An Energy-aware Scheduling Algorithm in DVFS-enabled Networked Data Centers

CLOSER 2016 - TEEC Session

Mohammad Shojafar, Claudia Canali, Riccardo Lancellotti, and Saeid Abolfazli

Department of Engineering Enzo Ferrari, University of Modena and Reggio Emilia, Modena, Italy

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

April 24, 2016

Agenda

- Introduction
  - Problem in data centers
  - Our contribution
- Model
  - Model Architecture
  - Computing Model
  - Frequency Reconfiguration Model
  - Channel/Communication Model
- Optimization problem and solution
- Performance Evaluation
- Conclusion

Introduction

- Cloud Data Centers: Energy-saving computing is critical
- Our focus is in the Virtualized Networked Data center (VNetDC) supporting cloud
- Qualifying point of our approach, we consider:
  - Traffic exchange in VNetDCs
  - Load balancing for incoming request
  - DVFS (multi-frequency CPUs) hardware technology
- **QoS: processing time + communication time → challenging constraint**

Introduction

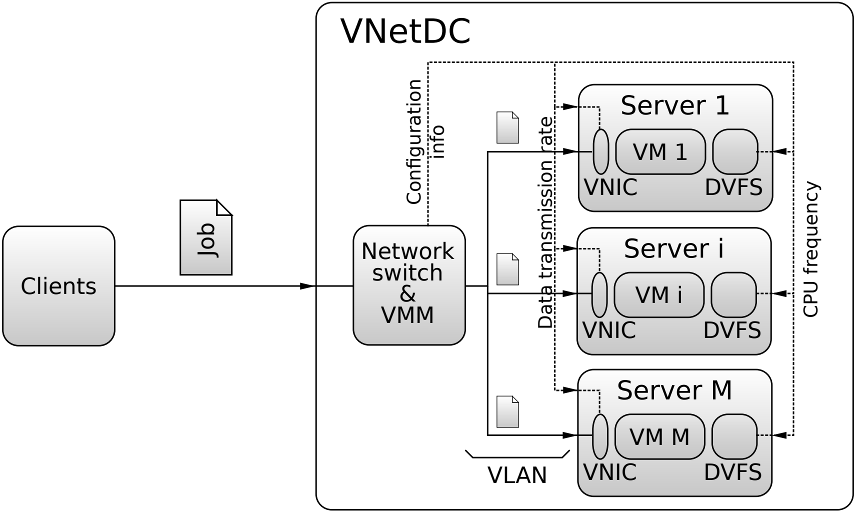
Our solution addresses:

- Minimize the overall energy for the computing-plus-communication resources in VNetDCs
- Guaranteeing the time limit of each task and bandwidth limitation of each server jointly by changing the reconfiguration capability

Detail:

- Dynamic load balancing
- Job = chunk of data to process
- Online job decompositions and scheduling
- Distribute the workload among multiple VMs
- **Solve nonlinear/nonconvex optimization problem**

Model Architecture



Model

Assumptions:

1) Physical servers with DVFS
2) Each server hosts one heterogeneous VM (private cloud scenario)
3) VNetDC comprises M independent congestion-free half-duplex channels
4) A VM on server i is capable to process $F(i)$ bits per second
5) No queue is considered for incoming/outgoing workload into/from the system
6) Data centers utilize off-the-shelf rackmount physical servers, which are interconnected by commodity Fast/Giga Ethernet switches
7) Each job has size of L_{tot}
8) Maximum processing (computation and communication) time for each job is $\overline{T}$ (QoS constraints)

Optimization Problem

Goal: minimize the overall resulting communication-plus-computing energy, formally defined as:

$$\mathcal{E}_{tot} \triangleq \sum_{i=1}^{M} \mathcal{E}_{CPU}(i) + \sum_{i=1}^{M} \mathcal{E}_{Reconf}(i) + \sum_{i=1}^{M} \mathcal{E}_{net}(i) \ \ [Joule], \qquad (1)$$

