Scheduling for Cloud-Based Computing Systems to Support Soft Real-Time Applications

Yuhuan Du, Gustavo de Veciana
(University of Texas at Austin)

Thanks to Huawei for their support and to Alan Gatherer for extensive feedback!
Cloud-Based Real-Time Services

- Streams of tasks generated by producers.
- Producers → Computing Centers → Consumers
- End-to-end deadline for each task.
Some Examples

- Cloud-based Radio Access Networks (CRAN)
  - Baseband processing centralized.

- Collaborative Video Conferencing
  - Video streams sent to cloud for processing.

- Real-time Control, Augmented Reality Apps, etc.
Soft Real-Time (SRT) Applications/Users

- Processing deadline
  - Uncompleted tasks are dropped.

- Can miss some deadlines
  - Long-term QoS.

- User: an instance of SRT applications

Challenge: Allocate resources in the cloud to support such applications/users.
Selected Related Work

- Hard real-time [Liu & Layland, 1973; Dertouzos & Mok, 1989; etc.]
- Soft real-time
  - (m, k)-firm deadline [Hamdaoui et al., 1995; etc.]
  - Imprecise computation [Liu et al., 1987; etc.]
  - Bounded tardiness [Liu et al., 2009; etc.]
  - Timely throughput [Hou & Kumar, 2012 & 2013 & 2014]
- Longest-Queue-First scheduling [Dimakis & Walrand, 2006; Kang et al., 2013; etc.]
- Stochastic scheduling [Piendo, 2012; etc.]
SRT-MIC (Multiple Identical Cores) Model

- System operates over period length $\delta$.
- In each period, each user generates one task with random, IID workloads.
  - Different distributions for different users.
  - No parallel processing.
- $\mu_i = \mathbb{E}[W_i]$
- Release time: start, Deadline: end
- QoS requirements: long-term effect
  - $q_i =$ % tasks completed on time.
  - Let $\mathbf{q} = (q_1, q_2, \cdots, q_n)$.
- $m$ identical cores.
NBUE Task Workload Assumption

**Assumption**

We assume all task workloads \( W_i \) satisfy NBUE.

**Def** A random variable \( X \) is said to satisfy New Better than Used in Expectation (NBUE) if for all \( t > 0 \),

\[
E[X - t | X > t] \leq E[X].
\]  

(1)

- In fact, we only require (1) for \( t \leq \delta \).
Resource Allocation Policies to Be Considered

In each period $\delta$:

- **Non-clairvoyant**
  - No future knowledge, e.g., workload realization.
  - Workload distribution available.
  - Preemption and migration allowed
  - We ignore the overheads.
  - System feasibility region $F$: set of $q$ fulfilled by non-clairvoyant policies.

Objective: Design stationary non-clairvoyant resource allocation policies to fulfill $q$. 
Outer Bound for System Feasibility Region $F$

$F$ is unknown except for single core and geometric workload*.

**Theorem:** For SRT-MIC model with NBUE workloads, the feasibility region for all non-clairvoyant policies is such that

$$F \subseteq R_{OB} \equiv \{ q \in \mathbb{R}_+^n \mid q \preceq 1, \sum_{i \in N} q_i \mu_i \leq m \delta \}.$$ 

- Intuitively, $q_i \mu_i$ “represents” expected time spent on user $i$.
- Total time spent on all users cannot exceed $m \delta$.
- Surprisingly, NOT hold for non-NBUE workloads.
  - E.g., $W_i = \begin{cases} 1 & \text{with probability 0.5} \\ 9 & \text{with probability 0.5.} \end{cases}$

*I.-H. Hou, and P. R. Kumar, “Queueing systems with hard delay constraints: a framework for real-time communication over unreliable wireless channels”.

9/39
Our Focus: Prioritization-Based Resource Allocation Policies

- **User Prioritization**: dynamically set priorities for users.
- **Priority-Based Resource Allocation**: allocate resources to favor users with higher priority.

Decomposition of Concerns.
Largest Deficit First: A Simple Policy

**Def** Largest Deficit First (LDF) policy: at period t

– Compute deficits

\[ X_i(t) = t \times q_i - (\# \text{ completed tasks up to } t). \]

  \[ \text{required \#} \quad \text{tracked from feedback} \]

– Sort \( X_i(t) \) from largest to smallest.

