Real-Time Calculus

Prof. Dr. Jian-Jia Chen

LS 12, TU Dortmund

18, Jan., 2017
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 Module Performance Analysis

Input Stream

Concrete Instance

Abstract Representation

Prof. Dr. Jian-Jia Chen (LS 12, TU Dortmund)
Overview

System View
- Module Performance Analysis (MPA)
- Real-Time Calculus (RTC)
- Min-Plus Calculus, Max-Plus Calculus

Math. View
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
- Input Stream
- Processor
- Tasks
- Load Model
- Service Model
- Processing Model
- Abstract Representation

Abstract Representation
- \( \alpha(\Delta) \)
- \( \beta(\Delta) \)

Concrete Instance
- \( R(t) \)
- \( C(t) \)
- \( R'(t) \)
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.

![Diagram showing arrival curve with time intervals and events]

0 2 4 6 8 10 12 14 → time
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.
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

Periodic

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

Periodic with Jitter

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

Prof. Dr. Jian-Jia Chen (LS 12, TU Dortmund)
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 $i$. 

![Diagram showing TDMA allocation]
Example 2: TDMA Resource

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

\[ \beta^l(\Delta) = B \max \left\{ \begin{bmatrix} \frac{\Delta}{\bar{c}} \end{bmatrix} s_i, \Delta - \begin{bmatrix} \frac{\Delta}{\bar{c}} \end{bmatrix} (\bar{c} - s_i) \right\} \]
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

\[
\begin{align*}
C(t) & \quad \rightarrow \quad C'(t) \\
R(t) & \quad \rightarrow \quad R'(t) \\
\beta'(\Delta) & \quad \rightarrow \quad \beta(\Delta) \\
\alpha'(\Delta) & \quad \rightarrow \quad \alpha(\Delta)
\end{align*}
\]

time domain cumulative functions

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
  \[ \alpha'' \leq (\alpha' \odot \beta') \]
  with a simple and pessimistic calculation.
- The remaining lower service curve of a component satisfies
  \[ \beta''(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta'(\lambda) - \alpha'(\lambda)) \]
More Relations (only for your reference)

\[ \alpha'' = \left[ (\alpha' \otimes \beta') \otimes \beta \right] \land \beta' \]
\[ \alpha'' = \left[ (\alpha' \otimes \beta') \otimes \beta \right] \land \beta' \]
\[ \beta'' = (\beta' \land \alpha') \triangleleft 0 \]
\[ \beta'' = (\beta' \land \alpha') \triangleleft 0 \]

Without formal proofs....
Graphical Interpretation

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

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

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

Concrete Instance

How to Interconnect service?

Scheduling

β_{CPU} → GPC

β_{BUS} → GPC

β_{DSP} → GPC

α → GPC

α' → GPC
Mixed Hierarchical Scheduling

...and many other combinations:
- RR + EDF
- FP/RM + RR
- FP/RM + GPS
- GPS + EDF
Complete System Composition

Input Stream

Concrete Instance

Abstract Representation

β_{CPU}

β_{BUS}

β_{DSP}
RTC Toolbox (http://www.mpa.ethz.ch/Rtctoolbox)
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.