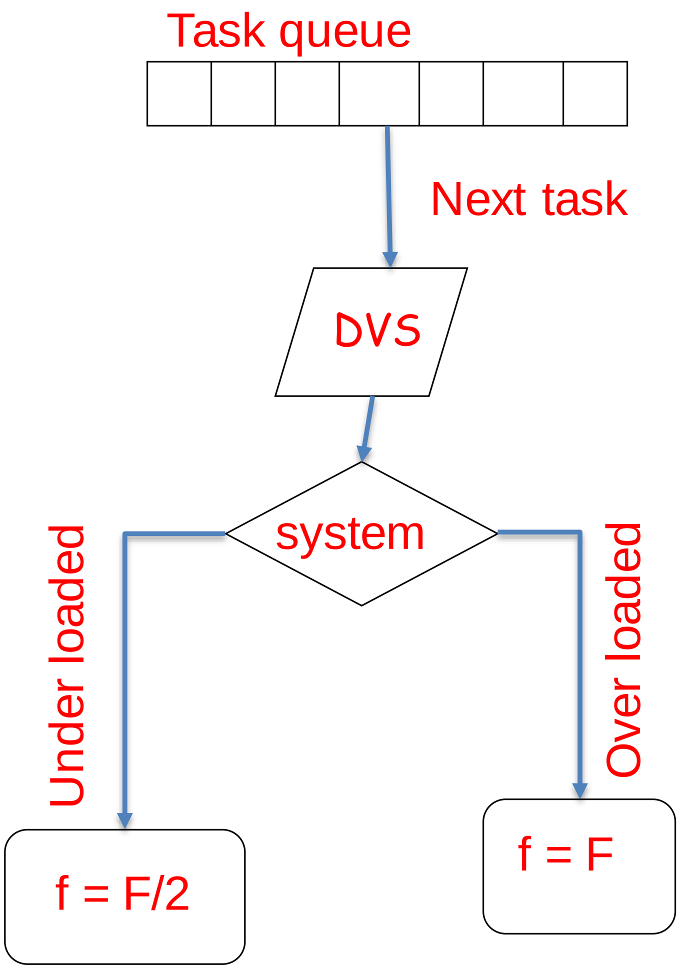
Source: Raj


Data center level
Virtualization
Yes
No
System resources
Single resource
Multiple resources
Target systems
Homogeneous
Heterogeneous
Goal
Minimize power / energy consumption
Minimize performance loss
Meet power budget
Power saving techniques
DVFS
Resource throttling
DCD
Workload consolidation
Workload
Arbitrary
Real-time applications
HPC-applications

