Elastic Mixed-Criticality Parallel Real-Time Scheduling for Real-Time Hybrid Simulation

Joint ETH-MECHS Workshop: New Frontiers and Innovative Methods for Real-Time Hybrid Simulation, at ETH Zurich, March 13-15, 2019

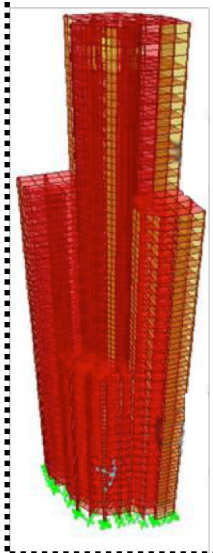
Chris Gill
Professor of Computer Science and Engineering
Washington University in St. Louis, MO, USA
cdgill@cse.wustl.edu

State of the Art in Real-Time Hybrid Simulation

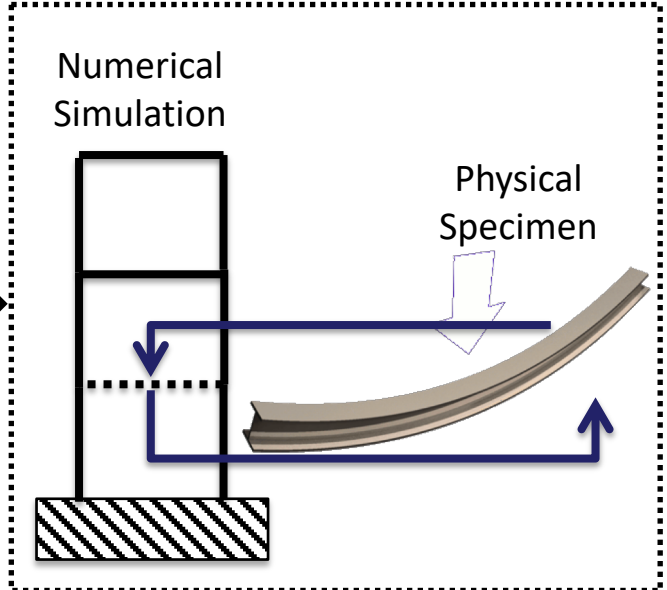
Finite element model **simulation**, integrated **safely** with control and sensing and actuation of **physical components**, scheduled with **real-time** guarantees atop a **multi-core server**