– Assign priorities accordingly.
Largest Deficit First (LDF) + Greedy Task Scheduler

User-Level QoS \( q = \{q_1, q_2, \cdots, q_n\} \)

LDF User Prioritization

Greedy Task Scheduler

Def Greedy Task Scheduler: Greedily process tasks by priority.

- Do not require knowledge of workload distribution.
- Non-preemptive approach.

Feedback
History
Events

Priority Decisions \( d \)

Decreasing Priority

1
2
3
Performance Evaluation for LDF+Greedy

**Def** The efficiency ratio of a non-clairvoyant policy $\eta$ is defined as

$$\gamma_{\eta} = \sup\{\gamma | \gamma F \subseteq F_{\eta}\}.$$  

E.g., $\gamma_{\eta} = 0.9 \Rightarrow F_{\eta}$ covers at least 90% $F$.

**Theorem**: For the SRT-MIC model with NBUE workloads, the efficiency ratio of LDF+Greedy exceeds $\gamma_1$ where

$$\gamma_1 = 1 - \frac{\max_{i \in N} \mu_i}{\delta}.$$
Efficiency Ratio of LDF+Greedy

**Theorem:** For the SRT-MIC model with NBUE workloads, the efficiency ratio of LDF+Greedy exceeds $\gamma_1$ where

$$\gamma_1 = 1 - \frac{\max_{i \in N} \mu_i}{\delta}.$$

- Intuition: a task is *unfinished* if it starts but does not complete.
  - At most 1 unfinished task per core per period.
- May NOT be true for non-NBUE.
- For large period $\delta$: LDF+Greedy is close to optimal.
- For small period $\delta$: we may need smarter task scheduler.
Deterministic Workloads: LDF + TS/OPT

- Let $q = \{q_1, q_2, \cdots, q_n\}$

### LDF Task Prioritization
- Feedback
- Priority Decisions $d$

### Task Selection
- User Subset $J(d)$

### Optimal Scheduling for $J(d)$

**Deterministic workloads:**

$$\Pr(W_i = \mu_i) = 1, \forall i \in N.$$  

**Task Selection (TS):** Greedily select users based on $d$ until sum workload $> m\delta$.

- $d_1 | d_2 | \cdots | m\delta$

- Let $J(d)$ be selected users.

**Optimal for $J(d)$:** Optimal policy guaranteeing $J(d)$ are completed.

- Need preemption/migration.
Efficiency Ratio of LDF+TS/OPT under Deterministic Workloads

**Theorem:** For the SRT-MIC model with deterministic workloads, the efficiency ratio of LDF+TS/OPT exceeds \( \gamma_2 \) where

\[
\gamma_2 = 1 - \frac{\max_{i \in N} \mu_i}{m \delta}.
\]

- Intuition: in each period, the wasted time \( m \delta - \sum_{i \in J(d)} \mu_i < \max_{i \in N} \mu_i \).
- Better than \( \gamma_1 = 1 - \frac{\max_{i \in N} \mu_i}{\delta} \).
- Applicable to workloads with small variability.
Simulation: Resource Savings vs. Some Baseline Approach

Resource requirements of policy $\eta$: under policy $\eta$, the required # of cores $m$ given $n$, $\delta$, $W_i$ distributions and $Q$.

Resource savings of policy $\eta$ vs. reservation-based approach.

Simulation results:
• Savings can be as large as 40% - 80%.
Generalization of Results

• **Uniform cores**: Cores have different processing speeds.
  • SRT-MIC: all speeds = 1.

• **Different periods**: Users generate tasks with different periods.
  • SRT-MIC: all periods are the same.

• **Sub-tasks**: Each task further consists of several sub-tasks that need to be processed in order.
  • SRT-MIC: each task is an entity.
# Generalization: Uniform Cores