- $\mathcal{E}_{CPU}(i)$: Computation energy for server i
- $\mathcal{E}_{Reconf}(i)$: Reconfiguration energy for server i
- $\mathcal{E}_{net}(i)$: Channel/Communication energy for server i

Computing Model

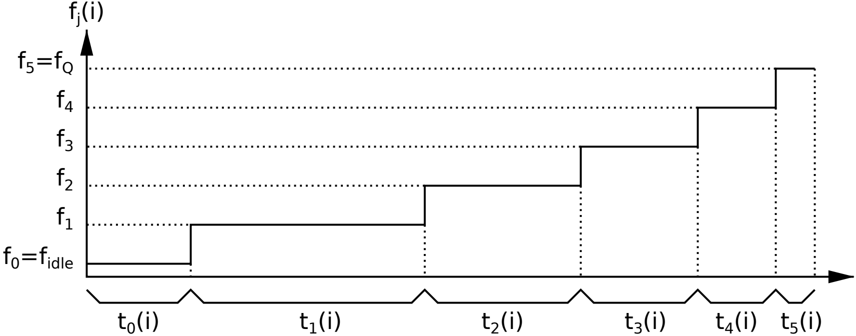
$VM(i)$ **attributes:**

$$\{Q, \mathbf{f}(i), \mathbf{t}(i), f_i^{max}, T,\ i = 1, \ldots, M\}, \tag{2}$$

- Q: number of CPU frequencies allowed for each VM (plus an idle state)
- $\mathbf{f}(i) = \{F_j(i) |\, j = 0, \ldots, Q\}$: discrete frequency set in $VM(i)$–using DVFS
- $f_i^{max} \triangleq F_Q(i)$: maximum available frequency in $VM(i)$
- $\mathbf{t}(i) = \{t_j(i) |\, j = 0, \ldots, Q\}$: discrete time set in $VM(i)$ corresponding to $f_j(i)$ in $VM(i)$
- $\sum_{j=0}^{Q} t_j(i) \leq T$: time allowed the $VM(i)$ to fully process each submitted task, computation only constraint

Computing Model

Fig. 2 illustrates an example for $Q = 5$.



$$\mathcal{E}_{CPU}(i) \triangleq \sum_{j=0}^{Q} AC_{eff} f_j(i)^3 t_j(i),\ [Joule], \forall i = \{1, \ldots, M\}, \qquad (3)$$

A: active percentage of gates; C_{eff}: effective load capacitance

Frequency Reconfiguration Model

Frequency policy: Scale up/down VMs' processing rates at the minimum cost.

We define internal switching cost and external switching cost

Internal switching cost: $f_j(i) \rightarrow f_{j+k}(i)$ (k steps movement to reach the next active discrete frequency)

External switching cost: the cost for external-switching from the final active discrete frequency of $VM(i)$ at the end of a job to the first *active discrete frequency* for the next incoming job of size L_{tot}

$$\sum_{i=1}^{M} \mathcal{E}_{Reconf}(i) \triangleq k_e \sum_{i=1}^{M} \sum_{k=0}^{K} (\Delta f_k(i))^2 + Ext\_Cost \qquad (4)$$

k_e $(J/(Hz)^2)$:an unit-size frequency switching

$\Delta f_k(i) \triangleq f_{k+1}(i) - f_k(i)$

$Ext\_Cost \triangleq k_e \, M(f_Q^t - f_0^{t-1})^2$

Channel/Communication Model

Shannon-Hartley exponential formula

$$P_{net}(i) = \zeta_i \left(2^{R(i)/W_i} - 1\right) + P_{idle}(i),\ [Watt], \quad (5)$$