Physical Test

Computational Simulation

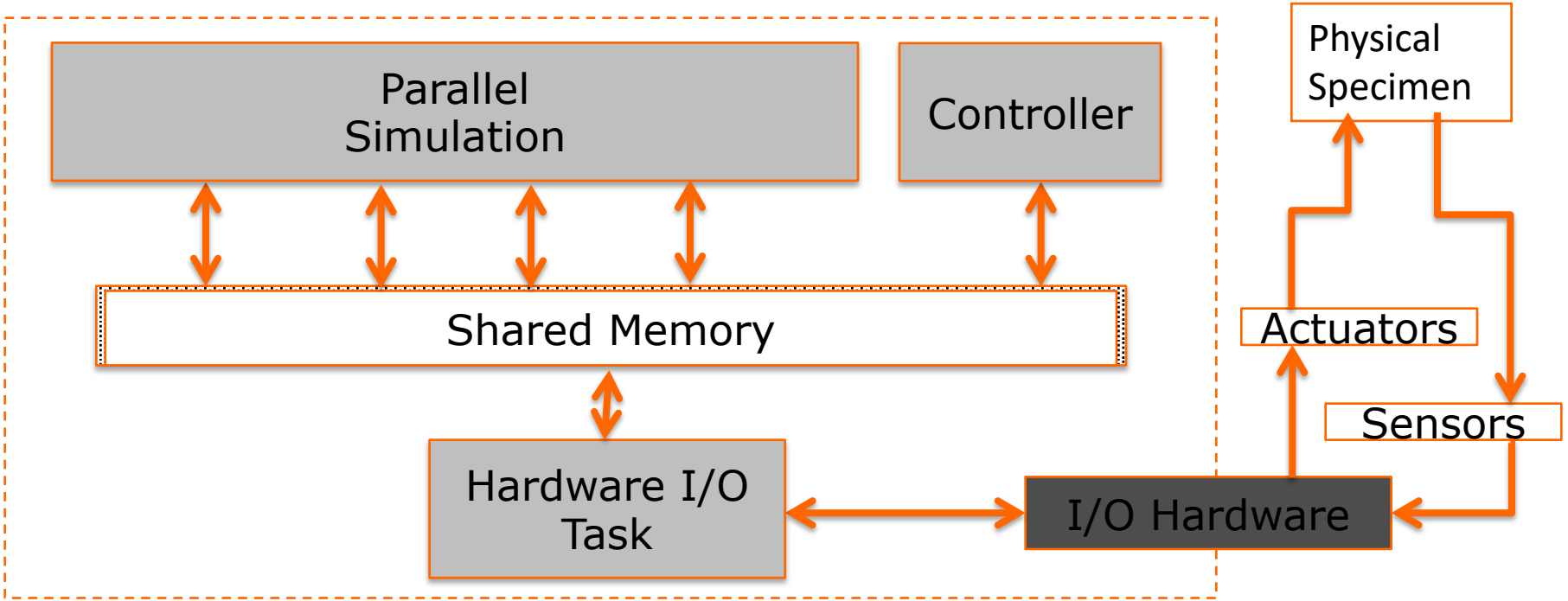


Real-Time Hybrid Simulation



CyberMech Parallel RTHS Architecture

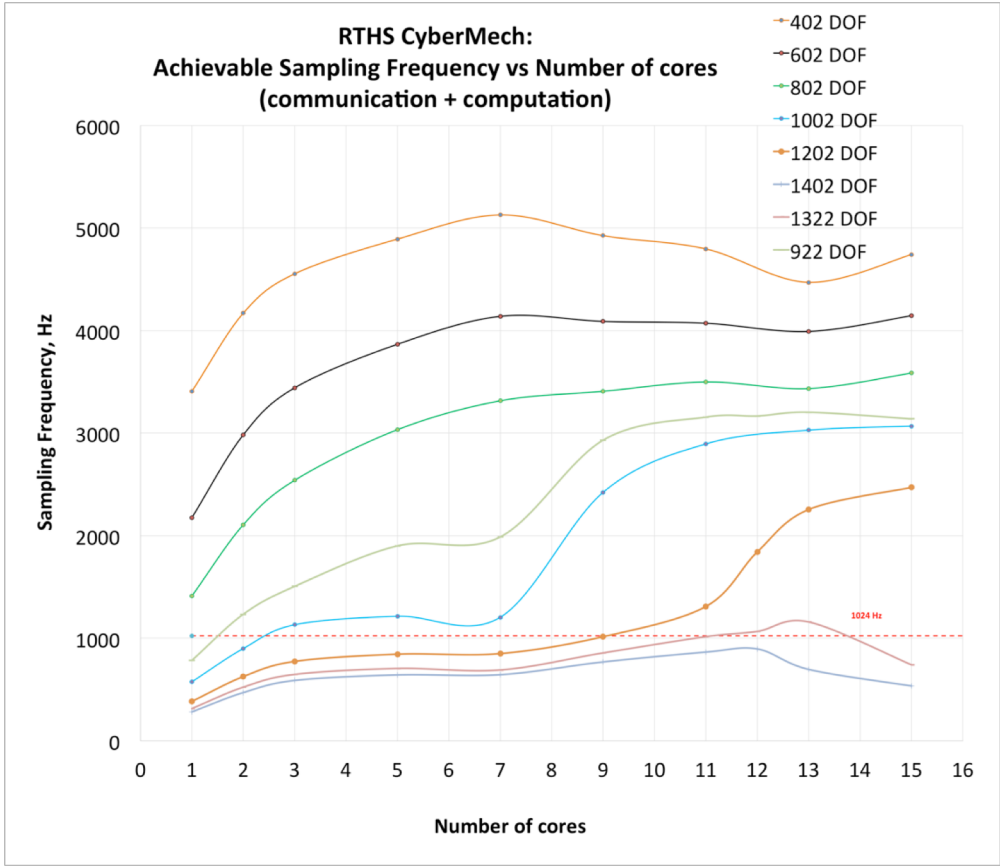
Federated scheduling in OpenMP/Cilk Plus, safe multithreading removes I/O bottlenecks