<table>
<thead>
<tr>
<th></th>
<th>SRT-MIC</th>
<th>Uniform Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{RB}$</td>
<td>${q \in \mathbb{R}<em>+^n \mid q \preceq 1, \quad \sum</em>{i \in N} w_i(q_i) \leq m \delta, \quad w_i(q_i) \leq \delta, \forall i \in N}$</td>
<td>${q \in \mathbb{R}_+^n \mid q \preceq 1, \quad B_n(q) \leq S_m \cdot \delta, \quad B_k(q) \leq S_k \cdot \delta, \quad 1 \leq k \leq m}$</td>
</tr>
<tr>
<td>$R_{OB}$</td>
<td>${q \in \mathbb{R}<em>+^n \mid q \preceq 1, \quad \sum</em>{i \in N} q_i \mu_i \leq m \delta}$</td>
<td>${q \in \mathbb{R}<em>+^n \mid q \preceq 1, \quad \sum</em>{i \in N} q_i \mu_i \leq S_m \cdot \delta}$</td>
</tr>
<tr>
<td>$\gamma_1$ (preemptive)</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{\delta}$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{S \cdot \delta}$</td>
</tr>
<tr>
<td>$\gamma_1$ (non-pre)</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{\delta}$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{\min_{c \in C} s_c \cdot \delta}$</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{m \delta}$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{S_m \cdot \delta}$</td>
</tr>
</tbody>
</table>

**Notation:**

- $s_c$ : speed for core $c$.
- $S_k$ : sum of largest $k$ speeds.
- $B_k(q)$ : sum of largest $k$ $w_i(q_i)$.
## Generalization: Different Periods

<table>
<thead>
<tr>
<th></th>
<th>SRT-MIC</th>
<th>Different Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_{RB} )</td>
<td>( { q \in \mathbb{R}^n_+</td>
<td>q \leq 1, \sum_{i \in N} w_i(q_i) \leq m\delta, w_i(q_i) \leq \delta, \forall i \in N } )</td>
</tr>
<tr>
<td>( R_{OB} )</td>
<td>( { q \in \mathbb{R}^n_+</td>
<td>q \leq 1, \sum_{i \in N} q_i \mu_i \leq m\delta } )</td>
</tr>
<tr>
<td>( \gamma_1 ) (preemptive)</td>
<td>( 1 - \frac{\max_{i \in N} \mu_i}{\delta} )</td>
<td>N/A</td>
</tr>
<tr>
<td>( \gamma_1 ) (non-pre)</td>
<td>( 1 - \frac{\max_{i \in N} \mu_i}{m\delta} )</td>
<td>( 1 - \frac{\max_{i \in N} \frac{\mu_i}{\delta_i}}{m} )</td>
</tr>
</tbody>
</table>

**Notation:**
- \( \delta_i \): period for user \( i \).
# Generalization: Sub-tasks

<table>
<thead>
<tr>
<th></th>
<th>SRT-MIC</th>
<th>Sub-tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{RB}$</td>
<td>${q \in \mathbb{R}^n_+ \mid q \leq 1, \sum_{i \in N} w_i(q_i) \leq m\delta, w_i(q_i) \leq \delta, \forall i \in N}$</td>
<td>${q \in \mathbb{R}^n_+ \mid q \leq 1, \sum_{i \in N} w_i(q_i) \leq m\delta, w_i(q_i) \leq \delta, \forall i \in N}$</td>
</tr>
<tr>
<td>$R_{OB}$</td>
<td>${q \in \mathbb{R}^n_+ \mid q \leq 1, \sum_{i \in N} q_i \mu_i \leq m\delta}$</td>
<td>${q \in \mathbb{R}^n_+ \mid q \leq 1, \sum_{i \in N} q_i \mu_i \leq m\delta}$</td>
</tr>
<tr>
<td>$\gamma_1$ (preemptive)</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{\delta}$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{\delta}$</td>
</tr>
<tr>
<td>$\gamma_1$ (non-pre)</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{m\delta}$</td>
<td>$1 - \frac{\max_{i \in N} \mu_i}{m\delta}$</td>
</tr>
</tbody>
</table>

Everything remains unchanged.
Conclusions

Resource allocation in complex systems supporting real-time users can be “easy”

• One can consider priority-based resource allocation and adopt simple LDF policies.

• Given dynamic priorities:
  • If real-time constraints are “loose”, LDF+Greedy are near-optimal and can achieve substantial resource savings.
  • For “tight” constraints, it is worth exploring other policies. E.g., for workload with small variability, we propose LDF+TS/OPT.
Thank you!
Reservation-Based Static Sharing

- Dedicated resources for each user to guarantee QoS.
- Define $q_i$-percentile point $w_i(q_i)$ such that
  \[ \Pr(W_i \leq w_i(q_i)) = q_i. \]
- Not efficient.

\[
F_{RB} = \{ q \in \mathbb{R}^n_+ \mid q \preceq 1, \sum_{i \in N} w_i(q_i) \leq m\delta, w_i(q_i) \leq \delta, \forall i \in N \}
\]

