Real-Time Calculus

Prof. Dr. Jian-Jia Chen

LS 12, TU Dortmund

20, Jan., 2016
Arbitrary Deadlines

The worst-case response time of $\tau_i$ by only considering the first job of $\tau_i$ at the critical instant is too optimistic when the relative deadline of $\tau_i$ is larger than the period.

Consider two tasks:

- $\tau_1$ has period 70 and execution time 26 and $\tau_2$ is with period 100 and execution time 62.
- $\tau_2$’s seven jobs have the following response times, respectively: 114, 102, 116, 104, 118, 106, 94.
- Note that the first job’s response time is not the longest.
Busy Intervals

Definition

A $\tau_i$-level busy interval $(t_0, t]$ of task $\tau_i$ begins at an instant $t_0$ when

1. all jobs in $\tau_i$ released before $t$ have completed, and
2. a job of $\tau_i$ releases.

The interval ends at the first instant $t$ after $t_0$ when all jobs in $\tau_i$ released since $t_0$ are complete.
Abstract Models for Real-Time Calculus

Concrete Instance

Abstract Representation

Input Stream

Processor

Tasks

Service Model

Load Model

Processing Model

Abstract Representation

Concrete Instance
Abstract Models for Module Performance Analysis

Concrete Instance

Input Stream

Abstract Representation

\[\alpha, \alpha', \beta_{CPU}, \beta_{BUS}, \beta_{DSP}\]
### Overview

<table>
<thead>
<tr>
<th>Math. View</th>
<th>Real-Time Calculus (RTC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System View</td>
<td>Module Performance Analysis (MPA)</td>
</tr>
<tr>
<td>Math. View</td>
<td>Min-Plus Calculus, Max-Plus Calculus</td>
</tr>
</tbody>
</table>
Backgrounds

• Real-Time Calculus can be regarded as a worst-case/best-case variant of classical queuing theory. It is a formal method for the analysis of distributed real-time embedded systems.

• Related Work:
Definition of Arrival Curves and Service Curves

- For a specific trace:
  - Data streams: $R(t) = \text{number of events in } [0, t)$
  - Resource stream: $C(t) = \text{available resource in } [0, t)$
- For the worst cases and the best cases in any interval with length $\Delta$:
  - Arrival Curve $[\alpha^l, \alpha^u]$:
    \[
    \alpha^l(\Delta) = \inf_{\lambda \geq 0, \forall R} \{ R(\Delta + \lambda) - R(\lambda) \}
    \]
    \[
    \alpha^u(\Delta) = \sup_{\lambda \geq 0, \forall R} \{ R(\Delta + \lambda) - R(\lambda) \}
    \]
  - Service Curve $[\beta^l, \beta^u]$:
    \[
    \beta^l(\Delta) = \inf_{\lambda \geq 0, \forall C} \{ C(\Delta + \lambda) - C(\lambda) \}
    \]
    \[
    \beta^u(\Delta) = \sup_{\lambda \geq 0, \forall C} \{ C(\Delta + \lambda) - C(\lambda) \}
    \]
Abstract Models for Real-Time Calculus

Concrete Instance

Abstract Representation

Input Stream

Processor\( C(t) \)

Tasks

Load Model

Service Model

\( R(t) \)

\( R'(t) \)

\( \alpha(\Delta) \)

\( \beta(\Delta) \)

Prof. Dr. Jian-Jia Chen (LS 12, TU Dortmund)
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.

![Diagram showing arrival curve with upward arrows indicating events and a sliding window to calculate the upper bound.](image)

Prof. Dr. Jian-Jia Chen  (LS 12, TU Dortmund)
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.
Arrival Curve: An Example

Use a sliding window to get the upper bound of the number of events in a specified interval length.