D. Ferry, G. Bunting, A. Maghareh, S. Dyke, A. Prakash, K. Agrawal, C. Gill, C. Lu, "Real-time System Support for Hybrid Structural Simulation," EMSOFT 2014

Scalability of Parallel RTHS in CyberMech

- Realistic experiments
  - » Using hydraulic actuators at Purdue University Bowen Lab
- First ever fully parallel RTHS
  - » Experiments with up to 1322 degree-of-freedom simulations on 13 (of 16) cores at 1024Hz
  - » Conducted in October 2017
- Baseline for ongoing work
  - » Beyond ***linear*** simulations, control, compensation, etc.

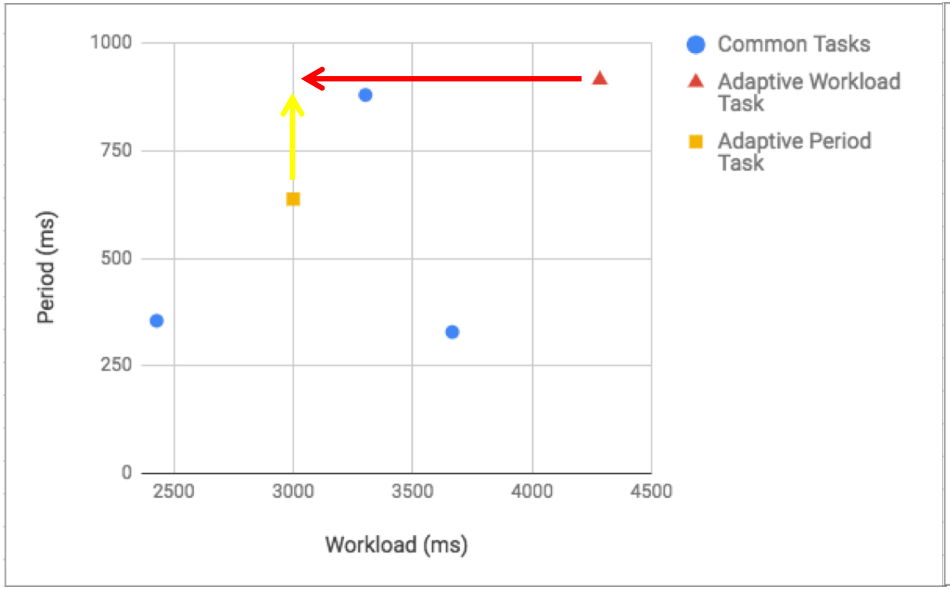


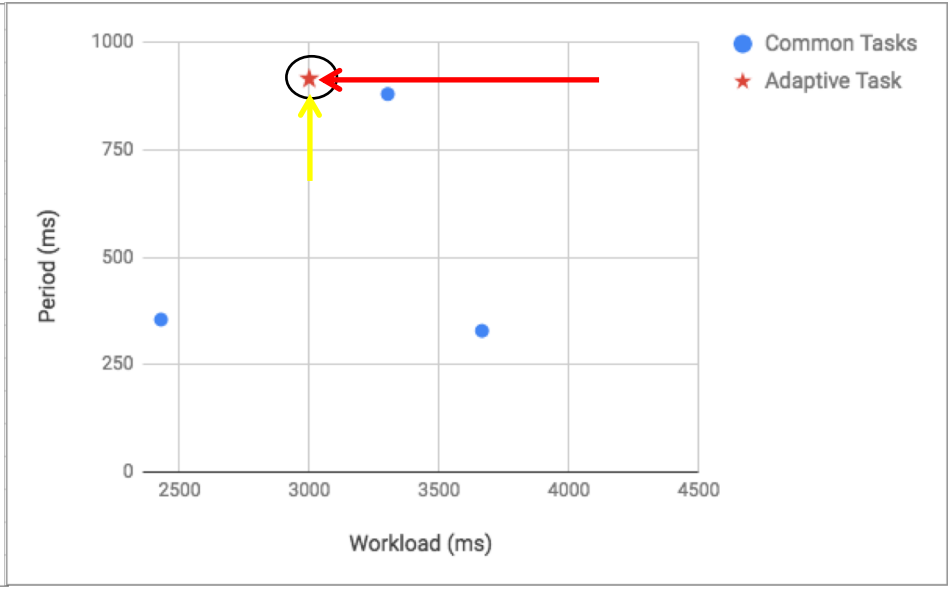
Challenges given the Current State of the Art