- $\zeta_i \triangleq \frac{\mathcal{N}_0^{(i)} W_i}{g_i}$, $i = 1, \ldots, M$–noise spectral power density
- $\mathcal{N}_0^{(i)}$ (W/Hz)
- W_i (Hz) Transmission bandwidth
- $R(i)$: Transmission rate over link i
- g_i: gain of the i-th link

i) One-way transmission delay: $D(i) = \sum_{j=1}^{Q} F_j(i) t_j(i)/R(i)$

ii) $\max_{1 \leq i \leq M}\{2D(i)\} + T \leq \overline{T}$. (Minimize the slowest VM)

$$\mathcal{E}_{net}(i) \triangleq P_{net}(i) \left(\sum_{j=1}^{Q} \frac{F_j(i) t^j(i)}{R(i)} \right)\ [Joule]. \quad (6)$$

Optimization problem and solution

$$\min \sum_{i=1}^{M} \mathcal{E}_{CPU}(i) + \sum_{i=1}^{M} \mathcal{E}_{Reconf}(i) + \sum_{i=1}^{M} \mathcal{E}_{net}(i) \tag{7.1}$$

$$\text{s.t.:} \sum_{i=1}^{M} \sum_{j=0}^{Q} F_j(i) t_j(i) = L_{tot}, \tag{7.2}$$

$$\sum_{i=1}^{M} R(i) \leq R_t, \tag{7.3}$$

$$\sum_{j=0}^{Q} t_j(i) \leq T, \quad i = 1, \ldots, M, \tag{7.4}$$

$$\sum_{j=0}^{Q} \frac{2F_j(i) t_j(i)}{R(i)} \leq \overline{T} - T, \;\; i = 1, \ldots, M, \tag{7.5}$$

$$0 \leq t_j(i) \leq T, 0 \leq R(i) \leq R_t, \; i = 1, \ldots, M, \; j = 0, \ldots, Q, \tag{7.6}$$

Optimization problem and solution

(6.1) Eq. (7.1) is the objective function which consists of the sum of three terms which accounts for the computing energy, the reconfiguration energy cost is the networking energy

(6.2) Eq. (7.2) is the (global) constraint which guarantees that the overall job is decomposed into M parallel tasks $F_j(i)t_j(i)$ is the workload processed for each discrete frequency f_j which is processed by VM i during the interval $t_j(i)$

(6.3) Eq. (7.3) ensures that the bandwidth summation of each VM must be less than the maximum available bandwidth of the global network

(6.4) Eq. (7.4) is the constraint on computation time

(6.5) Eq. (7.5) guarantees that the duration of each computing interval is no negative and less than T

Optimization problem and solution

1) We can simplify communication part as:

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} 2P_{net}(i)\left(\frac{F_j(i)t_j(i)}{R(i)}\right) = (\overline{T} - T)\sum_{i=1}^{M}\sum_{j=0}^{Q} P_{net}(i)\left(\frac{2F_j(i)t_j(i)}{\overline{T} - T}\right) \tag{8}$$

2) The problem feasibility:

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} F_j(i)t_j(i) \leq R_t(\overline{T} - T)/2 \tag{9}$$

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} F_j(i)t_j(i) \leq \sum_{i=1}^{M} Tf_i^{max}. \tag{10}$$

Performance Evaluation-Simulation setup

i) Comparison with
   - Standard (or Real) available DVFS-enabled technique (Kimura et al., 2006),
   - Lyapunov (Urgaonkar et al., 2010)
   - IDEAL no-DVFS (Mathew et al., 2012) and NetDC (Cordeschi et al., 2010) **[Theoretical Lower bounds]**

ii) CVX solver (Grant and Boyd, 2015) + MATLAB

iii) **Three** different scenarios: **two** synthetic workloads and a **real-world workload** trace

iv) L_{tot}: $[\overline{L}_{tot} - a, \overline{L}_{tot} + a]$

Performance Evaluation-Simulation setup

Significant parameters and sensevity analysis:

- $\overline{\mathcal{E}}_{tot} \triangleq \frac{1}{Max\_slot} \sum_{i=1}^{Max\_slot} \sum_{i=1}^{M} \mathcal{E}_{tot}(i)$
- $\overline{\mathcal{E}}_{CPU} \triangleq \frac{1}{Max\_slot} \sum_{i=1}^{Max\_slot} \sum_{i=1}^{M} \mathcal{E}_{CPU}(i)$
- $\overline{\mathcal{E}}_{Reconf} \triangleq \frac{1}{Max\_slot} \sum_{i=1}^{Max\_slot} \sum_{i=1}^{M} \mathcal{E}_{Reconf}(i)$
- $\overline{\mathcal{E}}^{net} \triangleq \frac{1}{Max\_slot} \sum_{i=1}^{Max\_slot} \sum_{i=1}^{M} \mathcal{E}_{net}(i)$
- k_e, ζ
- T, $\overline{T}$ (QoS parameters)
- AET= average execution time

First Scenario

$\overline{L}_{tot} \equiv 8$ $[Gbit]$ $a = 2$ $[Gbit]$
DVFS: Intel Nehalem Quad-core Processor (Kimura et al., 2006) called $F1 = \{0.15, 1.867, 2.133, 2.533, 2.668\}$

Table: Default values of the main system parameters for the first test scenario.

Parameter	**Value**	**Parameter**	**Value**
PE=M	$[1, \ldots, 10]$	$\overline{T}$	7 $[s]$
T	5 $[s]$	R_t	100 $[Gbit/s]$
C_{eff}	1 $[\mu F]$	k_e	0.05 $[Joule/(GHz)^2]$
F	$F1$ $[GHz]$	Q	5
A	100%	P_i^{idle}	0.5 $[Watt]$
ζ_i	0.5 $[mWatt]$	f_i^{max}	2.668 $[GHz]$

Second Scenario

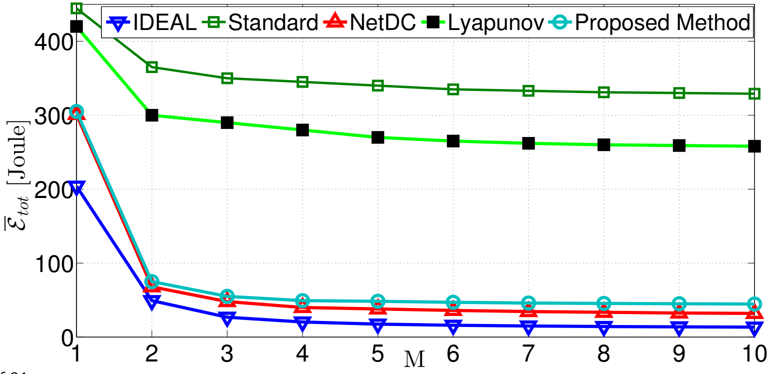
$\overline{L}_{tot} \equiv 70$ $[Gbit]$ $a = 10$ $[Gbit]$
DVFS: Crusoe cluster with TM-5800 CPU in (Almeida et al., 2010), e.g., $F2 = \{0.300, 0.533, 0.667, 0.800, 0.933\}$

Table: Default values of the main system parameters for the second test scenario.

Parameter	**Value**
k_e	0.005 $[Joule/(GHz)^2]$
Q	5
F	$F2$ $[GHz]$
$\overline{L}_{tot}$	70 $[Mbit]$
M	$\{20, 30, 40\}$
f_i^{max}	0.933 $[GHz]$

