Gestion Efficace des Ressources dans les Plates-formes Hétérogènes

Efficient Management of Resources in Heterogeneous Platforms

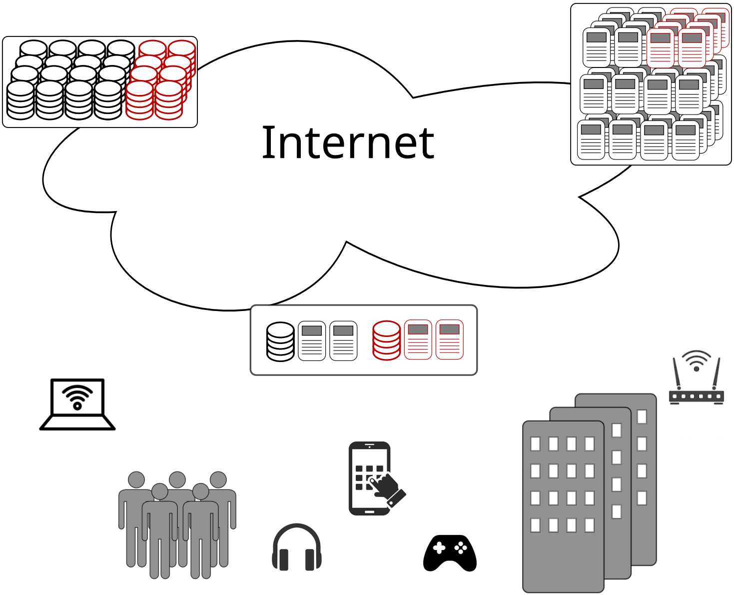
Clément Mommessin

Supervisors: Denis Trystram and Giorgio Lucarelli

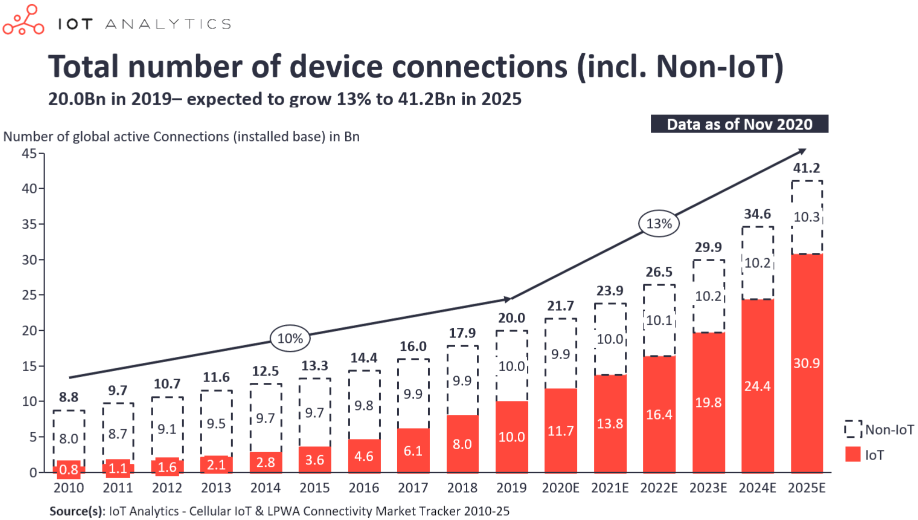
Univ. Grenoble Alpes

Dec. 11th, 2020

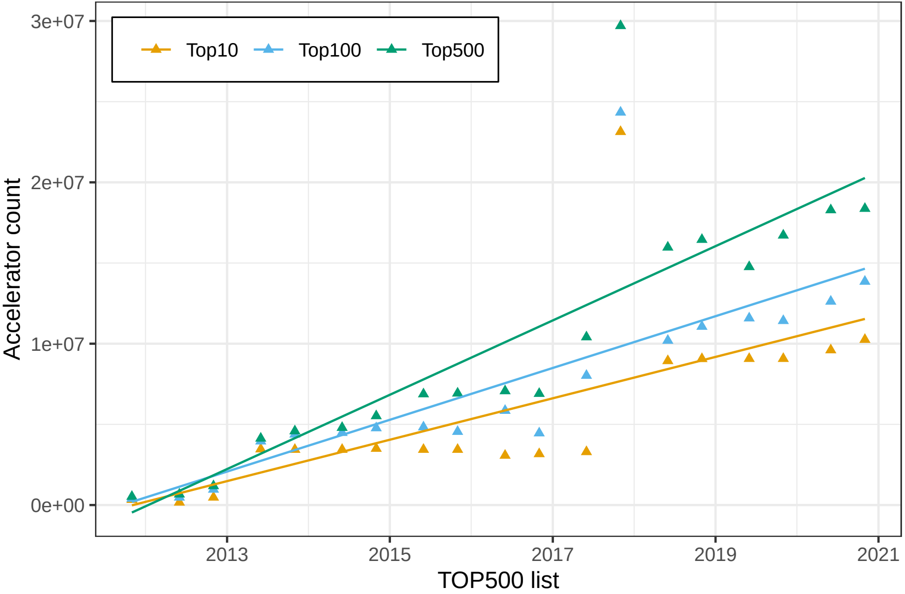
IT World Overview



Evolution of Device Connections



Accelerator Count in HPC Platforms



Thesis Statement

Multiple ever growing numbers

- Digital and connected devices
- Users, computing requests, data generated
- (Heterogeneous) resources

Emergence of optimisation challenges

- Need for better resource management systems

Contributions Overview

→ Focus on optimisation problems for distributed and parallel platforms with heterogeneous resources

High Performance Computing (HPC)

Scheduling on two types of resources

- Theoretical analysis
- Performance evaluation

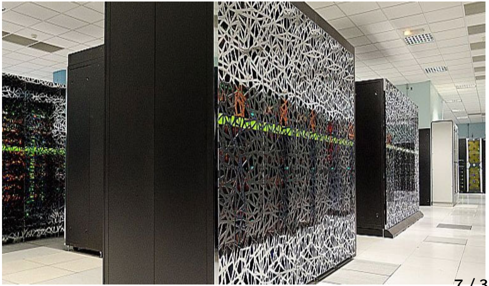
Edge Computing

Qarnot Computing: a case study

- Simulator extensions
- Platform simulation
- Temperature prediction method
- Scheduling problem formulation

From Boxes and Trucks...

...To Tasks and Machines



Scheduling on Two Types of Resources

A Scheduling Problem

Scheduling parallel applications on hybrid multi-core machines

- A parallel machine with 2 types of processors
  - m identical CPUs
  - $k \leq m$ identical GPUs
- An application composed of n sequential tasks
  - Known processing times on CPU ($\overline{p_j}$) and on GPU ($\underline{p_j}$)
  - Known at time 0 (off-line setting)
  - **Precedence relations expressed as a Directed Acyclic Graph**

$\Rightarrow$ Objective: minimise the maximum completion time $C_{\max}$ (known as **makespan**)

Informal Definition of Scheduling

Two questions

The problem is to answer the two following questions for each task of the application:

- **Where?** – Determine which resource will execute the task
- **When?** – Determine the execution interval of the task

$\rightarrow$ For hybrid machines **the 'where?' is crucial** – a wrong decision may be very costly

$\Rightarrow$ We are interested in designing **generic scheduling algorithms** with **performance garantees** in the worst case.