- What if RTHS component changes rate or computation?
  - E.g., moving from linear control to non-linear control on-line
  - E.g., part of FEM model can (by design) go non-linear on-line
- Can some components be more critical than others?
  - E.g., control of physical components of an RTHS vs. linear FEM components whose error increases as its utilization drops
  - E.g., how far can a linear FEM component's performance degrade without violating the integrity of the RTHS experiment?
- Can components trade-off time scales / computation?
  - With strong assurance from real-time scheduling theory?

Allowing Components to Change Work vs. Period

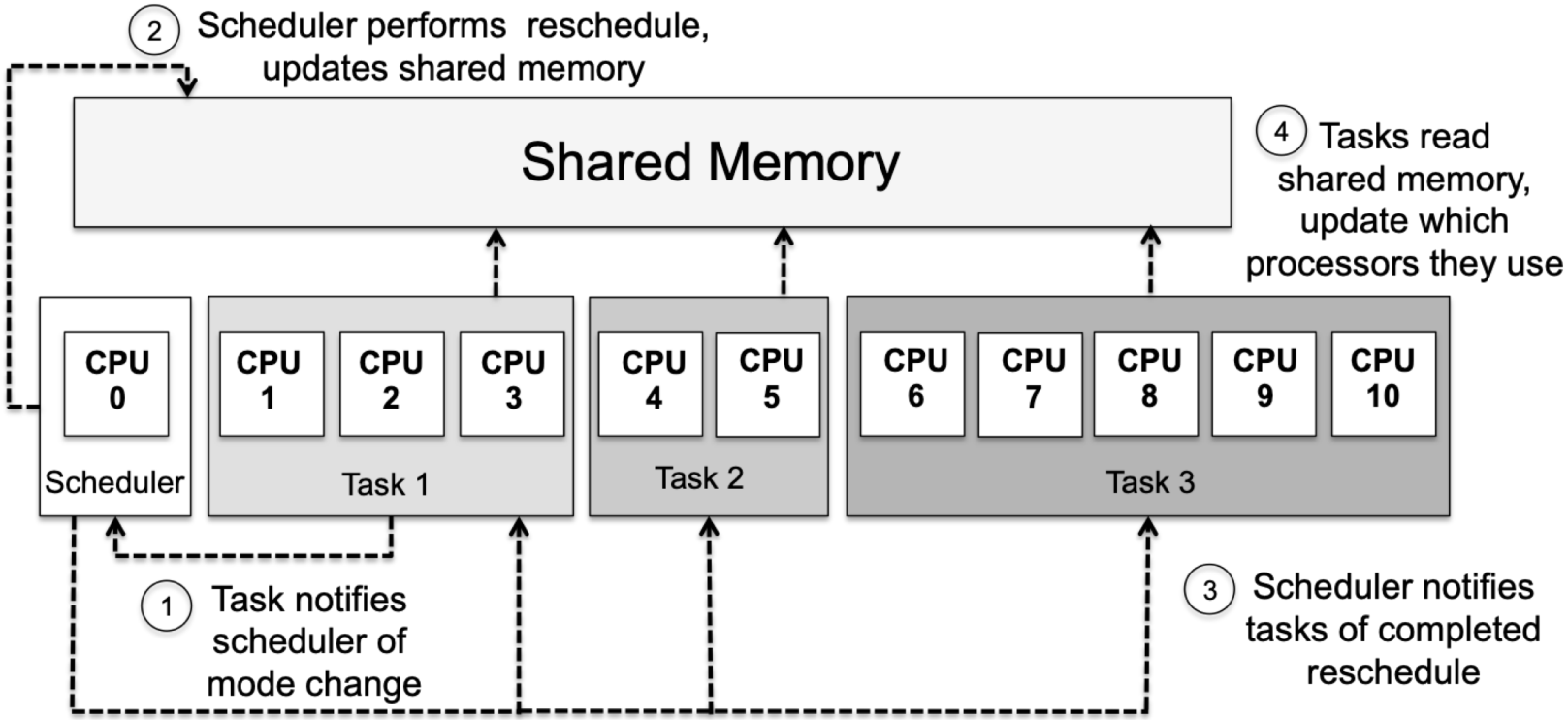
- Still with strong assurance of timing feasibility





