Deriving Monitoring Bounds for Distributed Real-Time Systems

Moritz Neukirchner, Steffen Stein, Rolf Ernst – TU Braunschweig
Ensuring Timing Constraints

To ensure real-time constraints systems are
- **verified** at design-time (e.g. RTC, SymTA/S)
Ensuring Timing Constraints

To ensure real-time constraints systems are

- verified at design-time (e.g. RTC, SymTA/S)
- monitored at run-time (key to efficient mixed-criticality [Baruah11])
- both based on model and formal specification

Monitoring According to Verification Model

**Monitoring** according to **verification model**
- Bounds are **safe**
- Bounds are fairly **efficient to derive** through performance analysis *e.g.* Real-Time Calculus, Compositional Performance Analysis
- overly **pessimistic**
- does **not** allow **worst acceptable** timing behavior

**Sensitivity Analysis** derives **maximum parameter** variation under which **constraints** still **hold**

**MONITORING** SHOULD BE PERFORMED ACCORDING TO **SENSITIVITY BOUNDS**
Outline

- Monitoring based on Sensitivity Analysis
- Compositional Sensitivity Analysis
- Evaluation
Outline

• Monitoring based on Sensitivity Analysis

• Compositional Sensitivity Analysis

• Evaluation
Search-based Sensitivity Analysis

- **Modify parameters** until constraints are violated
- **System-level Performance Analysis** (e.g. SymTA/S, MPA) as feasibility test
- Yields Sensitivity **Bounds at sources**
Search-based Sensitivity Analysis

- **Modify parameters** until constraints are violated
- **System-level** Performance **Analysis** (e.g. SymTA/S, MPA) as feasibility test
- Yields Sensitivity **Bounds at sources**

This approach yields **sensitivity bounds only at the sources**.
Interdependence of Sensitivity Bounds

- **Sensitivity Bounds** are **NOT independent** of each other
- Existing Analysis yield **entire pareto-front**

![Diagram showing the interdependence of sensitivity bounds between two processors and a bus.](image)
Interdependence of Sensitivity Bounds

- Sensitivity Bounds are **NOT independent** of each other.

Existing Analysis yield entire pareto-front.

For **monitoring** only one point of the **pareto front** can be applied.

- For monitoring only one point of the pareto front can be applied.
Outline

- Monitoring based on Sensitivity Analysis
- Compositional Sensitivity Analysis
- Evaluation
Compositional Sensitivity Analysis

- Perform **sensitivity analysis of resources** in isolation
Compositional Sensitivity Analysis

- Perform **sensitivity analysis of resources** in isolation (WCRT analysis)
- Resource gives **guarantee** on allowed **input jitter**
- **Guarantee** serves as constraint at other resource
- Execution as **distributed fixed point algorithm**
Compositional Sensitivity Analysis

- Perform **sensitivity analysis of resources** in isolation
- Resource gives **guarantee** on allowed **input jitter**
- Guarantee serves as **constraint** at other resource
- Execution as **distributed fixed point algorithm**

This is **Compositional Performance Analysis reversed!**
Isn’t this **trivial**?

**YES. NO.**
Problems in Compositional Sensitivity Analysis

Starting Value (see paper):
- Some tasks may **not** have valid guarantees when analyzed
- Cannot be resolved when cyclic dependencies exist
- **Conservative starting point** has to be defined

Convergence (see paper):
- Distributed fixed point algorithm
- Convergence has to be ensured

Pareto-Choice and Consistency:
- Each **local analysis** performs pareto choice
- **Local pareto choices** have to be **globally consistent**
Pareto Choice And Consistency

Consistent?

Pareto Choice

Processor

Guarantee

Constraint

Local Sensitivity Analysis

Processor

Guarantee

Constraint

Local Sensitivity Analysis

11 July 2012 | Moritz Neukirchner | Deriving Monitoring Bounds for Distributed Real-Time Systems | Slide 16
Consistency

• Assume initial guarantee/constraint assignment correct
• Local sensitivity analysis are increasing
  i.e. larger constraint at output → larger or equal guarantee at input
  → Tuple $G$ of all guarantees/constraints can only increase
  → Increasing a single guarantee/constraint cannot violate constraint
Pareto Choice

- **Execution order** of greedy local analyses determines *pareto choice*
- Guarantee, that is *analyzed first*, is *maximized*

Possible exploitation:
- Analyze *low criticality tasks first*
→ *Low criticality tasks can accommodate largest design uncertainty*

\[
G_1(G_2) \leq G_1 \quad G_2(G_1) \leq G_2
\]
Pareto Choice and Consistency (Summary)

- **Consistency/Correctness** of guarantees formally **proven**
  
  *Theorems 2 & 5 in the paper*

- All guarantee assignments (**sensitivity bounds**) **conservative**
  
  - All **intermediate** guarantee assignments are **conservative**
  
  → Algorithm can be **stopped at any time** and **results** are valid

- **Correctness** holds for **any execution order** of local sensitivity analyses

- **Local sensitivity analyses** can be (partly) performed **in parallel**

- **Execution Order** of Local Analyses **determines pareto choice**
  
  *e.g. analyze **low criticality** applications **first** to allow for largest uncertainty*
Outline

• Monitoring based on Sensitivity Analysis

• Compositional Sensitivity Analysis

• Evaluation
Evaluation

- **Our algorithm** yields **one approximated n-dimensional** pareto point
- **Existing** system-level sensitivity **analyses** [8,18] yield **pareto front** of **up to 4 dimensions** (in reasonable runtimes)
- [8,18] build on the **same performance analysis** algorithms

- Comparison of **solution quality** for systems up to 4 dimensions  
  *i.e. where comparison to exact solution is possible*
- Evaluation of **runtime**  
  *in terms of required WCRT analyses*

---


Test Setup

- **Synthetic testcases** generated with System Models for Free (SMFF)
  
  *see paper for complete parameter set*

- Key characteristics of testcases:
  
  5 processors + 2 busses
  
  **Utilization 35%-45%** (UUnifast)

  - Small systems: 4x chain of 3 tasks = 12 tasks,
    2-8 comm. tasks
    4 dimensions

  - Large systems: 50x chain of 3 tasks = 150 tasks,
    52-79 comm. tasks
    50 dimensions
Solution Quality

- Solution quality: Normalized Manhattan distance to closest comparable pareto point

![Graph showing the relation between sensitivity bound quality (q) and relative frequency](image)

- exact solution
- no guarantee
## Runtime

- **Existing analyses** require $\sim 10^4$ [8] and $\sim 10^8$ [18] WCRT analyses to derive entire pareto front (4 sources/dimensions)

### Our Approach

![Histograms showing the distribution of local performance analyses for 4 sources and 50 sources.](image)

4 sources

50 sources


Conclusion

- **Monitoring** should be performed **according to sensitivity bounds**

- **Existing** system-level sensitivity **analyses** yield
  - entire pareto front of bounds at sources

- We have introduced **Compositional Sensitivity Analysis**
  - yields sensitivity **bounds at every resource**
  - yields **one multi-dimensional** sensitivity bound

- Analysis **significantly faster** than previous approaches
- **Accuracy comparable**

*Thank you for your attention.*