Worst-case Analysis

“How much a solution can be away from the optimal?”

Definition: Approximation ratio (for min. problems)

$$\max_{I \in \text{problem instances}} \frac{\text{algorithm solution for instance I}}{\text{optimal solution for instance I}}$$

Scheduling Algorithms

HEFT: Heterogeneous Earliest Finish Time [THW99]

1. **Tasks prioritisation**: Ranking from precedence constraints and processing times
2. **Tasks scheduling**: Earliest Finish Time policy

HLP: Heterogeneous Linear Program [KSMT15]

1. **Allocation**: Relaxed linear program + rounding technique
2. **Scheduling**: Earliest Starting Time policy (= List Scheduling [Gra69])

HLP with Ordered List Scheduling (OLS) policy

1. **Allocation**: Same as HLP-EST (linear program + rounding)
2. **Scheduling**: HEFT ranking + List Scheduling (**OLS policy**)

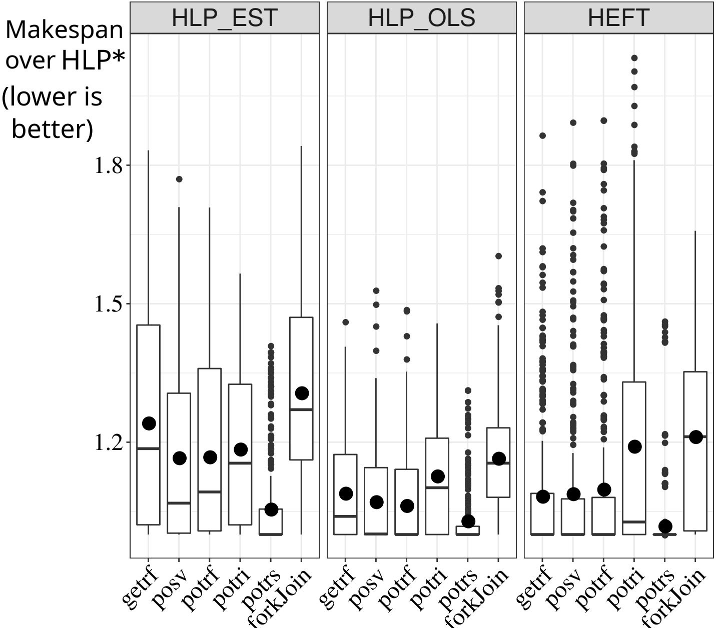
Theoretical Results

- **HEFT**: Approximation ratio at least $\frac{m+k}{k^2}(1-\frac{1}{e^k})$, with $k^2 \leq m$ (no constant performance guarantee)
- **HLP-EST**: Approximation ratio at least $6 - O(\frac{1}{m})$ (Approximation ratio at most 6 [KSMT15])
- **HLP-OLS**: Same tight approximation ratio as HLP-EST

Extensions to $q \geq 2$ types of resources

- Linear program HLP extended to qHLP (+ rounding)
- Algorithms HLP-EST and HLP-OLS extended
- (Tight) approximation ratio of $q(q+1)$.

Experiments: 2 Resource Types Off-line



A More Complicated Problem

On-line setting

- Tasks arrive in any order respecting precedence constraints
- Processing times are only known when the task arrives

$\rightarrow$ An algorithm must take an irrevocable scheduling decision upon arrival of a task

Definition: Competitive ratio (for min. problems)

$$\max_{I \in \text{problem instances}} \frac{\text{algorithm solution for instance I}}{\text{optimal } \textbf{off-line} \text{ solution for instance I}}$$

On-line Algorithm

ER-LS (Enhanced Rules - List Scheduling)

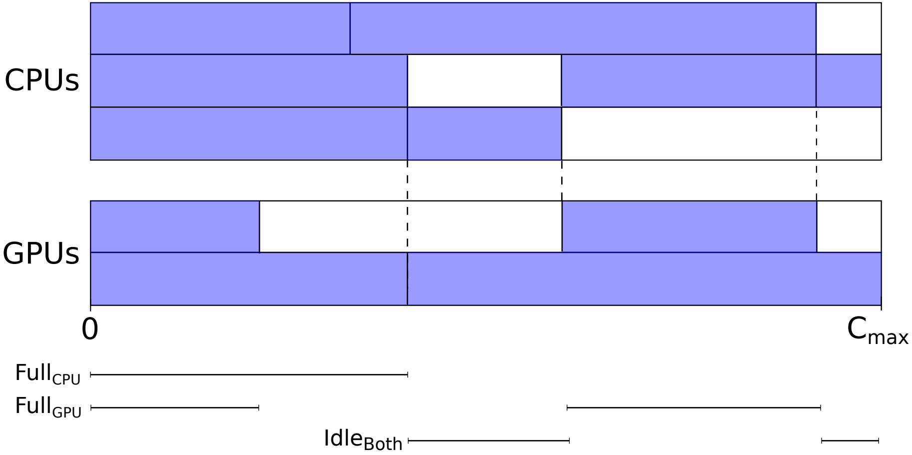
1. **Allocation**:
   - **Rule 1**: **If** $\overline{p_j} > \underline{\tau'} + \underline{p_j}$ **then** $T_j \rightarrow$ GPU
     ($\underline{\tau'}$: first time a GPU can start T_j)
   - **Rule 2: If** $\overline{p_j}/\sqrt{m} \leq \underline{p_j}/\sqrt{k}$ **then** $T_j \rightarrow$ CPU **else** $T_j \rightarrow$ GPU
2. **Scheduling**: List Scheduling (EST policy)

$\rightarrow$ First on-line scheduling algorithm on hybrid machines to take into account precedence constraints

$\Rightarrow$ The competitive ratio of ER-LS is at least $\sqrt{m/k}$ and at most $4\sqrt{m/k}$

Sketch of Proof (1)

Partition the schedule into 3 interval subsets



Sketch of Proof (2)

Bound the value of $C_{\max}$:

$$\begin{aligned} C_{\max} &\leq |Full_{\text{CPU}}| + |Full_{\text{GPU}}| + |Idle_{\text{Both}}| \\ &\leq \frac{Load_{\text{CPU}}}{m} + \frac{Load_{\text{GPU}}}{k} + |CritPath| \end{aligned}$$