J. Orr, C. Gill, K. Agrawal, S. Baruah, C. Cianfarani, P. Ang, and C.Wong, "Elasticity of Periods and Workloads of Parallel Real-Time Tasks," RTNS 2018

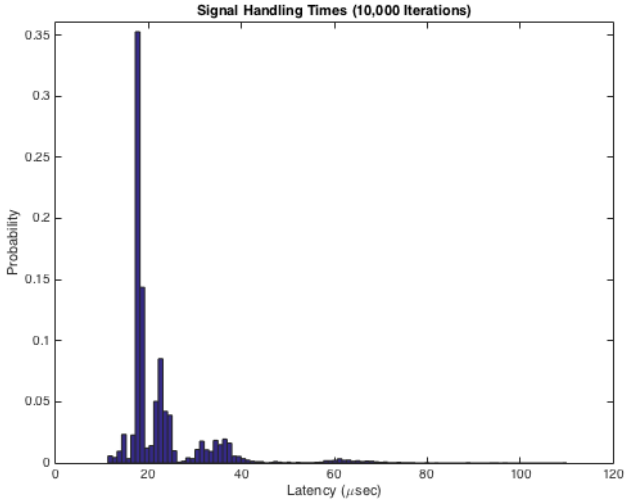
Giving Components More or Fewer Cores On-Line

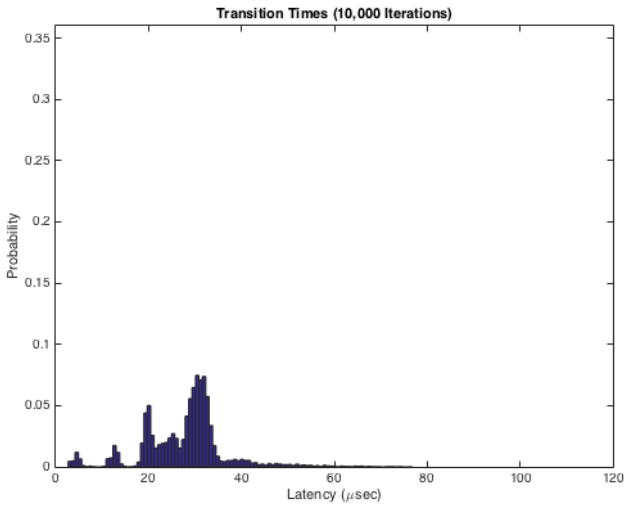


J. Orr, C. Gill, K. Agrawal, S. Baruah, C. Cianfarani, P. Ang, and C.Wong, "Elasticity of Periods and Workloads of Parallel Real-Time Tasks," RTNS 2018

Reasonable Overheads for 1024 Hz (RTNS '18)

- Task notification via POSIX RT signals
  - » Ranged from 11.23 µsec to 110.03 µsec, often around 18 µsec
- Thread priority change (and possible core migration)
  - » Ranged from 2.67 µsec to 76.77 µsec, often around 30 µsec