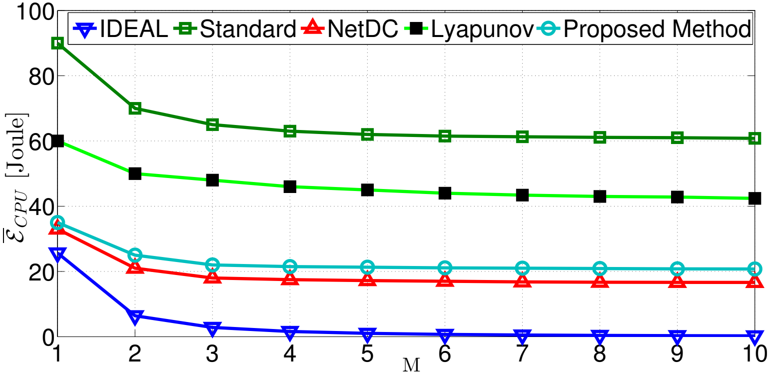
$\overline{\mathcal{E}}_{tot}$-vs.-$M$

- $\uparrow M \propto \overline{\mathcal{E}}_{tot} \downarrow$
- **The average energy-saving of the proposed method is approximately 50% and 60% compared to Lyapunov-based and Standard schedulers, respectively**



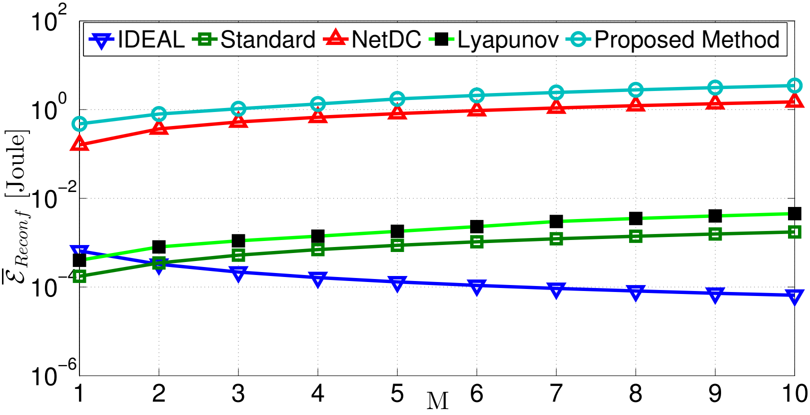
$\overline{\mathcal{E}}_{CPU}$-vs.-$M$

- $\uparrow M \propto \overline{\mathcal{E}}_{CPU} \downarrow$
- **The average energy-saving of the proposed method is approximately 25% and 33% compared to Lyapunov-based and Standard schedulers, respectively**



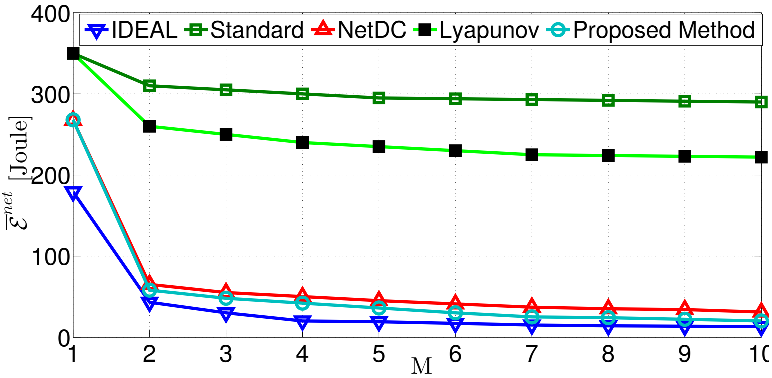
$\overline{\mathcal{E}}_{Reconf}$-vs.-$M$

- $\uparrow M \propto \overline{\mathcal{E}}_{Reconf} \uparrow \ll \overline{\mathcal{E}}_{CPU}$ or $\overline{\mathcal{E}}^{net}$