- Maximum events in 3 units
- Possible events in 3 units
- Minimum events in 3 units
Service Curve: An Example

Resource
Availability

Service Curves
$\beta = [\beta^l, \beta^u]$
Example 1: Periodic with Jitter

A common event pattern that is used in literature can be specified by the parameter triple \((p, j, d)\), where \(p\) denotes the period, \(j\) the jitter, and \(d\) the minimum inter-arrival distance of events in the modeled stream.
Example 1: Periodic with Jitter

\[ \alpha^u(\Delta) = \left\lceil \frac{\Delta}{p} \rightceil \]

\[ \alpha^l(\Delta) = \left\lfloor \frac{\Delta}{p} \right\rfloor \]

\[ \alpha^u(\Delta) = \left\lceil \frac{\Delta + j}{p} \right\rceil \]

\[ \alpha^l(\Delta) = \left\lfloor \frac{\Delta - j}{p} \right\rfloor \]
Example 1: Periodic with Jitter

\[ \alpha^u(\Delta) = \min \left\{ \left\lceil \frac{\Delta + j}{p} \right\rceil, \left\lceil \frac{\Delta}{d} \right\rceil \right\} \]

\[ \alpha^l(\Delta) = \left\lfloor \frac{\Delta - j}{p} \right\rfloor \]
More Examples on Arrival Curves

(a) 

(b) 

(c) 

(d)
Example 2: TDMA Resource

- Consider a real-time system consisting of \( n \) applications that are executed on a resource with bandwidth \( B \) that controls resource access using a TDMA (Time Division Multiple Access) policy.

- Analogously, we could consider a distributed system with \( n \) communicating nodes, that communicate via a shared bus with bandwidth \( B \), with a bus arbitrator that implements a TDMA policy.

- TDMA policy: In every TDMA cycle of length \( \bar{c} \), one single resource slot of length \( s_i \) is assigned to application \( i \).
Example 2: TDMA Resource

\[ \beta^u(\Delta) = B \min \left\{ \frac{\Delta}{\bar{c}} s_i, \Delta - \left\lfloor \frac{\Delta}{\bar{c}} \right\rfloor (\bar{c} - s_i) \right\} \]

\[ \beta^l(\Delta) = B \max \left\{ \frac{\Delta}{\bar{c}} s_i, \Delta - \left\lceil \frac{\Delta}{\bar{c}} \right\rceil (\bar{c} - s_i) \right\} \]
More Examples on Service Curves

- **Full resource**
  - $\beta^u$
  - $\beta^l$

- **Bounded delay**
  - $\beta^u$
  - $\beta^l$

- **TDMA resource**
  - $\beta^u$
  - $\beta^l$

- **Periodic resource**
  - $\beta^u$
  - $\beta^l$
Abstraction

\[ C(t) \]

\[ R(t) \rightarrow GPC \rightarrow R'(t) \rightarrow C'(t) \]

- time domain cumulative functions

\[ \beta(\Delta) \]

\[ \alpha(\Delta) \rightarrow GPC \rightarrow \alpha'(\Delta) \rightarrow \beta'(\Delta) \]

- time-interval domain variability curves
Greedy Processing Component (GPC)

- Component is triggered by incoming events.
- A fully preemptable task is instantiated at every event arrival to process the incoming event.
- Active tasks are processed in a greedy fashion in FIFO order.
- Processing is restricted by the availability of resources.
Some Relations (only for your reference)

- The output upper arrival curve of a component satisfies
  \[ \alpha^{u'} \leq (\alpha^u \ominus \beta^l) \]
  with a simple and pessimistic calculation.