Idea: Compare the allocation (CPU-GPU) of tasks in the optimal schedule with the allocation given by the algorithm

$$\frac{Load_{\text{CPU}}}{m} + \frac{Load_{\text{GPU}}}{k} \leq 3\sqrt{\frac{m}{k}} C_{\max}^{\text{OPT}}$$

$$|CritPath| \leq \sqrt{\frac{m}{k}} C_{\max}^{\text{OPT}}$$

$$\Rightarrow C_{\max} \leq 4\sqrt{\frac{m}{k}} C_{\max}^{\text{OPT}}$$

Related Work: On-line

For independent tasks [CYZ14]

- The competitive ratio of any algorithm is at least 2
- The competitive ratio of Al_5 is at most 3.85

For tasks with precedences [CMSV19]

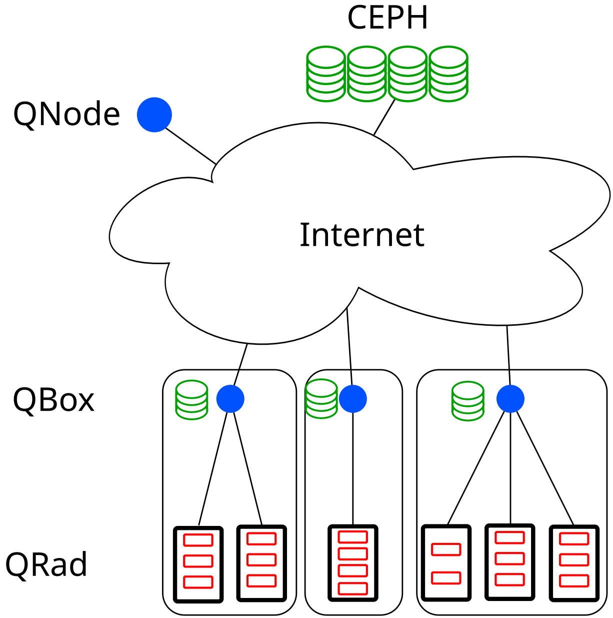
- The competitive ratio of any algorithm is at least $\sqrt{m/k}$
- The competitive ratio of QA is at most $2\sqrt{m/k} + 1$

Edge Computing: A Case Study

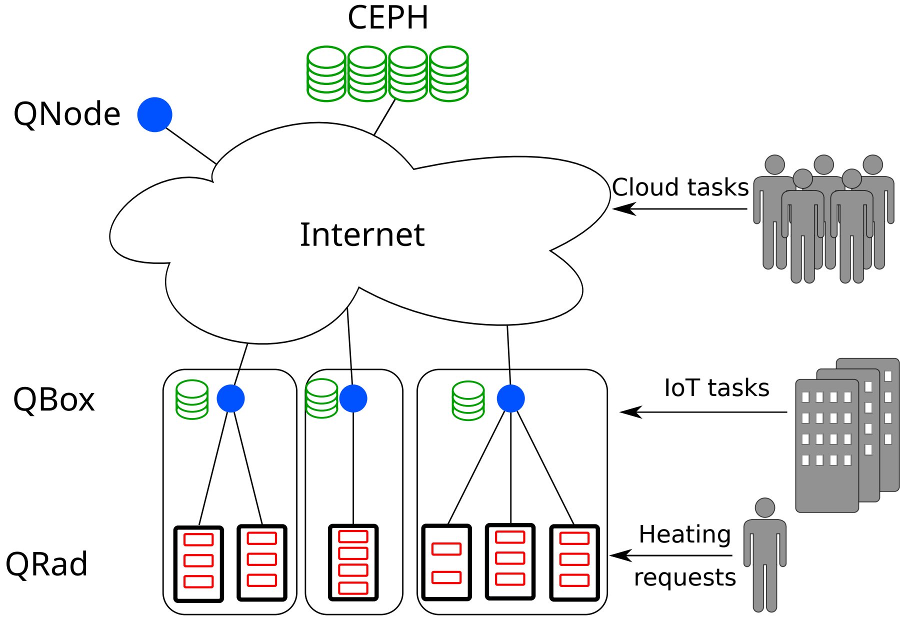
Qarnot Computing

"A disruptive solution to turn IT waste heat into a viable heating solution for buildings."

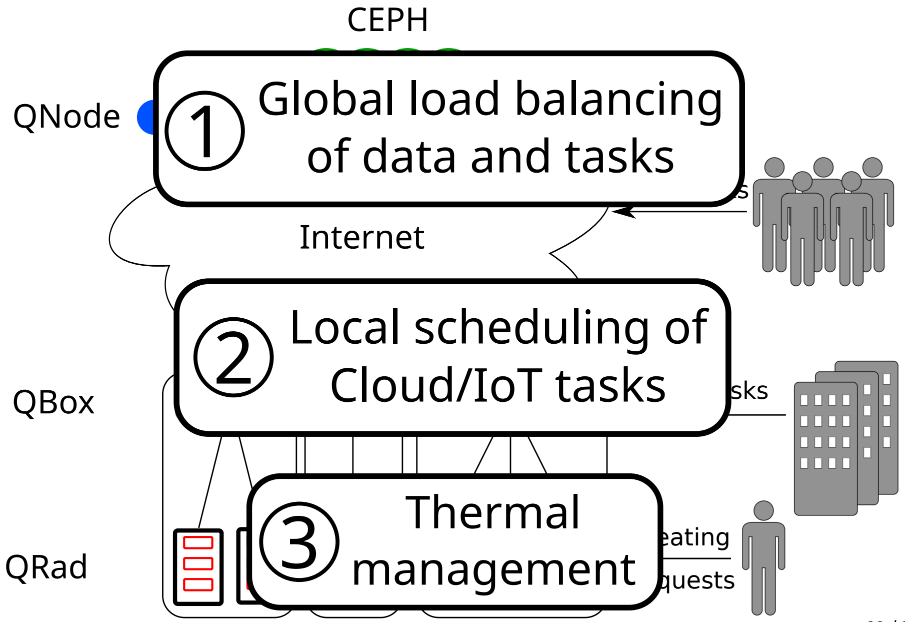
The Qarnot Platform



The Qarnot Platform with Users



Several Resource Management Problems



Case Study of Qarnot Computing

Main goals

1. Study the different resource management problems
2. Propose solutions
3. Test and validate them

Problem

Testing on a real platform is not conceivable

- Costly and time consuming
- Tested solutions may be difficult to (quickly) deploy
- Users will not be happy

$\rightarrow$ This is the production platform!

$\Rightarrow$ **The solution is the simulation**

Simulation Rules!

Testing through simulation

- Fast, deterministic, in a controlled environment
- Easy to switch between solutions
- Easy to test complicated/unfeasible scenarios in production

$\rightarrow$ We can test whatever we want!