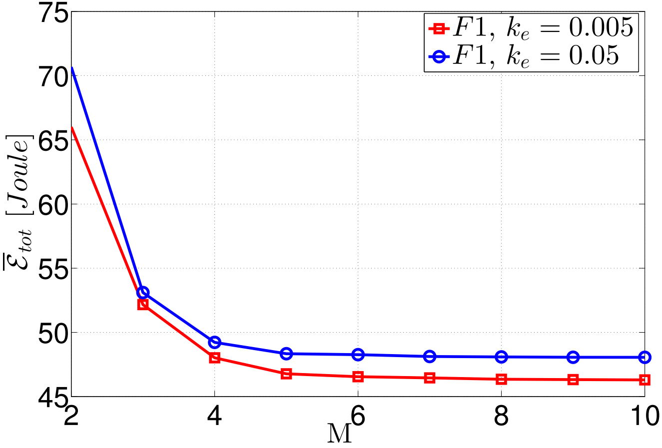
$\overline{\mathcal{E}}^{net}$-vs.-$M$

- $\uparrow M \propto \overline{\mathcal{E}}^{net} \downarrow$
- **The proposed scheduler is about 10%, 50%, 65% better than NetDC, Lyapunov, and Standard schedulers, respectively**



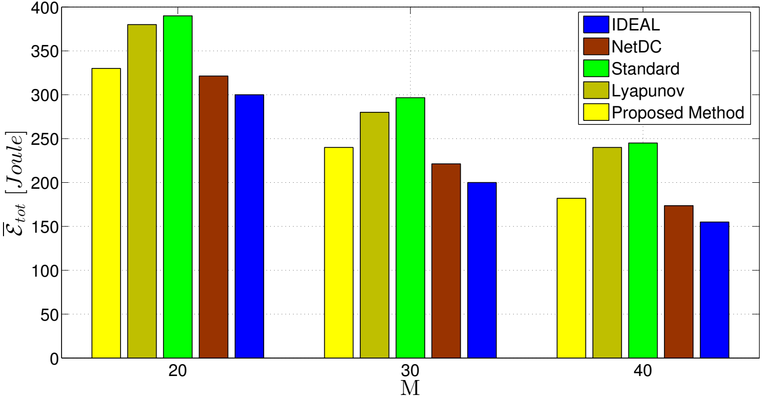
$\overline{\mathcal{E}}_{tot}$-vs.-$M$

- $\uparrow M \propto \overline{\mathcal{E}}_{tot} \downarrow$
- $\uparrow k_e \propto \overline{\mathcal{E}}_{Reconf} \uparrow \propto \overline{\mathcal{E}}_{tot} \uparrow$



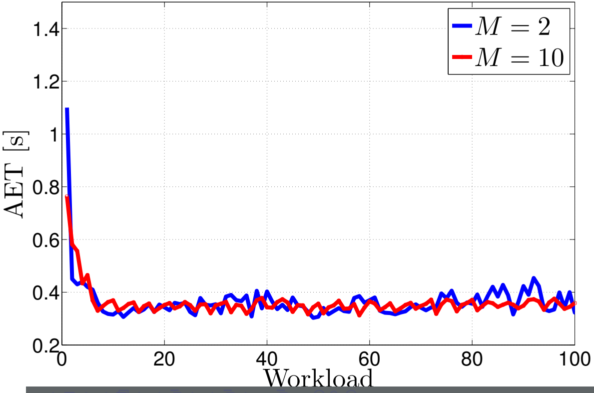
$\overline{\mathcal{E}}_{tot}$-vs.-$M$-Second Scenario

- $\uparrow M \propto \overline{\mathcal{E}}_{tot} \downarrow$
- **The energy reduction of proposed method compared to Standard and Lyapunov is about 20% and 15%,respectively**



