Real-Time Calculus and Module Performance Analysis

Prof. Dr. Jian-Jia Chen

LS 12, TU Dortmund

14 June 2016
Abstract Models for Real-Time Calculus

Concrete Instance

Abstract Representation

Load Model

Processor

Service Model

Input Stream

Tasks

Processing Model
Abstract Models for Module Performance Analysis

Input Stream

Concrete Instance

Abstract Representation
Overview

System View

- Module Performance Analysis (MPA)

Math. View

- Real-Time Calculus (RTC)
- Min-Plus Calculus, Max-Plus Calculus
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:
Plus-Times and Min-Plus Algebras

- Algebraic structure
  - a set of (finite or infinite) elements $S$
  - one or more operators defined on the elements of this set
- Plus-Times Algebra: Two operators $+$ and $\times$, denoted by $(S, +, \times)$
- Min-Plus Algebra
  - Two operators $\oplus$ (min) and $\otimes$ (plus), denoted by $(S \cup \{+\infty\}, \inf, +)$
  - Infimum:
    - The infimum of a subset of some set is the greatest element, not necessarily in the subset, that is less than or equal to all other elements of the subset.
    - For example, $\inf\{[a, b]\} = a$, $\inf\{(a, b)\} = a$, where $\min\{[a, b]\} = a$, $\min\{(a, b)\} = \text{undefined}$.
  - Supremum:
    - The supremum of a subset of some set is the smallest element, not necessarily in the subset, that is more than or equal to all other elements of the subset.
    - For example, $\sup\{[a, b]\} = b$, $\sup\{[a, b]\} = b$, where $\max\{[a, b]\} = b$, $\max\{[a, b]\} = \text{undefined}$.
Min-Plus Algebra: Properties for $\otimes$

Suppose that $a, b, c \in S$. We have

- **Closure:** $a \otimes b \in S$
- **Associativity:** $a \otimes (b \otimes c) = (a \otimes b) \otimes c$
- **Commutativity:** $a \otimes b = b \otimes a$
- **Existence of identity element:** $\exists \nu : a \otimes \nu = a$.
- **Existence of negative element:** $\exists a^{-1} : a \otimes a^{-1} = \nu$.
- **Distributivity of $\otimes$ with respect to $\oplus$:**
  \[ a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \]

**Examples**

- **plus-times:** $a \times (b + c) = a \times b + a \times c$
- **min-plus:** $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) = \inf\{a + b, a + c\}$. 

Prof. Dr. Jian-Jia Chen (LS 12, TU Dortmund) 7 / 39
Min-Plus Algebra: Properties for $\oplus$

Suppose that $a, b, c \in S$. We have

- **Closure:** $a \oplus b \in S$
- **Associativity:** $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
- **Commutativity:** $a \oplus b = b \oplus a$
- **Existence of identity element:** $\exists \epsilon : a \oplus \epsilon = a$.
- **Property of $\epsilon$ regarding $\otimes$:** $a \otimes \epsilon = \epsilon$.

Examples:

- **plus-times:** $\exists 0 : a + 0 = a$.
- **min-plus:** $a \oplus a = a$. 

Definition of Arrival Curves and Service Curves

- For a specific trace:
  - Data streams: $R(t) =$ number of events in $[0, t)$
  - Resource stream: $C(t) =$ 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

Load Model

Processing Model

Service Model

Processor

Tasks

Input Stream

R(t)

C(t)

R'(t)

β(∆)

α(∆)

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.