Batsim and SimGrid

Simulation tools for HPC platforms

- **SimGrid**: Large-scale parallel/distributed system simulator
- **Batsim**: Infrastructure simulator for job and I/O scheduling

New extensions for Edge Computing platforms

- **External events injector:** Replay machine failures, temperature changes, etc.
- **Storage controller:** Manage storage entities and data movements

$\rightarrow$ **Merged in Batsim and PyBatsim Git projects**

Simulation Difficulties

Network and data transfer

- Communications via the Internet
- Coarse-grain data transfers

Temperature

- Temperature-driven computing resources availabilities
- Prediction method not validated

$\rightarrow$ 3rd resource management problem of the platform

Proof of Concept

Main goals

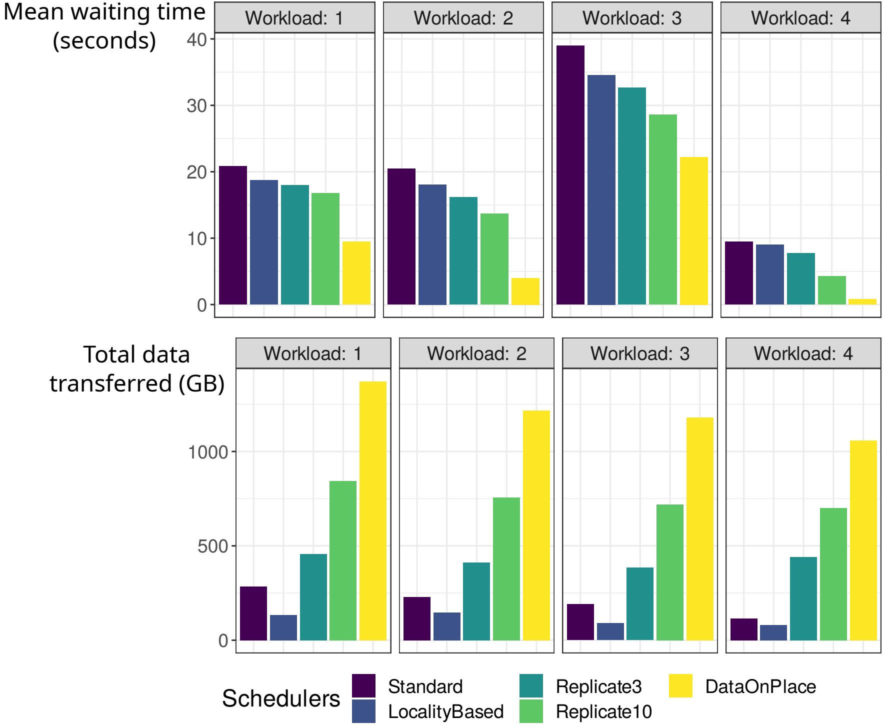
- Demonstrate the simulation extensions for Edge Computing
- Test different job and data placement strategies at QNode-level

Job and data placement strategies

- **Standard**: Basic Qarnot scheduler
- **LocalityBased**: Favours re-use of data-sets
- **Replicate3**: Replicates data-sets on 3 QBoxes
- **Replicate10**: Replicates data-sets on 10 QBoxes
- **DataOnPlace**: Assumes instantaneous data transfers

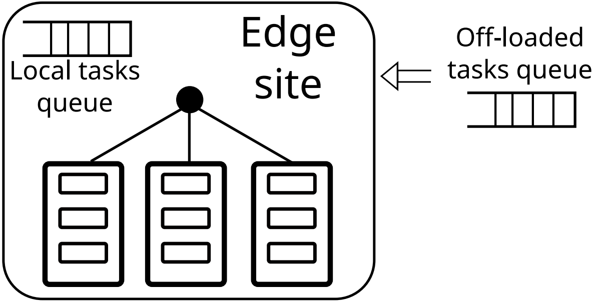
Simulations of 1-week workloads from Qarnot logs

$\rightarrow$ 1st resource management problem of the platform



Mean waiting time (seconds)
Workload: 1
Workload: 2
Workload: 3
Workload: 4
40
30
20
10
0
Total data transferred (GB)
1000
500
0
Schedulers
Standard
LocalityBased
Replicate3
Replicate10
DataOnPlace

Local Scheduling



Basic settings

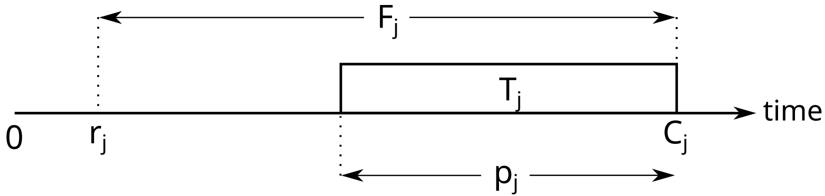
- Multiple machines
- A queue of local tasks
- A queue of off-loaded tasks
- Bi-objective

→ 2nd resource management problem of the platform

Two-agent Scheduling

Problem formulation

- Identical parallel machines
- Local agent:
  - On-line non-preemptive sequential tasks with release dates (r_j)
  - Processing times known when released (p_j)
  - Objective: minimise sum flow-time (F_j)
- Global agent:
  - Off-line non-preemptive sequential tasks
  - Known processing times
  - Objective: minimise maximum completion time (C_{max})

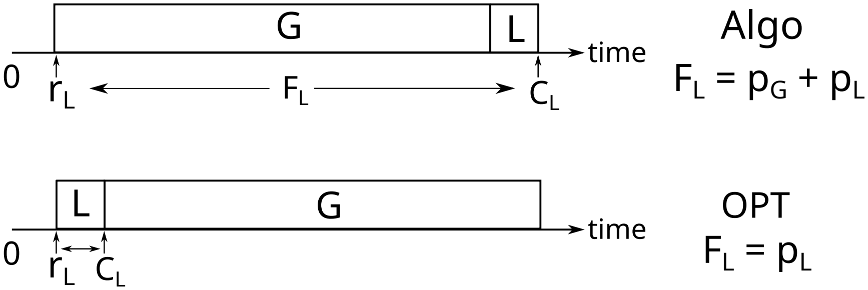


Strong Competitive Ratio Lower Bound

Worst-case example

- One machine
- One long global task G
- One short local task L

$\rightarrow$ Release L right after G has started



Current Research

Task rejection

- Cope with strong lower bounds
- Give more power to the algorithm

Allow rejection of global tasks

$\rightarrow$ Trade-off between number of rejections and quality of the solution

Primal-Dual approach

1. **Formulate Primal/Dual linear programs**
2. **Interpret Dual variables**
3. Design/analyse algorithm with performance guarantees

Concluding Words

$\Rightarrow$ **Achieving an efficient management of resources in heterogeneous platforms is not that easy**

High Performance Computing

Scheduling on hybrid machines

- Theoretical analysis of scheduling algorithms
- Performance evaluation

Edge Computing

Case-study of Qarnot Computing

- Platform simulation
- Study of different resource management problems in theory and practice

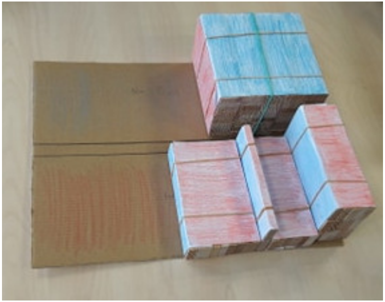
What's Next?