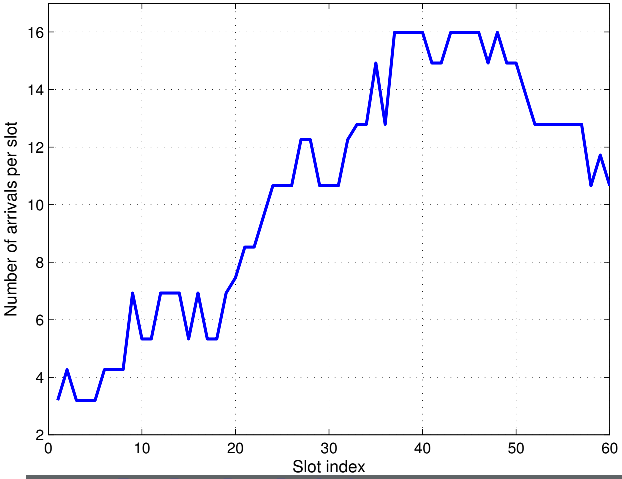
Average execution time (AET) per-job

- *Workload* $\uparrow \propto$ AET $\downarrow$ per-job: proposed scheduler being able to *adapt* itself to the incoming traffic using optimization technique (see (7.1)), with a consequent reduction in the AET per job
- $M \uparrow \propto$ AET $\downarrow$



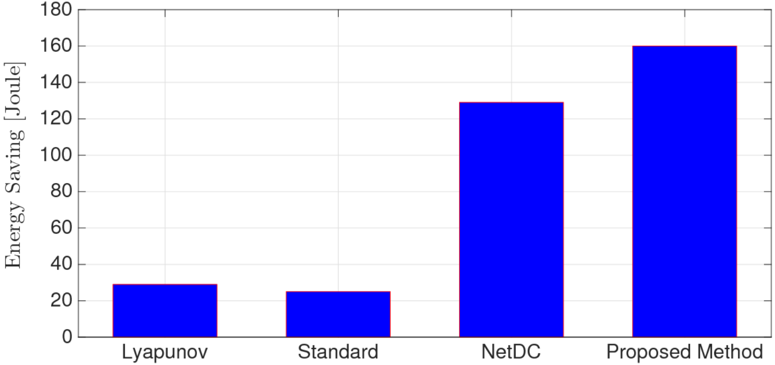
Third Scenario- Real traces

- *Real-world* workload trace (Urgaonkar et al., 2007)



Third Scenario- Real traces

- **Average energy reduction of the proposed scheduler with NetDC, Lyapunov and Standard is 19%, 85%, and 82%, respectively.**



Performance Evaluation-achievements

According to the simulations we understand:

+ The scheduler is a scalable and adaptive. It can save energy and meet QoS demands better than alternatives
+ Our scheduler outperforms Lyapunov, because Lyapunov is unable to manage the online/instantaneous job fluctuations which is handled in our approach
+ Our scheduler outperforms NetDC and IDEAL no-DVFS techniques, because these methods work with the continue ranges of frequencies, which is unrealistic and not feasible in real scenarios
- Our method needs some estimations for applying in the real system (open issue)

Conclusion

1. We propose a novel scheduler to:
   - Minimize the overall energy for the computing-plus-communication resources in VNetDCs
   - Guaranteeing the time limit of each task, bandwidth limitation of each server by changing the reconfiguration capability
2. Our proposed scheduler manages online workloads, and inter-switching costs among active discrete frequencies for each VM
3. Our method is able to approach the IDEAL algorithm significantly faster than Lyapunov, Standard and NetDC models, respectively
4. **Future research:** The energy saving using workload estimating and management of WAN TCP/IP mobile connections

Thanks for the attention and ready for the questions!!!