A. Atlas and A. Bestavros, “Statistical Rate Monotonic Scheduling”. 24/39
General Payoff Model

- System operates over periods.
- n users generate tasks with random workloads in each period.
- In each period, system picks a priority decision d
  - \( d = (d_1, d_2, \cdots, d_n) \) where \( d_m \) is the user with priority m.
- Given d, each user i gets a random payoff \( V_i(d) \geq 0 \).
  - Randomness due to workload, system execution, etc.
  - E.g., # of tasks completed, or partial task completion.
- \( V(d) = (V_1(d), V_2(d), \cdots, V_n(d)) \).
  - Independent across periods; Distribution depends on d.
- \( p(d) = E[V(d)] \) expected payoff vector.
QoS Requirements & Policies to Be Considered

**QoS requirements**: Each user $i$ requires long-term average payoff $q_i \geq 0$.

- Let $\mathbf{q} = (q_1, q_2, \cdots, q_n)$.

We consider stationary **user prioritization policies** that pick priority decisions at each period based on the following:

- Expected payoff vectors $P = \{ \mathbf{p}(d) \}$;
- QoS requirement vector $\mathbf{q}$;
- Past history of payoffs (not including current period).

**Objective**: Develop user prioritization policies to fulfill $\mathbf{q}$
What q’s are Feasible Given $P = \{p(d)\}$?

Visualizing Feasibility Region

Theorem: $q \in C \iff q$ is feasible under some user prioritization policy

$$C \equiv \{q \in \mathbb{R}^n_+ \mid \exists x \in \text{Conv}(P) \text{ such that } q \preceq x\}$$
“Shape” of Feasibility Region

Feasibility region depends on

\[ P = \{ p(d) \} \]

\[ P = \{ p(d) \} \text{ need not lie on same hyper-plane} \]
Feasibility “Optimal” Policy

**Def** A user prioritization policy is *feasibility optimal* if it can achieve any \( q \in C \).

**Def** Deficit-based max-weight policy: at period \( t \)

- Compute deficit

\[
X_i(t) = (\text{required payoffs}) - (\text{sum of payoffs up to } t)
\]

\[
t \times q_i
\]

- Choose decision \( d^*(t) \) such that

\[
d^*(t) \in \text{arg max}_d \langle X(t), p(d) \rangle
\]
**Theorem:** The deficit-based max-weight policy is feasibility optimal.

BUT,

- Need to know \( P = \{ p(d) \} \).
- Solve high complexity optimization \( O(n!) \).
Monotonicity in Payoffs

**Notation** \( S_i(d) \): users having higher priority than \( i \) in decision \( d \)

\[
d = (j, k, \ldots, l, i, \ldots)
\]

\( S_i(d) \)

**Def** The system with \( P = \{p(d)\} \) is said to satisfy **monotonicity in payoffs** if, for any two priority decisions \( d_1, d_2 \) and any user \( i \),

\[
S_i(d_1) \subseteq S_i(d_2) \Rightarrow p_i(d_1) \geq p_i(d_2).
\]

- E.g., \( d_1 = (j, k, i, \ldots) \), \( d_2 = (j, k, l, i, \ldots) \).
- This characterizes how priorities impact the expected payoffs.
Geometry of $R_{IB}$

**Theorem:** If the system satisfies monotonicity in payoffs, then

$$R_{IB} \sim C - \text{Conv}(P).$$

$C = R_{IB}$ is a surface.
New Results on Optimality of w-LDF

• Sufficient condition for w-LDF’s optimality
  • Intuition: expected payoff vectors \( P = \{p(d)\} \) are on the same hyper-plane*.

• Efficiency analysis for w-LDF
  • By characterizing the “heterogeneity” of vectors in \( P \)

• w-LDF policies are optimal when \( n=2 \)
  • Hierarchical-LDF for 2 classes of exchangeable users.

• Effect of weights

* Akin but different from the conditions proposed by Dimakis and Walrand.
Resource Savings vs. Reservation-Based Approach

Resource requirements of policy $\eta$: under policy $\eta$, the required # of cores $m$ given $n, \delta, W_i$ distributions and $q$.

Resource savings of policy $\eta$ vs. reservation-based approach.

• We study $1 - \frac{m_\eta}{m_{RB}}$.