- Find good data to validate the temperature prediction method
- Combine temperature prediction with scheduling

- Add more features to Batsim/SimGrid

- Continue the work on the 2-agent scheduling problem

Computer Science Unplugged



List of Publications

International journals

- Beaumont et al., *"Scheduling on Two Types of Resources: A Survey"*. In ACM Computing Surveys (May 2020).
- Amaris et al., *"Generic Algorithms for Scheduling Applications on Heterogeneous Platforms"*. In CCPE (July 2018).

International conferences with procedings

- Bauskar et al., *"Investigating Placement Challenges in Edge Infrastructures through a Common Simulator"*. In SBAC-PAD 2020.
- Amaris et al., *"Generic Algorithms for Scheduling Applications on Hybrid Multi-core Machines"*. In Europar 2017.
- Mommessin et al., *"Automatic Data Filtering in In Situ Workflows"*. In Cluster 2017.

Gestion Efficace des Ressources dans les Plates-formes Hétérogènes

Efficient Management of Resources in Heterogeneous Platforms

Clément Mommessin

Supervisors: Denis Trystram and Giorgio Lucarelli

Univ. Grenoble Alpes

Dec. 11th, 2020

Scheduling on Hybrid Platforms: State of Art

Setting		Off-line	On-line
Independent tasks	Lower bound	-	2
	Best known	$1+\epsilon$	3.85
Tasks with precedences	Lower bound	3	$\sqrt{\frac{m}{k}}$
	Best known	$3+2\sqrt{2}$	$2\sqrt{\frac{m}{k}}+1$

HLP

minimise λ subject to:

$$C_i + \overline{p_j} x_j + \underline{p_j}(1 - x_j) \leq C_j \qquad \forall T_j \in \mathcal{T}, T_i \in \Gamma^-(T_j) \tag{1}$$

$$\overline{p_j} x_j + \underline{p_j}(1 - x_j) \leq C_j \qquad \forall T_j \in \mathcal{T} : \Gamma^-(T_j) = \emptyset \tag{2}$$

$$C_j \leq \lambda \qquad \forall T_j \in \mathcal{T} \tag{3}$$

$$\frac{1}{m} \sum_{T_j \in \mathcal{T}} \overline{p_j} x_j \leq \lambda \tag{4}$$

$$\frac{1}{k} \sum_{T_j \in \mathcal{T}} \underline{p_j}(1 - x_j) \leq \lambda \tag{5}$$

$$x_j \in \{0, 1\} \qquad \forall T_j \in \mathcal{T} \tag{6}$$

$$C_j \geq 0 \qquad \forall T_j \in \mathcal{T} \tag{7}$$

Ranking Methods

Ranking of HEFT (unrelated resources)

$$Rank(j) = \tilde{p_j} + \max_{i \in Succ(j)} \{\tilde{Comm}_{j,i} + Rank(i)\}$$

Ranking of HLP-OLS (CPU/GPU)

$$Rank(j) = \overline{p_j} x_j + \underline{p_j}(1 - x_j) + \max_{i \in Succ(j)} \{Rank(i)\}$$

qHLP

minimise λ subject to:

$$C_i + \sum_{q=1}^{Q} p_{j,q} x_{j,q} \leq C_j \qquad \forall T_j \in \mathcal{T}, T_i \in \Gamma^-(T_j) \tag{8}$$

$$\sum_{q=1}^{Q} p_{j,q} x_{j,q} \leq C_j \qquad \forall T_j \in \mathcal{T} : \Gamma^-(T_j) = \emptyset \tag{9}$$

$$C_j \leq \lambda \qquad \forall T_j \in \mathcal{T} \tag{10}$$

$$\frac{1}{m_q} \sum_{T_j \in \mathcal{T}} p_{j,q} x_{j,q} \leq \lambda \qquad 1 \leq q \leq Q \tag{11}$$

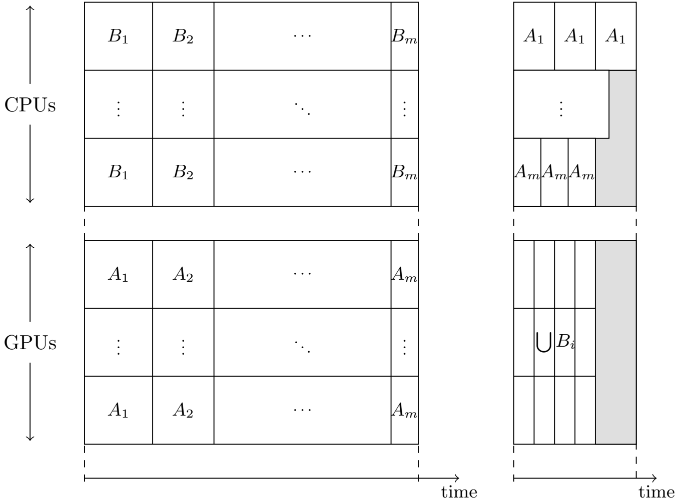
$$\sum_{q=1}^{Q} x_{j,q} = 1 \qquad \forall T_j \in \mathcal{T} \tag{12}$$

$$x_{j,q} \in \{0,1\} \qquad \forall T_j \in \mathcal{T},\ 1 \leq q \leq Q \tag{13}$$

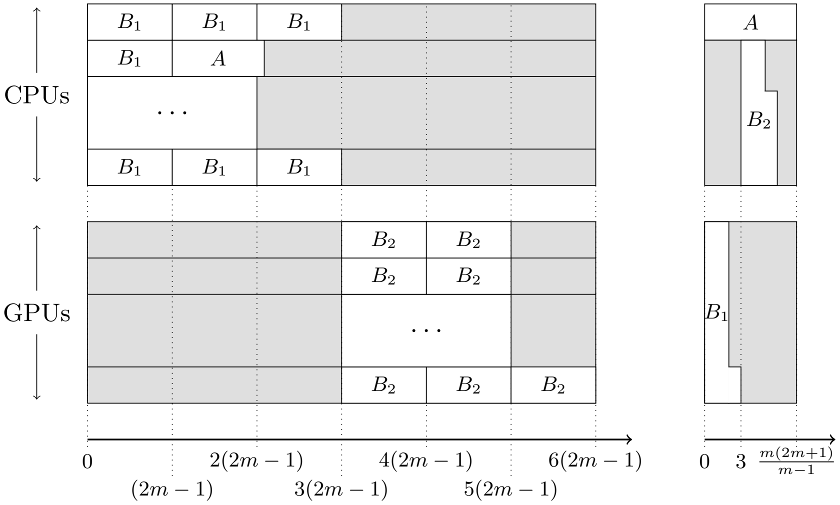
$$C_j \geq 0 \qquad \forall T_j \in \mathcal{T} \tag{14}$$