Source: Raj

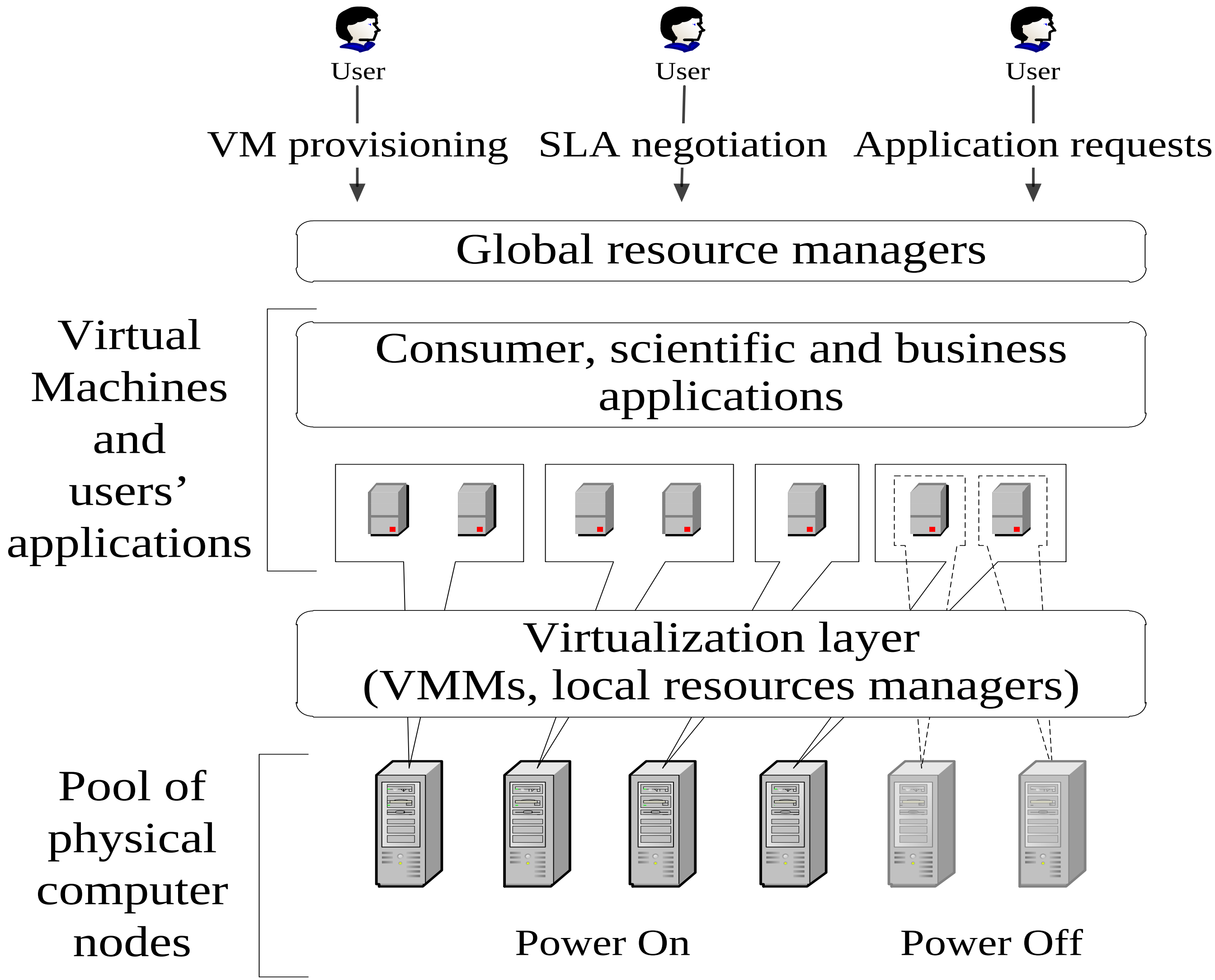
Dynamic Voltage Scaling (DVS)

- The idea is to adjust the supply voltage dynamically
  - DVS scales the operating voltage of the processor along with the frequency (f).
  - Since energy is proportional to f^2, DVS can potentially provide significant energy savings through frequency and voltage scaling.
- Note: if we reduce the frequency we save energy but, we spend more time in performing the same computation



Simple DVS-Scheme

Dynamic VM Consolidation



Application Scheduling

- One effective method: Application Scheduling
  - Consolidate running applications to a small number of servers
  - Make idle servers sleep or power-off
- Migration cost-aware scheduling
  - Task scheduling usually involves energy-cost of virtual machine migration
  - Consider the task migration-cost between servers