![Diagram showing time and events](image)
Arrival Curve: An Example

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

```
0  2  4  6  8  10  12  14
```

`\Delta`
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 the concept of an arrival curve with a sliding window. The diagram includes a time axis marked with intervals 0 to 14 and vertical arrows indicating events. The diagram also shows a sliding window and a triangle symbolizing the upper bound of the number of events.]
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.

\[
\begin{array}{c}
\uparrow \quad \uparrow \uparrow \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \\
0 \quad 2 \quad 4 \quad 6 \quad 8 \quad 10 \quad 12 \quad 14 \quad \text{time}
\end{array}
\]
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 Diagram]
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

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

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

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

\[ \beta^l(\Delta) = B \max \left\{ \left\lceil \frac{\Delta}{\bar{c}} \right\rceil s_i, \Delta - \left\lfloor \frac{\Delta}{\bar{c}} \right\rfloor (\bar{c} - s_i) \right\} \]
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**
  - $\beta^u$
  - $\beta^l$
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.
By conservation law:

\[ C(t) = C'(t) + R'(t) \]

\[ B(t) = R(t) - R'(t) \]

Therefore,

\[ R'(t) = \inf_{0 \leq \lambda \leq t} \{ R(\lambda) + C(t) - C(\lambda) \} \]

and

\[ C'(t) = \sup_{0 \leq \lambda \leq t} \{ C(\lambda) - R(\lambda) \} \]
Analysis on GPC

- By conservation law: \( R'(\lambda) \leq R(\lambda) \) for any \( \lambda \geq 0 \).
- Since the output cannot be larger than the available resource, we also have \( R'(t) \leq R'(\lambda) + C(t) - C(\lambda) \).
- By the two items above, we know \( R'(t) \leq R(\lambda) + C(t) - C(\lambda) \).
- Suppose that \( \lambda^* \) is the latest time before \( t \) such that the buffer is empty. That is, \( R'(\lambda^*) = R(\lambda^*) \) and \( R'(t) = R'(\lambda^*) + C(t) - C(\lambda^*) = R(\lambda^*) + C(t) - C(\lambda^*) \).
- As a result, we know that

\[
R'(t) = \inf_{0 \leq \lambda \leq t} \{ R(\lambda) + C(t) - C(\lambda) \}
\]
Analysis on GPC

- By conservation law: $R'(\lambda) \leq R(\lambda)$ for any $\lambda \geq 0$.
- Since the output cannot be larger than the available resource, we also have $R'(t) \leq R'(\lambda) + C(t) - C(\lambda)$.
- By the two items above, we know $R'(t) \leq R(\lambda) + C(t) - C(\lambda)$.
- Suppose that $\lambda^*$ is the latest time before $t$ such that the buffer is empty. That is, $R'(\lambda^*) = R(\lambda^*)$ and $R'(t) = R'(\lambda^*) + C(t) - C(\lambda^*) = R(\lambda^*) + C(t) - C(\lambda^*)$.
- As a result, we know that

$$R'(t) = \inf_{0 \leq \lambda \leq t} \{ R(\lambda) + C(t) - C(\lambda) \}$$

- The analysis is similar for

$$C'(t) = \sup_{0 \leq \lambda \leq t} \{ C(\lambda) - R(\lambda) \}$$
Convolutions

- Plus-times system theory: signals $f$, impulse response $g$, convolution in time domain:

$$h(t) = (f \times g)(t) = \int_0^t f(t - s)g(s)ds,$$

where $f, g$ can be thought of as signals and impulse response, respectively.

- Min-Plus system theory: streams $R$, variability curves $g$, convolution in time-interval domain:

$$R'(t) \geq (R \otimes g)(t) = \inf_{0 \leq \lambda \leq t} \{R(t - \lambda) + g(\lambda)\}.$$
Abstraction

time domain cumulative functions

time-interval domain variability curves
Convolution and De-convolution

- \( f \otimes g \) is called \textit{min-plus convolution}

\[
(f \otimes g)(t) = \inf_{0 \leq \lambda \leq t} \{ f(t - \lambda) + g(\lambda) \}
\]

- \( f \oslash g \) is called \textit{min-plus de-convolution}

\[
(f \oslash g)(t) = \sup_{0 \leq \lambda} \{ f(t + \lambda) - g(\lambda) \}
\]