$$\lambda \geq 0 \tag{15}$$

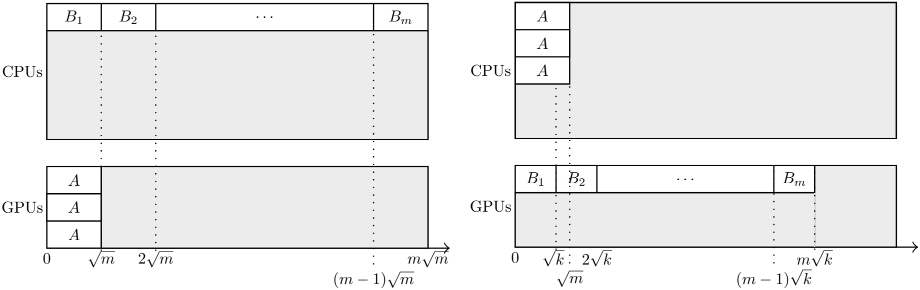
HEFT Lower Bound



HLP-EST/HLP-OLS Lower Bound



ER-LS Lower Bound



Performance Evaluation

Benchmark creation (by M. Amaris)

- 18 instances of 5 linear algebra applications for dense matrices (Chameleon) from real traces
- 15 instances of a fork-join application generated "by hand"

Each instance generated with varying size of task graphs
$\rightarrow$ Between 50 and 5,000 tasks per instance

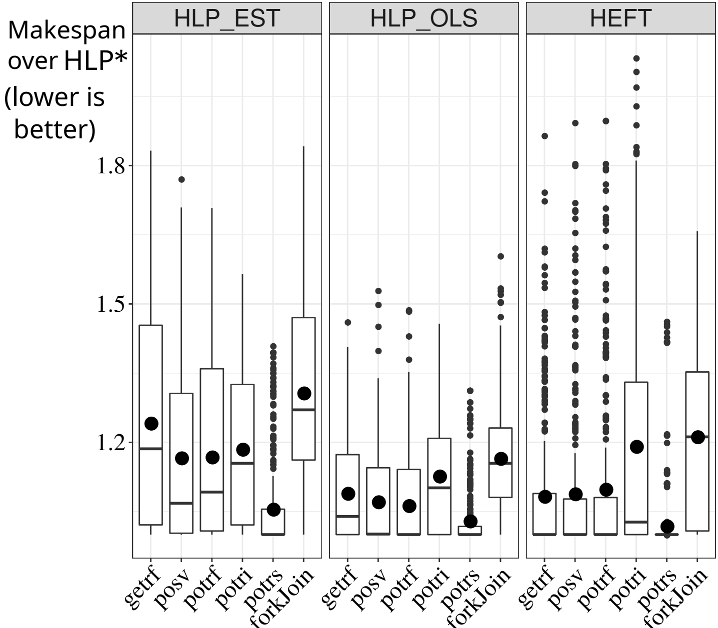
Simulation execution

- 16 machine settings with various numbers of CPUs/GPUs
- Each application instance simulated on each machine setting

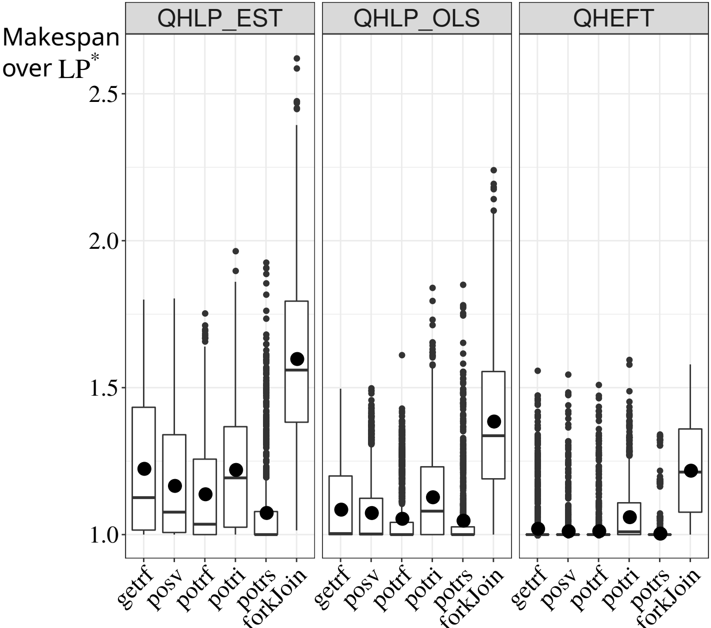
$\Rightarrow$ 288 runs for each Chameleon applications, 240 for fork-join, for each scheduling algorithm

$\rightarrow$ Comparison of $\frac{\text{makespan}}{\text{LP*}} \leq \frac{\text{makespan}}{\text{OPT}}$ (achieved approx. ratio)