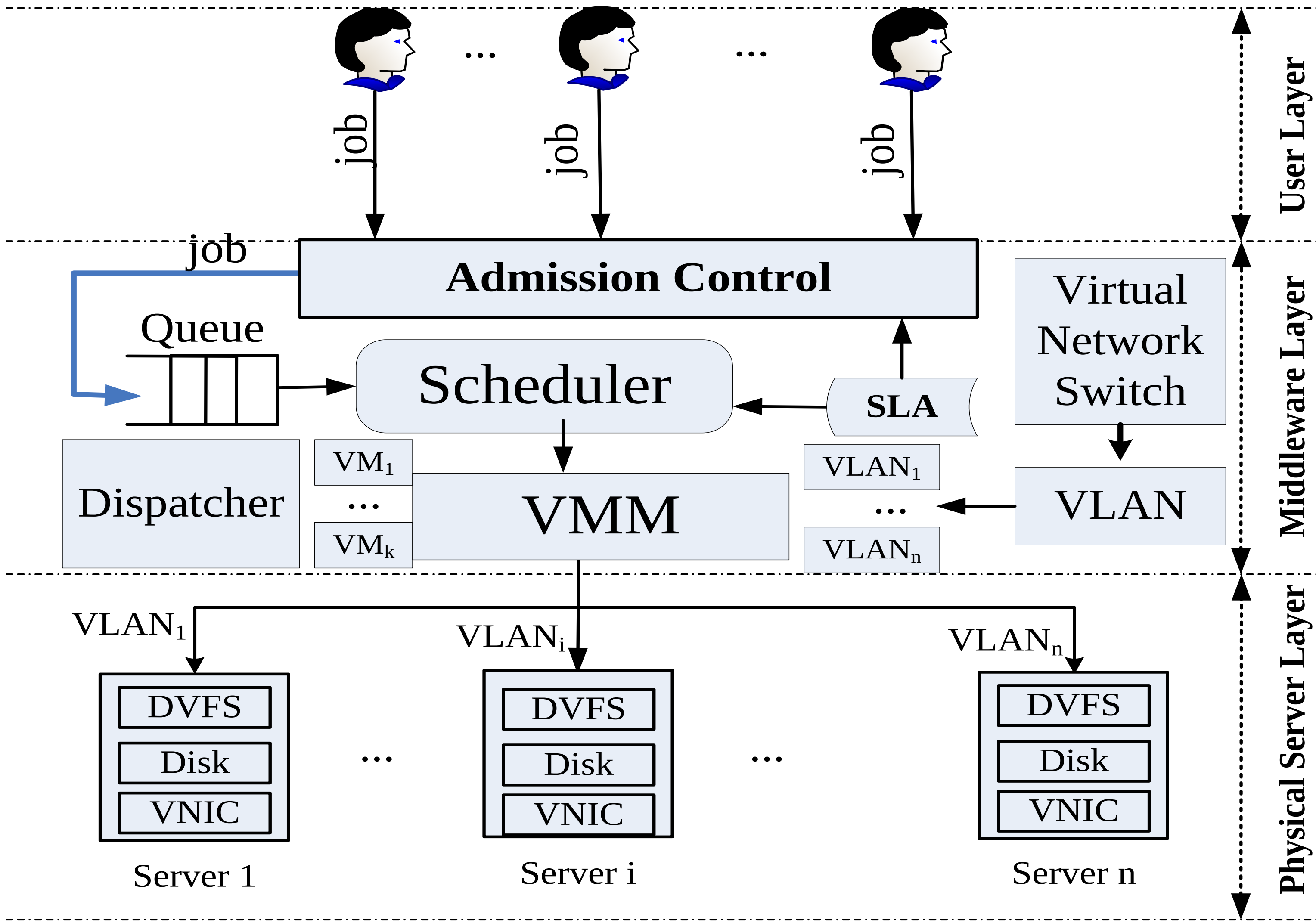
Adaptive Energy-aware Scheduling Algorithm on Virtualized Network Datacenters

- **Goal:** Design scheduling algorithm with aim of further reduce total energy consumption –
  - Considering Computing-plus-communication energy consumption and the VMs reconfiguration and disk energy consumption .
  - Converting non-convexity form of communication energy consumption into convex form,
  - Theoretically supported: Introducing the stochastic service system theory.
  - Experimental evaluating and verifying the effectiveness of the proposed algorithm.

Optimization problem

- The energy optimization problem can be formulated as follows:
  - $\mathbf{minimize}\ (\boldsymbol{E_{total}})$
- where (E_{total}) is defined as follows:
  - $\boldsymbol{E_{total}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \Big(\boldsymbol{E_{cpu}}(\boldsymbol{i}) + \boldsymbol{E_{com}}(\boldsymbol{i}) + \boldsymbol{E_{disk}}(\boldsymbol{i}) +$

System Model



Computational Energy Consumption

- Dynamic power consumption of a CPU is related to processing frequency and supply voltage as follows

$$P = A \cdot C_{eff} \cdot f \cdot V^2$$

  - active percentage of gates,
  - effective capacitance load,
  - processing frequency and
  - the supply voltage of the CPU.