Simulation setup:

• Same workload distribution and $q_i$.
• In each period, independently generate task workloads.
• $q$ feasible if, fractions of task completions over 3000 periods exceeds $q$.
Resource Savings of LDF+Greedy: Large Period Regime

• # of users $n = 200$
• Period $\delta = 50$
• $W_i \sim \text{Gamma}(5, 1)$

Savings vs. Reservation-Based

- LDF+Greedy close to optimal for large period.
- Savings can be large: 40% - 80%
- Savings curve depends on workload distribution.
Resource Savings of LDF+Greedy vs. LDF+TS/OPT: Deter. Workloads and Small $\delta$

- $n = 30$
- Period $\delta = 9$
- $W_i = 5$
  - Deterministic Workload

- LDF+TS/OPT (Task Selection/Optimal) is close to optimal.
- LDF+Greedy not working well.
- Savings monotonically decrease.
Resource Savings of LDF+TS/OPT: Workloads with Small Variability

- \( n = 30 \)
- Period \( \delta = 9 \)
- \( W_i \sim \text{Gamma}(100, 0.05) \)
  - Same mean = 5
  - Small variance

- Heuristic LDF+TS/OPT better than LDF+Greedy.
- Heuristic LDF+TS/OPT degrades for large \( q \).
  - Due to the fact that some selected tasks do not complete.
Old Backup Slides
System Model for Cloud-Based SRT Applications

- System operates over period length $\delta$.
- In each period, each user generates one task with random, IID workloads.
  - Task may consist of a sequence of sub-tasks.
  - No task parallel processing.
  - $\mu_i = E[W_i]$  
- Release time: start, Deadline: end
  - $Q_i = \%$ tasks completed on time.
    - i.e., payoff is whether task completes.
- $m$ uniform cores.
  - Possibly different speeds.
NBUE Workload Assumption

**Def** A random variable $X$ is said to satisfy New Better than Used in Expectation (NBUE) if for all $t > 0$,

$$E[X - t | X > t] \leq E[X].$$

**Assumption** We assume all $W_i$ satisfy NBUE.

- Weighted sum of NBUEs are also NBUE.
- NBUE includes Exp, Gamma with shape parameter $k \geq 1$, Deterministic, etc.
- A common non-NBUE class is the heavy-tailed one, but real-time task workloads are not likely to have such tails.
**Largest Deficit First (LDF) + Greedy Task Scheduler**

- **Greedily process tasks by priority.**
- Do not require knowledge of workload distribution.
- Non-preemptive approach.

Def **Greedy** Task Scheduler:

- Do not require knowledge of workload distribution.
- Non-preemptive approach.

User-Level QoS \( q = \{q_1, q_2, \cdots, q_n\} \)

LDF User Prioritization

Greedy Task Scheduler

Feedback

History

Events

Priority Decisions \( d \)

Core 1: 1 3 6

Core 2: 2 4 5
Efficiency Ratio of LDF+TS/OPT under Deterministic Workloads

**Theorem** For the SRT-MIC model with **deterministic** workloads, the efficiency ratio of LDF+TS/OPT exceeds $\gamma_2$ where

$$\gamma_2 = 1 - \frac{\max_{i \in N} \mu_i}{m \delta} .$$

- Intuition: in each period, the wasted time $m \delta - \sum_{i \in J(d)} \mu_i < \max_{i \in N} \mu_i$.
- Better than $\gamma_1 = 1 - \frac{\max_{i \in N} \mu_i}{\delta}$.
- If we know workload realization we can still use TS/LLREF, but that is clairvoyant design.
- For workloads with small variability, heuristic TS/LLREF based on some workload estimations outperforms Greedy in simulation.
Resource Requirements & Simulation Setup

Resource requirements: the required # of cores $m$ given $n$, $\delta$, $W_i$ distributions and $q$.

- Reservation-based static sharing:
  \[
  m_{RB} = \left\lceil \frac{\sum_{i \in N} w_i(q_i)}{\delta} \right\rceil
  \]

- Lower bound on $m$:
  \[
  m \equiv \left\lceil \frac{\sum_{i \in N} q_i \mu_i}{\delta} \right\rceil
  \]

Simulation setup:
- Homogeneous workload distribution and requirement.
- In each period, we independently generate task workloads.
- $q$ is feasible if, fractions of task completions over 3000 periods exceeds $q$. 