RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

ONERA
THE FRENCH AEROSPACE LAB

cnes

Towards Temporal Isolation for Heterogeneous Space Applications on Linux

Merlin Kooshmanian[1], Frédéric Boniol[1], Jérôme Ermont[2], Simon Corbin[3], Lucas Miné[3]
[1] ONERA, Toulouse, France, [2] IRIT, Toulouse, France, [3] CNES, Toulouse, France

1. Motivation & Problem Statement

Motivation

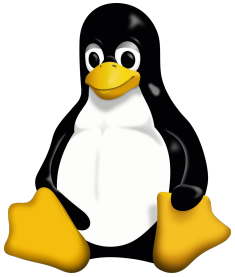
- Increasing onboard **computational complexity**
- **Linux**: rich ecosystem & advanced capabilities
- **Containers**: modularity & resource isolation

Problems

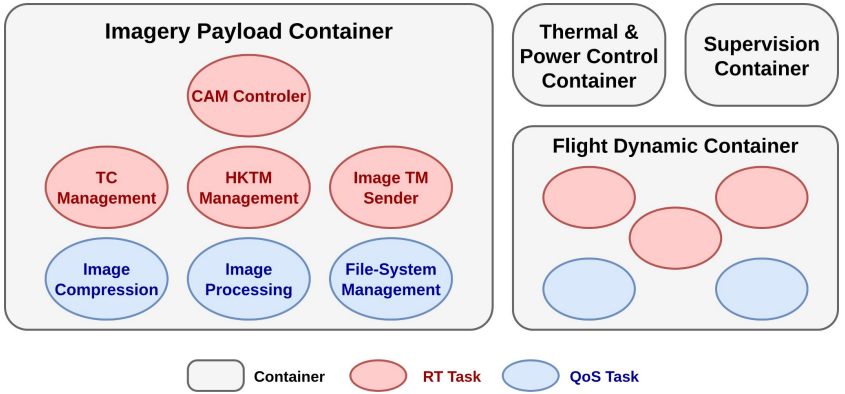
- **Temporal isolation** between containers
- Mixed **deadline-constrained real-time activities** and **progress-oriented background activities**
- **Limited support** for heterogeneous scheduling semantics within the same isolated container

Objective

Provide **predictable** execution of **heterogeneous workloads** inside **temporally isolated** Linux **containers**.



Tux



Motivating Example

2. Application Model

Container Model

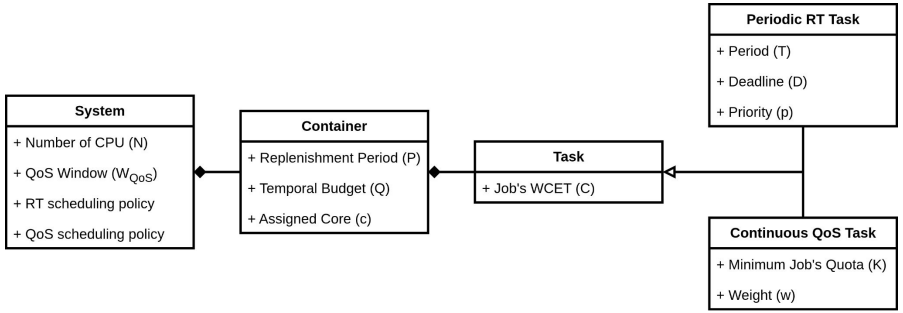
- One subsystem → one Linux container
- Periodic CPU reservation: **budget Q, period P**
- Reserved bandwidth: U = Q/P
- Native Linux scheduling preserved

Real-Time Tasks

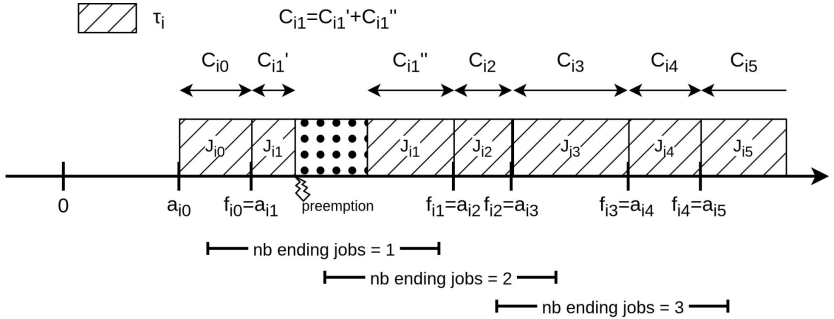
- Periodic / event-triggered activities
- **WCET, period, deadline, priority**
- Constraint: meet every deadline
- Scheduling: **SCHED_FIFO**

QoS Tasks

- **Continuously backlogged** activities
- New job released **after previous completion**
- Progress measured over a **QoS window W_{QoS}**
- Complete at least **K_i jobs per window**
- CPU sharing with **SCHED_OTHER**