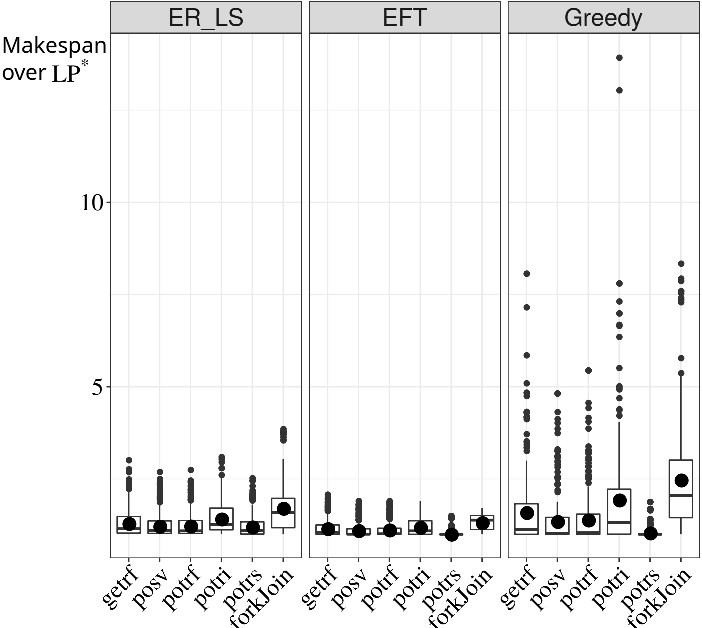
Experiments: 2 Resource Types Off-line



Experiments: 3 Resource Types Off-line



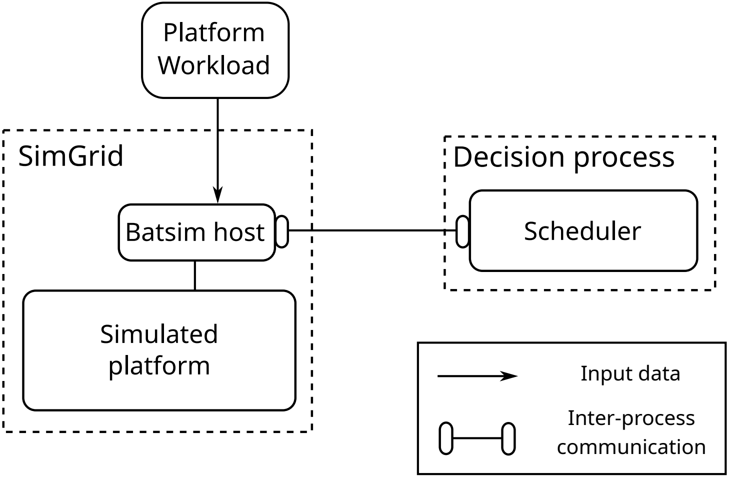
Experiments: 2 Resource Types On-line



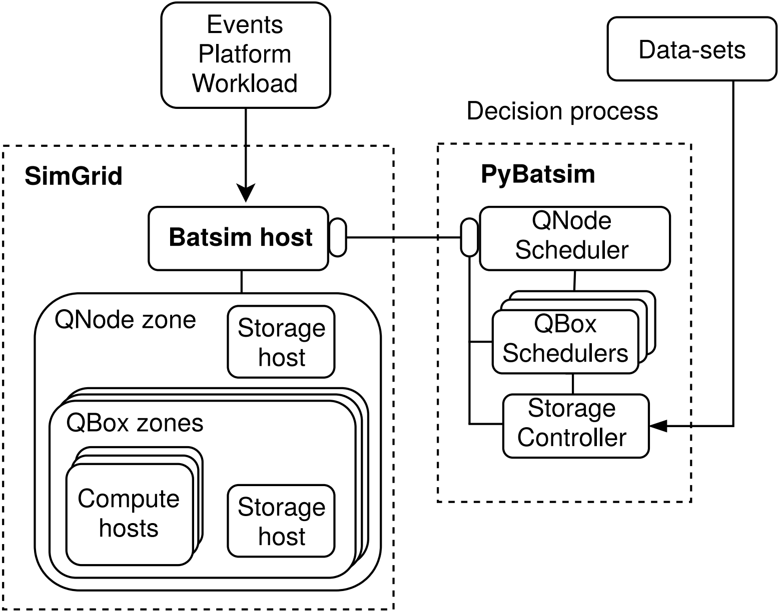
Simulation Tools

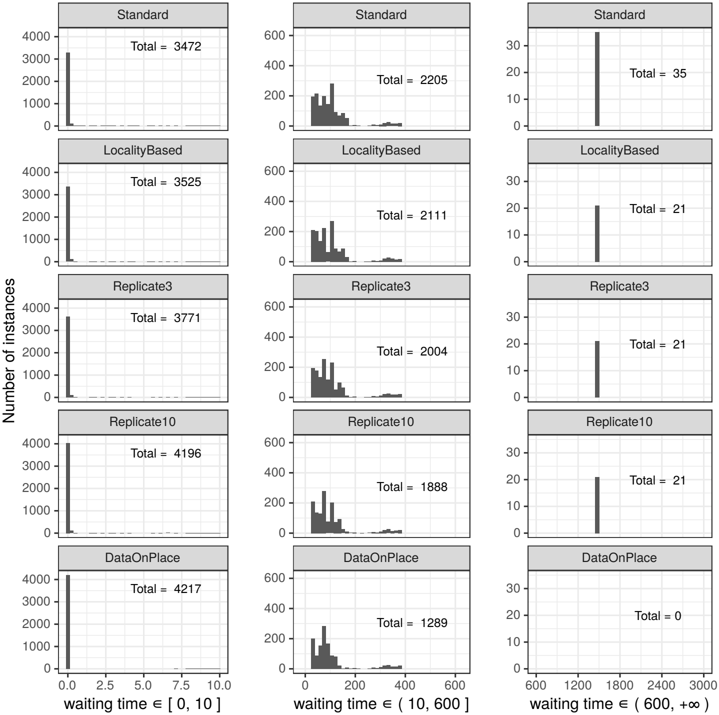
- **SimGrid**: `https://github.com/simgrid/simgrid`
- **Batsim**: `https://github.com/oar-team/batsim`
- **SimGrid Temperature for Qarnot**: `https://github.com/Mommessc/simgrid/tree/temperature-sbac-2020`
- **Batsim for Qarnot**: `https://gitlab.inria.fr/batsim/batsim/tree/temperature-sbac-2020`
- **PyBatsim for Qarnot**: `https://gitlab.inria.fr/batsim/pybatsim/tree/temperature-sbac-2020`

Batsim/SimGrid



Simulated qarnot Platform




Standard
Total = 3472
Total = 2205
Total = 35
LocalityBased
Total = 3525
Total = 2111
Total = 21
Replicate3
Total = 3771
Total = 2004
Total = 21
Replicate10
Total = 4196
Total = 1888
Total = 21
DataOnPlace
Total = 4217
Total = 1289
Total = 0
Number of instances
waiting time ∈ [0, 10]
waiting time ∈ (10, 600]
waiting time ∈ (600, +∞)

Thermodynamic Formulae

Thermal energy/heat capacity:

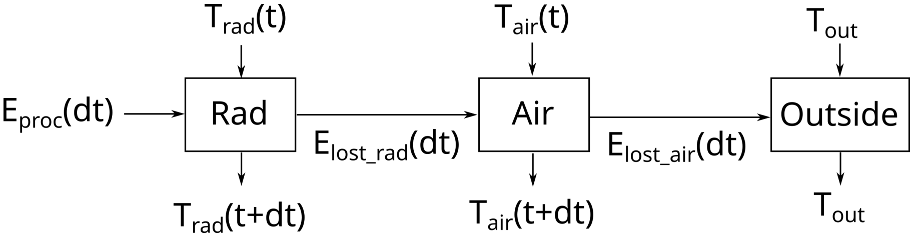
$$Q = C \times \Delta T$$

Conductive heat transfer:

$$\frac{Q}{dt} = \frac{\Delta T}{R}$$

Energy quantity Q $[J]$, thermal capacity $C = mc$ $[J.K^{-1}]$, thermal resistance R $[K.W^{-1}]$, temperature difference ΔT $[K]$, time period dt $[s]$.