- \( f \bar{\otimes} g \) is called \textit{max-plus convolution}

\[
(f \bar{\otimes} g)(t) = \sup_{0 \leq \lambda \leq t} \{ f(t - \lambda) + g(\lambda) \}
\]

- \( f \bar{\oslash} g \) is called \textit{max-plus de-convolution}

\[
(f \bar{\oslash} g)(t) = \inf_{0 \leq \lambda} \{ f(t + \lambda) - g(\lambda) \}
\]
Arrival and Service Curves Revisit

\[ \alpha^l(t - s) \leq R(t) - R(s) \leq \alpha^u(t - s) \quad \forall s \leq t. \]
\[ \beta^l(t - s) \leq C(t) - C(s) \leq \beta^u(t - s) \quad \forall s \leq t. \]

Therefore, by using the convolution and de-convolution, we know that

\[ \alpha^u = R \odot R; \quad \alpha^l = R \bar{\odot} R; \quad \beta^u = C \odot C; \quad \beta^l = C \bar{\odot} C; \]

The proof for \( \alpha^u \):

\[ \alpha^u(\Delta) = \sup_{\lambda \geq 0} \{ R(\Delta + \lambda) - R(\lambda) \} \geq R(\Delta + \lambda) - R(\lambda), \quad \forall \lambda \geq 0. \]
Tight Curves

A curve \( f \) is *sub-additive*, if

\[
f(a) + f(b) \geq f(a + b) \quad \forall a, b \geq 0.
\]

The sub-additive *closure* \( \overline{f} \) of a curve \( f \) is the largest sub-additive curve with \( \overline{f} \leq f \) and is computed as

\[
\overline{f} = \min\{f, (f \otimes f), (f \otimes f \otimes f), \ldots\}.
\]

*If \( f \) is interpreted as an arrival curve, then any trace \( R \) that is upper bounded by \( f \) is also upper bounded by the sub-additive closure \( \overline{f} \).*
Tight Curves

A curve \( f \) is \textit{sub-additive}, if

\[
f(a) + f(b) \geq f(a + b) \quad \forall a, b \geq 0.
\]

The sub-additive \textit{closure} \( \bar{f} \) of a curve \( f \) is the largest sub-additive curve with \( \bar{f} \leq f \) and is computed as

\[
\bar{f} = \min\{f, (f \otimes f), (f \otimes f \otimes f), \ldots\}.
\]

If \( f \) is interpreted as an arrival curve, then any trace \( R \) that is upper bounded by \( f \) is also upper bounded by the sub-additive closure \( \bar{f} \).

A tight upper arrival curve should satisfy the sub-additive property.
Some Relations

- The output stream of a component satisfies:

\[ R'(t) \geq (R \otimes \beta^l)(t) \]

- The output upper arrival curve of a component satisfies

\[ \alpha^{u'} \leq (\alpha^u \otimes \beta^l) \]

with a simple and pessimistic calculation.

- The remaining lower service curve of a component satisfies

\[ \beta^{l''}(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta^l(\lambda) - \alpha^u(\lambda)) \]
Some Relations

- The output stream of a component satisfies:

\[ R'(t) \geq (R \otimes \beta^l)(t) \]

Proof:

\[
R'(t) = \inf_{0 \leq \lambda \leq t} \{ R(\lambda) + C(t) - C(\lambda) \} \\
\geq \inf_{0 \leq \lambda \leq t} \{ R(\lambda) + \beta^l(t - \lambda) \} = (R \otimes \beta^l)(t).
\]

- The output upper arrival curve of a component satisfies

\[ \alpha^{u'} \leq (\alpha^u \otimes \beta^l) \]

with a simple and pessimistic calculation.