Computational Energy Consumption

- Therefore, the computational energy consumption of a VM *i* ($E_{cpu}(i)$) can be defined as follows:
  - $\boldsymbol{E_{cpu}(i) = \sum_{k=0}^{Q} AC_{eff}(f_k(i))^3 t_k(i) \quad \forall \mathbf{i} = \{1, 2, \dots, M\}}$
    - $f_k(i)$ is the *k*-th discrete frequency of the VM *i*,
    - $t_k(i)$ represents the time that the VM *i* operates at frequency $f_k(i)$
    - *k*=0 means the idle state of VM *i*,
    - *Q* represents the number of CPU frequencies permitted for each VM

VMs reconfiguration energy consumption

- The VMs reconfiguration energy consumption for VM i includes two parts:
  - $E_{reconf}(i) = E_{reconf}(\mathrm{Ext})(i) + E_{reconf}(\mathrm{Int})(i)$
- $E_{reconf}(\mathrm{Ext})(i)$ – external cost for VMs reconfiguration
  - transition from final active discrete frequency of VM *i* for processing the last job to the first active discrete frequency which processes the next incoming job
  - $E_{reconf}(\mathrm{Ext})(i) = k_c \times E_{exter}$
- $E_{reconf}(\mathrm{Int})(i)$ – cost of the VMs reconfiguration internal energy consumption
  - when VM *i* from the current active discrete frequency ($f_k(i)$) to another frequency ($f_{k+1}(i)$) transits
  - $E_{reconf}(\mathrm{Int})(i) = k_c \times \sum_{k=0}^{Q} (f_{k+1}(i) - f_k(i))^2$
- k_c is the coefficient of reconfiguration energy consumption.

Communication Energy Consumption

- Each VM needs to communicate with VMM with a dedicated virtual link,
- As for one-way transmission plus switching operation, the power consumption $\left(p_{net}(i)\right)$ can be described as follows:
  - $p_{net}(i) = p_{net}^{T}(i) + p_{net}^{R}(i)$
  - – $p_{net}^{T}(i)$ is the power consumption of transmitting and switching,
  - – $p_{net}^{R}(i)$ refers to the energy consumption of receiving operation.

Disk Energy Consumption

- The power consumption of storage service ($E_{disk}(i)$) can be calculated as follows:
  - $E_{disk}(i) = \frac{W}{3600}\left(E_T + 1.5\frac{P_{cs}}{C_{cs}}\right) + \frac{2S}{D} \times \frac{P_{hd}}{C_{hd}}$
- *S:* is the size of a task,
- E_T represents the per-bit energy consumption of transmission and switching,
- P_{cs} refers to the energy consumption of the considered server,
- C_{cs} is the capacity of the considered server,
- D represents the number of downloads per hour and this value can be obtained by empiric value,
- P_{hd} is the energy consumption of hard disk arrays (cloud storage),
- C_{hd} represents the capacity of the hard disk arrays.

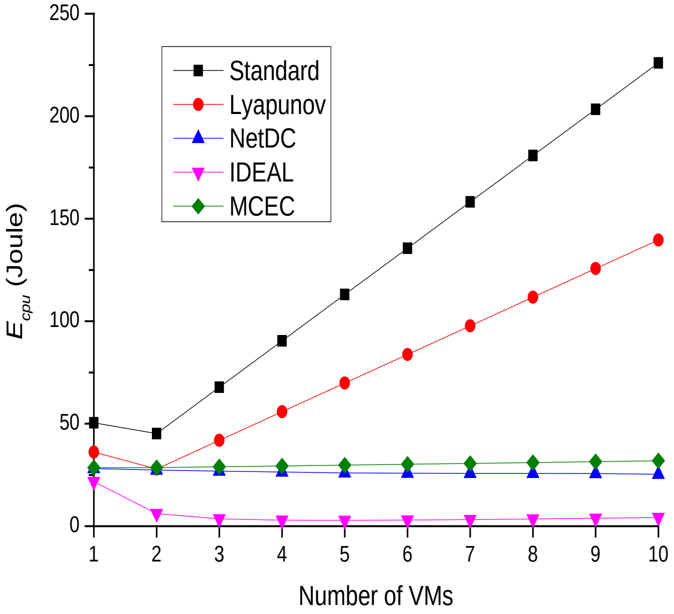
Performance Analysis

Parameter	Value	Parameter	Value
λ	4	Q	5
μ	5	R_{total}	100 (Gbit/s)
m	10	C_{cs}	604.8 (Tb)
η	0.5 (mW)	P_{cs}	4.9 (kW)
k_c	0.05 Joule/(GHz)^2	E_T	1.42×10^{-9} (W/per-bit)
T_{total}	7 (s)	D	20
T	5 (s)	P_{hd}	225 (W)
P_{idle}	0.5 (W)	C_{hd}	800 (Mb/s)