Application Model



Continuous QoS Task Chronogram

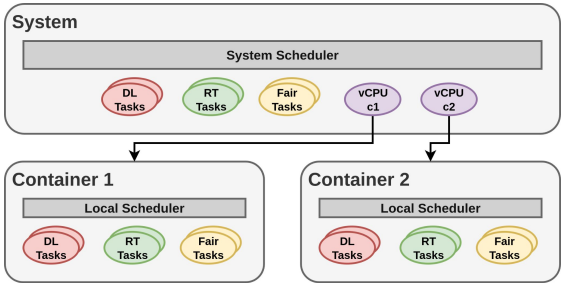
3. Linux CPU Reservation Mechanism

Task Group Bandwidth Server (TGBS)

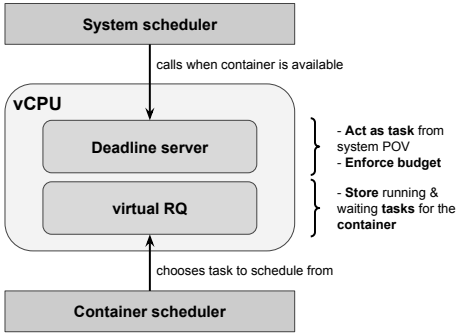
- Linux CPU reservations via **SCHED_DEADLINE** / **CBS**
- **TGBS**: container-level hierarchical reservations
- One container → one **vCPU** / **deadline server**
- vCPU **accounted and throttled** by its reservation
- **Native Linux scheduling classes preserved** inside

Key Idea

Temporal isolation at container level, without losing heterogeneous scheduling semantics



TGBS Overview



TGBS Implementation

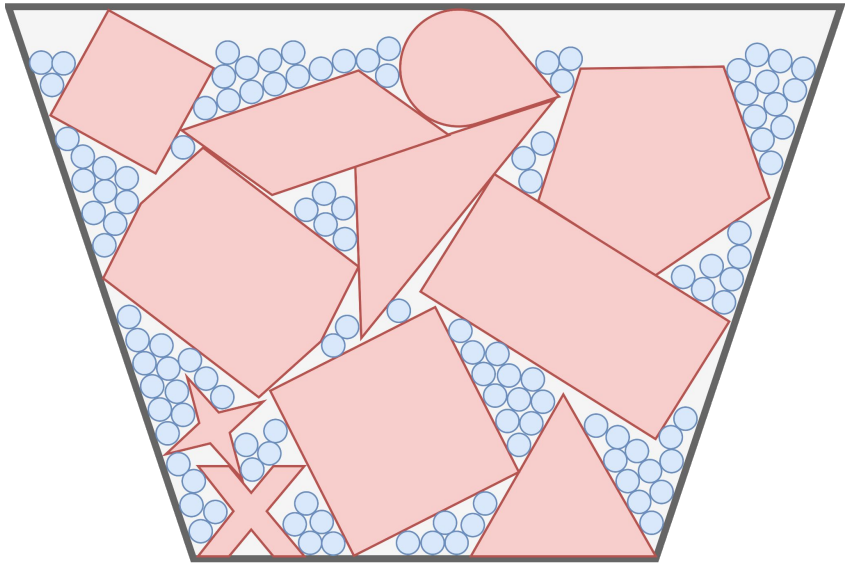
4. Reservation Sizing

Reservation Sizing

- Size reservation for **RT guarantees**
- Worst-case RT sizing leaves **unused capacity**
- Use available **slack for QoS**
- If needed, **increase container budget**

Key Idea

Preserve RT deadlines while ensuring QoS progress



Bucket Analogy

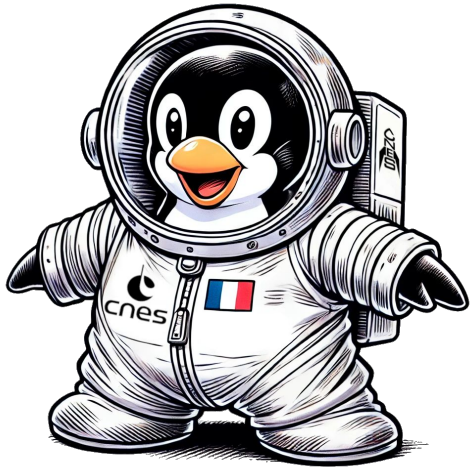
5. Takeways & Future Work

Takeaways

- Bi-class model: RT deadlines + QoS progress
- TGBS: container-level temporal isolation
- Reservation sizing: RT guarantees + QoS bandwidth

Future Work

- Strengthen analytical guarantees
- Assess ideal vs. Linux QoS sharing
- Evaluate TGBS vs. traditional isolation



RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ONERA
THE FRENCH AEROSPACE LAB

Thanks for listening !

contact : merlin.kooshmanian@onera.fr

cnes