A Small Dose of Real-Time Scheduling Theory

$$\textbf{minimize} \sum_{i=1}^{n} \frac{1}{E_i} \left(U_i^{(max)} - U_i \right)^2$$

$$\textbf{such that } \forall_i \left(U_i^{(min)} \leq U_i \leq U_i^{(max)} \right) \wedge m \geq \sum_{i=1}^{n} \left\lceil \frac{C_i - L_i}{T_i - L_i} \right\rceil$$

- Builds on prior research by Chantem et al. (RTSS 2006)
  - » New utilization definition for parallel real-time tasks on *m* cores
  - » Allows *either* **period** or **work** to vary elastically

J. Orr, C. Gill, K. Agrawal, S. Baruah, C. Cianfarani, P. Ang, and C.Wong, "Elasticity of Periods and Workloads of Parallel Real-Time Tasks," RTNS 2018

Lower-Criticality Tasks *Decrease* # Cores Used

Task	E_i	Z_i	$m_i[0]$	$m_i[1]$	$m_i[2]$
τ_1	5	0	8	*6*	*4*
τ_2	4	0	6	*4*	*2*
τ_3	3	1	4	**6**	*2*
τ_4	2	1	4	**6**	*2*
τ_5	1	2	2	2	**6**

$$U_i[j] = \frac{C_i[j]}{D_i[j]}$$

$$m_i[j] = \left\lceil \frac{C_i[j] - L_i[j]}{D_i[j] - L_i[j]} \right\rceil$$

degraded

overload

nominal

- Can specify nominal, overload, and degraded utilizations per task
- Only highest criticality overloads, only lowest degrades
- Some minimum utilization is supported even at lowest criticality level

C. Gill, J. Orr, S. Harris, "Supporting Graceful Degradation Through Elasticity in Mixed-Criticality Scheduling," WMC at RTSS 2018

An Appeal to the Hybrid Simulation Community

- Please suggest new examples this approach may help
  - Other kinds of earthquake engineering experiments
  - Experiments involving other kinds of natural hazards
  - Especially, experiments with significant ***on-line*** variation in computational demands and/or necessary time-scales
- Please suggest how we can improve this approach
  - Is speed of sound (~1KHz) sufficient? What specific advantages would supporting say 10KHz offer vs. what we can do currently?
  - Are there other theoretical or practical considerations that may improve the applicability or relevance of our approach?

Thanks!

Washington University in St.Louis

James Orr
Steven Harris
Abu Saifullah (Wayne State)
David Ferry (St. Louis U)
Jing Li (NJIT)
Chris Gill
Kunal Agrawal
Sanjoy Baruah
Chenyang Lu

PURDUE UNIVERSITY

Johnny Condori
Hugo Esquivel
Amin Maghareh
Greg Bunting (Sandia)
Arun Prakash
Shirley Dyke