- The remaining lower service curve of a component satisfies

\[
\beta^{l''}(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta^l(\lambda) - \alpha^u(\lambda))
\]
Remaining Service Curve

\[ \beta''(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta'(\lambda) - \alpha^u(\lambda)) \]

\[ C'(t) - C'(s) = \sup_{0 \leq a \leq t} \{ C(a) - R(a) \} - \sup_{0 \leq b \leq s} \{ C(b) - R(b) \} \]

\[ = \inf_{0 \leq b \leq s} \left\{ \sup_{0 \leq a \leq t} \{ C(a) - C(b) - (R(a) - R(b)) \} \right\} \]

\[ = \inf_{0 \leq b \leq s} \left\{ \sup_{0 \leq a - b \leq t - b} \{ C(a) - C(b) - (R(a) - R(b)) \} \right\} \]

\[ \geq \inf_{0 \leq b \leq s} \left\{ \sup_{0 \leq \lambda \leq t - b} \{ \beta'(\lambda) - \alpha^u(\lambda) \} \right\} \]

\[ \geq \sup_{0 \leq \lambda \leq t - s} \{ \beta'(\lambda) - \alpha^u(\lambda) \} = \sup_{0 \leq \lambda \leq \Delta} (\beta'(\lambda) - \alpha^u(\lambda)). \]
Tighter Bounds

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

Without formal proofs....
Graphical Interpretation

\begin{align*}
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)\}
\end{align*}
Proof of Buffer Size

Maximum buffer $B$

workload

interval length ($\Delta$)

\[ B(t) = R(t) - R'(t) \]

\[ = R(t) - \inf_{0 \leq u \leq t} \{ R(u) + C(t) - C(u) \} \]

\[ = \sup_{0 \leq u \leq t} \{ R(t) - R(u) - C(t) + C(u) \} \]

\[ \leq \sup_{0 \leq u \leq t} \{ \alpha^u(t - u) - \beta^l(t - u) \} \]

\[ \leq \sup_{0 \leq \lambda} \{ \alpha^u(\lambda) - \beta^l(\lambda) \} \]
System Composition

Concrete Instance

How to Interconnect service?

Scheduling

GPC

CPU

RM

TDMA

BUS DSP

GPC

Concrete Instance

GPC

How to Interconnect service?

Scheduling

GPC

GPC

GPC
Scheduling and Arbitration
Mixed Hierarchical Scheduling

\[ \text{TDMA} + \text{FP/RM} + \text{EDF} + \text{RR} \]

- TDMA
  - $\beta_{s1}$
  - $\beta_{s2}$
  - $\beta_{s3}$
  - $\alpha_A$ → $\alpha'_A$
  - $\alpha_B$ → $\alpha'_B$
  - $\alpha_C$ → $\alpha'_C$
  - $\alpha_D$ → $\alpha'_D$
  - $\alpha_F$ → $\alpha'_F$

- EDF
  - $\beta'$

- RR
  - $\beta'_{s1}$
  - $\beta'_{s2}$
  - $\beta'_{s3}$

- FP/RM
  - $\alpha_A$ → $\alpha'_A$
  - $\alpha_B$ → $\alpha'_B$
  - $\alpha_C$ → $\alpha'_C$
  - $\alpha_D$ → $\alpha'_D$
  - $\alpha_F$ → $\alpha'_F$

- EDF
  - $\beta'$

- FP
  - $\beta'$

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

Input Stream

Concrete Instance

Abstract Representation

\[ \alpha \rightarrow \text{GPC} \rightarrow \beta_{\text{CPU}} \rightarrow \beta_{\text{BUS}} \rightarrow \beta_{\text{DSP}} \rightarrow \alpha' \]
Extending the Framework

- New HW behavior
- New SW behavior
- New scheduling schemes
- New ...
Extending the Framework

- New HW behavior
- New SW behavior
- New scheduling schemes
- New \ldots \ldots
d

The hard part...

Find new relations

\[
\alpha'(\Delta) = f_\alpha(\alpha, \beta) \\
\beta'(\Delta) = f_\beta(\alpha, \beta)
\]
RTC Toolbox (http://www.mpa.ethz.ch/Rtctoolbox)

Modular Performance Analysis with Real-Time Calculus

Real-Time Calculus Toolbox

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.*