Performance Evaluation-Scenario 2

Total average consumed energy for 20, 30, and 40 VMs and high volume of incoming jobs with respect to R_t (maximum network data transfer rate) and the communication coefficient ζ in order to evaluate the energy consumption of the proposed method while facing various SLA ranges:

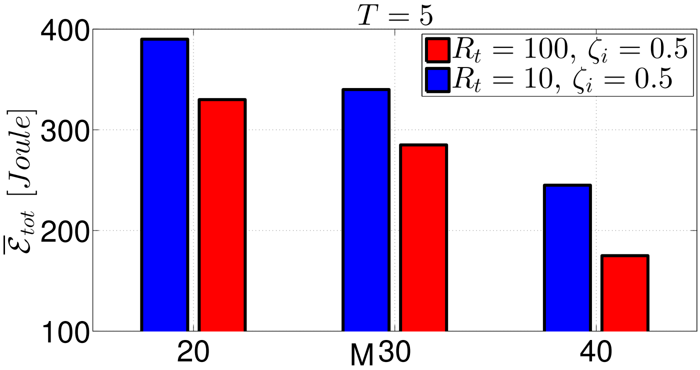
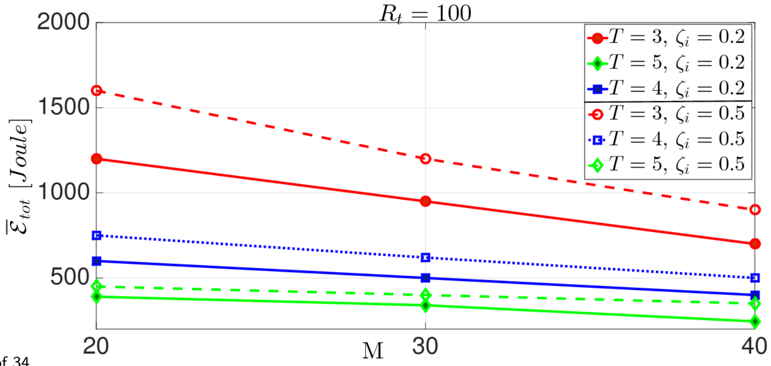


Figure: $\overline{\mathcal{E}}_{tot}$-vs.-$M$-vs.-$R_t$

$\overline{\mathcal{E}}_{tot}$-vs.-$M$-Second Scenario

- $\uparrow M \propto \overline{\mathcal{E}}_{tot} \downarrow$
- $\uparrow T \propto (\overline{\mathcal{E}}_{CPU}, \overline{\mathcal{E}}_{tot}) \downarrow$
- $\uparrow \zeta \propto (\overline{\mathcal{E}}^{net}, \overline{\mathcal{E}}_{tot}) \uparrow$
- **The scheduler can save energy depending on the assigned communication boundary**



Problem Solution-detail

Proof: Let $R(i)^*$ be the optimal solution of the eq. (7.1), and let

$$\mathcal{C} \triangleq \left(\overrightarrow{F_j(i)t^j(i)}\right) \in (\mathbb{R}_0^+)^M : \left(\sum_{j=0}^{Q} F_j(i)t^j(i)/R(i)^* \left(\overrightarrow{F_j(i)t^j(i)}\right)\right) \leq (\overline{T} - T)/2, i = \{1, \ldots, M\}, j = \{0, \ldots, Q\};$$

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} R(i)^* \left(\overrightarrow{F_j(i)t^j(i)}\right) \leq R_t$$

$$\sum_{j=0}^{Q} \frac{2F_j(i)t^j(i)}{R(i)} \leq \overline{T} - T \rightarrow \left(\sum_{j=0}^{Q} \frac{F_j(i)t^j(i)}{R(i)}\right) \leq \frac{(\overline{T} - T)}{2}. \qquad (11)$$

$$\sum_{j=0}^{Q} \frac{2F_j(i)t^j(i)}{R(i)} \leq \overline{T} - T \rightarrow R(i) \geq \sum_{j=0}^{Q} \left(\frac{2F_j(i)t^j(i)}{\overline{T} - T}\right). \qquad (12)$$

Why Shanon for channel model?

i) The theoretical relation of the transmission rate $R(i)$ and power of the channel for each server is more critical, so, we use one of the most complex relations to evaluate
ii) We already used easier model (linear or quadratic model) and the results are more appealing
iii) This model uses for the inside of data center on a physical wired connections