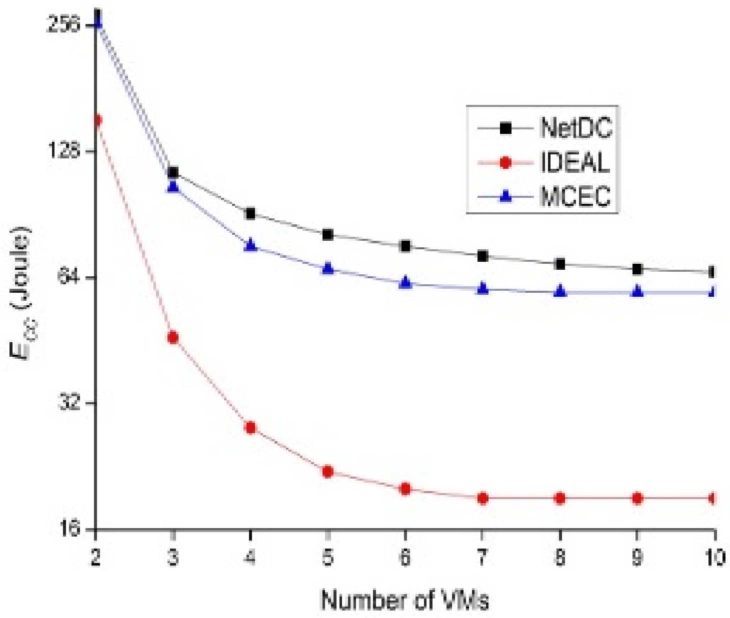
Comparison Algorithms

- Modified the following as they concentrate on computational energy consumption
  - Lyapunov - utilizes the queuing information to maximize the throughput and minimize the energy consumption
  - Standard - uses the DVFS policy to save energy consumption
- The following are just for comparison
  - NetDC - works depended on the calculated proportional of real frequency, which could not be used in real scenarios
  - IDEAL - needs to work on continue ranges of frequencies, which is unfeasible and unrealistic in real environments