- The remaining lower service curve of a component satisfies
  \[ \beta^{l''}(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta^l(\lambda) - \alpha^u(\lambda)) \]
More Relations (only for your reference)

\[ \alpha^{u'} = [(\alpha^u \otimes \beta^u) \otimes \beta^l] \land \beta^u \]
\[ \alpha'' = [(\alpha^u \otimes \beta^l) \otimes \beta^l] \land \beta^l \]
\[ \beta^{u'} = (\beta^u - \alpha^l) \overline{\otimes} 0 \]
\[ \beta'' = (\beta^l - \alpha^u) \overline{\otimes} 0 \]

Without formal proofs....
Graphical Interpretation

\[ B = \sup_{t \geq 0} \{ R(t) - R'(t) \} \leq \sup_{\lambda \geq 0} \{ \alpha^u(\lambda) - \beta^l(\lambda) \} \]

\[ D = \sup_{t \geq 0} \{ \inf \{ \tau \geq 0 : R(t) \leq R'(t + \tau) \} \} \]

\[ = \sup_{\Delta \geq 0} \{ \inf \{ \tau \geq 0 : \alpha^u(\Delta) \leq \beta^l(\Delta + \tau) \} \} \]
System Composition

Concrete Instance

How to Interconnect service?

Scheduling

\[ \beta_{CPU} \]

\[ \beta_{BUS} \]

\[ \beta_{DSP} \]
Scheduling and Arbitration

FP/RM \[\beta\] GPC \[\alpha_A \rightarrow \alpha'_A\] GPC \[\alpha_B \rightarrow \alpha'_B\] GPS \[\beta\] share GPC \[\alpha_A \rightarrow \alpha'_A\] GPC \[\alpha_B \rightarrow \alpha'_B\] sum \[\beta'\]

EDF \[\beta\] EDF \[\alpha_A \rightarrow \alpha'_A\] \[\alpha_B \rightarrow \alpha'_B\]

RR \[\beta\] RR \[\alpha_A \rightarrow \alpha'_A\] \[\alpha_B \rightarrow \alpha'_B\]

TDMA \[\beta\] TDMA \[\alpha_A \rightarrow \alpha'_A\] \[\alpha_B \rightarrow \alpha'_B\] \[\beta'_{s1} \beta'_{s2}\]
Mixed Hierarchical Scheduling

TDMA + FP/RM + EDF + RR

TDMA

FP

EDF

RR

FP/RM + EDF

FP

EDF

FP

EDF

FP

EDF

...and many other combinations:

RR + EDF

FP/RM + RR

FP/RM + GPS

GPS + EDF
Complete System Composition

Input Stream

Concrete Instance

Abstract Representation

Prof. Dr. Jian-Jia Chen (LS 12, TU Dortmund)
RTC Toolbox (http://www.mpa.ethz.ch/Rtctoolbox)

Modular Performance Analysis with Real-Time Calculus

Real-Time Calculus Toolbox

Latest News
[2010-07-20]: Interface to SymTA/S analysis tool.
[2010-07-20]: Extensions for structured event streams.
[2009-01-30]: BugFix and Update released.
[2008-12-23]: Beta Version 1.2 released.
[2008-10-14]: BugFix released.
[2008-07-16]: BugFix released.
[2008-02-05]: BugFix released.
[2007-09-24]: New components and tutorial.
[2007-03-21]: BugFix released.
[2006-10-02]: New tutorials and Java API released.
[2006-10-02]: BugFix released.
[2006-04-04]: First tutorial published.
[2006-02-27]: First official beta version released.

Overview
The Real-Time Calculus (RTC) Toolbox is a free Matlab toolbox for system-level performance analysis of distributed real-time and embedded systems.

The RTC Toolbox is based on an efficient representation of Variability Characterization Curves (VCC’s) and implements most min-plus and max-plus algebra operators for these curves. On top of the min-plus and max-plus algebra operators, the RTC Toolbox provides a library of functions for Modular Performance Analysis with Real-Time Calculus.
Advantages and Disadvantages of RTC and MPA

- **Advantages**
  - More powerful abstraction than “classical” real-time analysis
  - Resources are first-class citizens of the method
  - Allows composition in terms of (a) tasks, (b) streams, (c) resources, (d) sharing strategies.

- **Disadvantages**
  - Needs some effort to understand and implement
  - Extension to new arbitration schemes not always simple
  - *Not applicable for schedulers that change the scheduling policies dynamically.*