Temperature Transfers



Thermal Models

Naive Iterative Approach:

$$T_{\text{rad}}(t+1) = T_{\text{rad}}(t) + \frac{E_{\text{gained\_rad}} - E_{\text{lost\_rad}}}{C_{\text{rad}}}$$

$$T_{\text{air}}(t+1) = T_{\text{air}}(t) + \frac{E_{\text{lost\_rad}} - E_{\text{lost\_air}}}{C_{\text{air}}}$$

Closed-form:

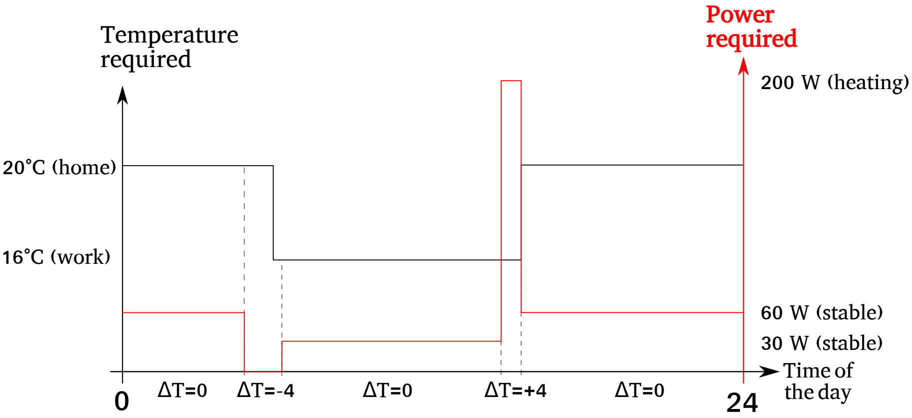
$$\begin{pmatrix} T_{\text{rad}}(n) \\ T_{\text{air}}(n) \end{pmatrix} = A^n \cdot \begin{pmatrix} T_{\text{rad}}(0) \\ T_{\text{air}}(0) \end{pmatrix} + S_n \cdot \begin{pmatrix} \frac{P_{\text{rad}}}{C_{\text{rad}}} \\ \frac{T_{\text{out}}}{R_{\text{air}} C_{\text{air}}} \end{pmatrix}$$

Lumped Thermal Model:

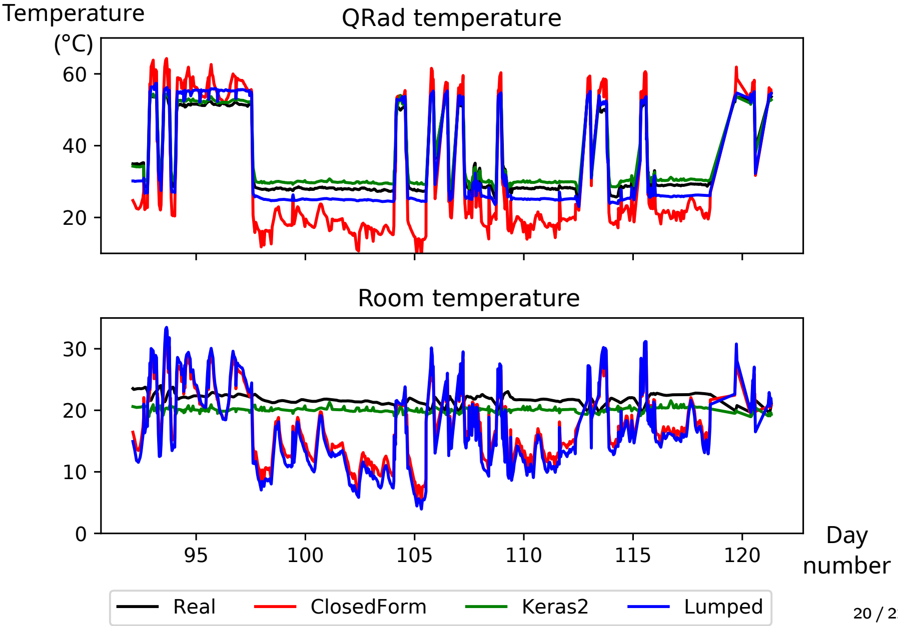
$$T_{\text{rad}}(t) = T_{\text{rad}}(0) \cdot e^{-\alpha t} + (T_{\text{air}} + P_{\text{rad}} \cdot R_{\text{rad}}) \cdot (1 - e^{-\alpha t})$$

where $\alpha = \frac{1}{RC}$ $[s^{-1}]$ for the rad.

Temperature Requirements VS Power



Temperature Experiments



Primal Program

$$\min \sum_{j \in \mathcal{L}} \int_{r_j}^{\infty} \left(\frac{(t - r_j)}{p_j} + \frac{1}{2} \right) x_j(t) dt \quad \text{subject to:}$$

$$\begin{aligned}
\int_{r_j}^{\infty} x_j(t) dt \quad & \geq \; p_j && \forall j \in \mathcal{L} \cup \mathcal{G} \\
\sum_{\forall j \in \mathcal{L} \cup \mathcal{G}} x_j(t) \quad & \leq |M| && \forall t \\
\int_{0}^{\infty} \left(\frac{t}{p_j} + \frac{1}{2} \right) x_j(t) dt \quad & \leq C_{max}^{G} && \forall j \in \mathcal{G} \\
x_j(t) + x_j(t') \quad & \leq 1 && \forall j \in \mathcal{G} \cup \mathcal{L}, \forall t, \forall t' \geq t + p_j \\
x_j(t) \quad & \in \{0; 1\} && \forall t, \forall j \in \mathcal{L} \cup \mathcal{G}
\end{aligned}$$

Dual Program

maximize $\sum_j p_j \alpha_j - \int_t M \beta_t dt - \sum_{j \in \mathcal{G}} C^G_{max} - \sum_{j \in LG} \int_0^\infty \int_{t'=t+p_j}^\infty \delta_{j,t,t'} \; dt' dt$

subject to:

$$
\begin{aligned}
\alpha_j - \beta_t - \int_0^\infty \delta_{j,t,t'} dt' + \int_0^\infty \delta_{j,t-p_j-t',t'} dt' &\leq \tfrac{t-r_j}{p_j} + \tfrac{1}{2} && \forall j \in L, \ \forall t \\
\alpha_j - \beta_t - \int_0^\infty \delta_{j,t,t'} dt' + \int_0^\infty \delta_{j,t-p_j-t',t'} dt' &\leq 0 && \forall j \in G, \ \forall t \\
\alpha_j &\geq 0 && \forall j \\
\beta_t &\geq 0 && \forall t \\
\delta_{j,t,t'} &\geq 0 && \forall j, \forall t, \forall t'
\end{aligned}
$$

References I

- Louis-Claude Canon, Loris Marchal, Bertrand Simon, and Frédéric Vivien, *Online scheduling of task graphs on heterogeneous platforms*, IEEE Transactions on Parallel and Distributed Systems (2019).
- Lin Chen, Deshi Ye, and Guochuan Zhang, *Online scheduling of mixed CPU-GPU jobs*, International Journal of Foundations of Computer Science **25** (2014), no. 06, 745–761.
- Ronald. L. Graham, *Bounds on multiprocessing timing anomalies*, SIAM Journal On Applied Mathematics **17** (1969), no. 2, 416–429.
- Safia Kedad-Sidhoum, Florence Monna, and Denis Trystram, *Scheduling tasks with precedence constraints on hybrid multi-core machines*, HCW - IPDPS Workshops, 2015, pp. 27–33.

References II

Haluk Topcuoglu, Salim Hariri, and Min-You Wu, *Task scheduling algorithms for heterogeneous processors*, Heterogeneous Computing Workshop (HCW), 1999, pp. 3–14.