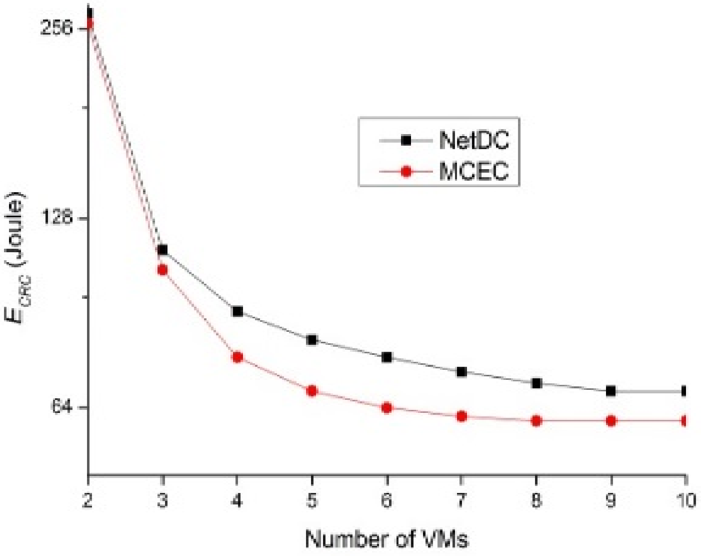
Computational energy consumption



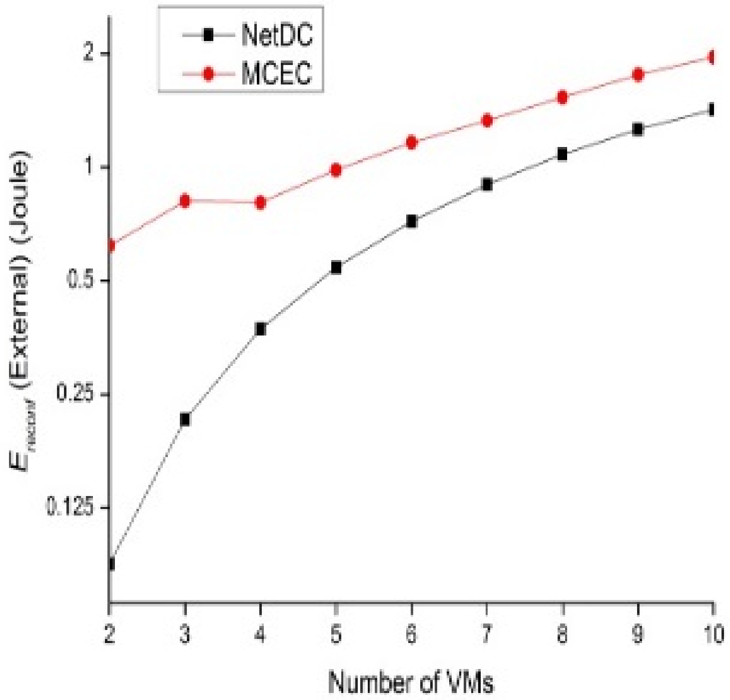
Computational + Communication energy consumption



CPU+Comm+Rec

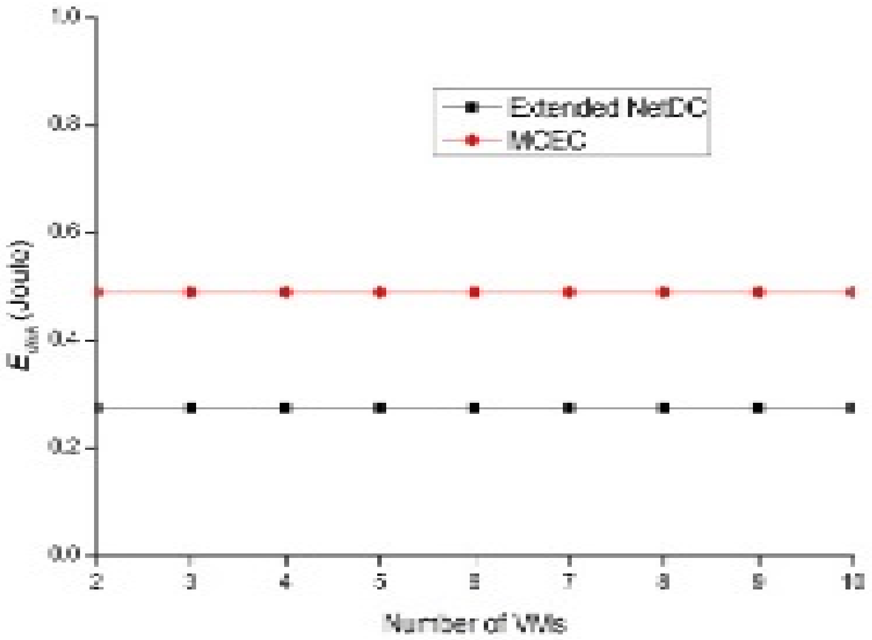


NetDC
MCEC
256
128
64
E_{CRC} (Joule)
2
3
4
5
6
7
8
9
10
Number of VMs



NetDC
MCEC
2
1
0.5
0.25
0.125
$E_{netconf}$ (External) (Joule)
2
3
4
5
6
7
8
9
10
Number of VMs

CPU+Comm+Recnfig+Disk Energy Cost



Summary

- Developed an adaptive energy-aware algorithm that considers four systems components
- The experimental results prove that
  - optimizing total energy consuming resource is more efficient than optimizing only one or two energy consumption;
  - We found that computation energy-consumption (E_{cpu}) and communication energy-consumption (E_{com}) account for the majority,
  - VMs configuration energy consumption (E_{reconf}) and disk energy consumption (E_{disk}) account for only small part.

Future Work

- Combine concepts of both Power-aware and Thermal-aware scheduling to minimize both energy and temperature.
- Integrated server, rack, and cooling strategies.
- Further improve VM Image minimization.
- Designing the next generation of Cloud computing systems to be more efficient.

Thank You