BROWN UNIVERSITY

Christian Cianfarani
Christopher Wong

TEXAS
The University of Texas at Austin

Phyllis Ang

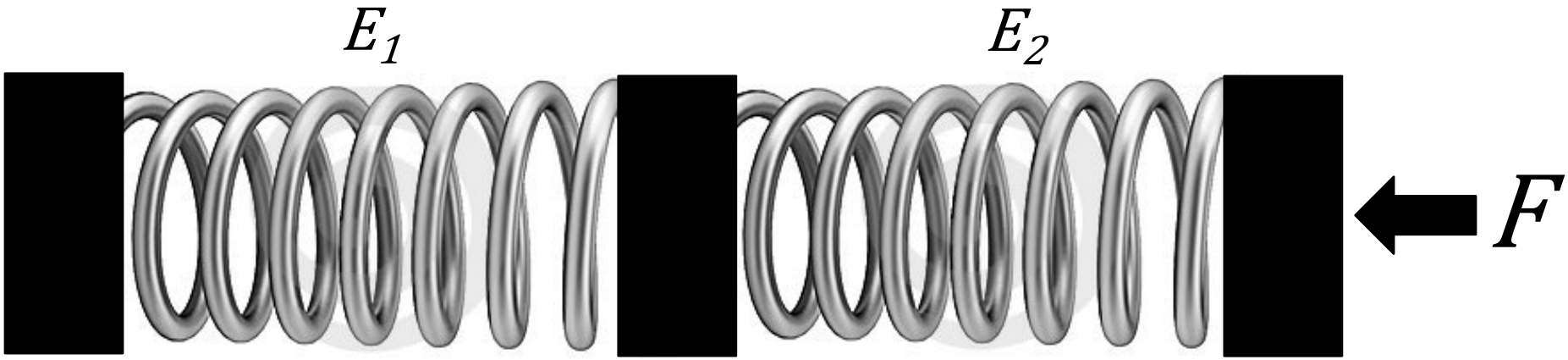
NSF

Research on the CyberMech platform has been supported in part by NSF grants CCF-1136073 (CPS) and CCF-1337218 (XPS)

Backup Slides

Backup Slide: Original Elastic Scheduling Model

- Buttazzo et al. introduced the elastic scheduling model
  - » Increases tasks' periods to compresses utilizations (RTSS '98)
  - » Analogous to elastic compression of physical springs (statics



  - » Model was also extended to consider blocking terms for critical sections accessed via the Stack Resource Policy (IEEE ToC '02)

Backup Slide: Constrained Optimization Version

- Chantem et al. defined this as an optimization problem
  - » Minimize a weighted sum of squares of the differences between the chosen utilization for each task and its maximum utilization
  - » Subject to utilizations from minimum and maximum values and the sum not exceeding the available utilization (RTSS '06)

$$\textbf{minimize} \sum_{i=1}^{n} \frac{1}{E_i} \left(U_i^{(max)} - U_i \right)^2$$

$$\textbf{such that } \forall_i \left(U_i^{(min)} \leq U_i \leq U_i^{(max)} \right) \wedge U_d \geq \sum_{i=1}^{n} U_i$$

Backup Slide: Mixed-Criticality with Elasticity

$$\textbf{minimize} \sum_{\forall i | Z_i < j} \frac{1}{E_i} (U_i[\alpha_i] - U_i)^2$$

(**criticality <u>insensitive</u> compression of lower criticality tasks**, α_i is Z_i if task overruns, 0 if not)

$$\textbf{such that } \forall_i \, (U_i[Z^{MAX}] \leq U_i \leq U_i[\alpha_i]) \wedge m \geq \sum_{k=1}^{n} m_k[j]$$

- Updates RTNS 2018 optimization (can vary D_i ***<u>or</u>*** C_i),
  Only compresses tasks whose criticality Z_i is **below** current level **j**
  - » Still checks whether **all** tasks are schedulable on m cores

C. Gill, J. Orr, S. Harris, "Supporting Graceful Degradation Through Elasticity in Mixed-Criticality Scheduling," WMC at RTSS 2018

Backup Slide: Cut (Utilization) to Fit

- Can run same amount of work less frequently
  - » Orr et al. RTNS ’18 (linear FEM component of RTHS)
  - » Su et al., DATE ‘13, RTCSA ’14 (alternative periods)
- Can run less work in same amount of time
  - » Liu et al. RTSS 2016; Huang et al. RTNS ’18 (mprcs cmptng)
  - » Anytime algorithms more generally (declare victory and retreat)
- Exploiting both forms of tailoring at once?
  - » Need to define carefully ***how much*** to modify work vs. rate
  - » E.g., run an ε-less-precise calculation λ-slower (tune ε vs. λ)

Backup Slide: Can Size RTHS to Available Cores

- Upper bound (overprovisioning)
  - » Sum of max cores each task needs: $\sum_{i=1}^{n} m_i [Z_i]$
  - » All tasks ***could*** run in overload
- Sufficient cores available (design target) $\max_j \left(\sum_{i=1}^{n} m_i [j] \right)$
  - » Tasks utilizations can be met at each level:
- Lower bound (underprovisioning)
  - » Below this tasks are dropped at ***every*** level: $\min_j \left(\sum_{i=1}^{n} m_i [j] \right)$

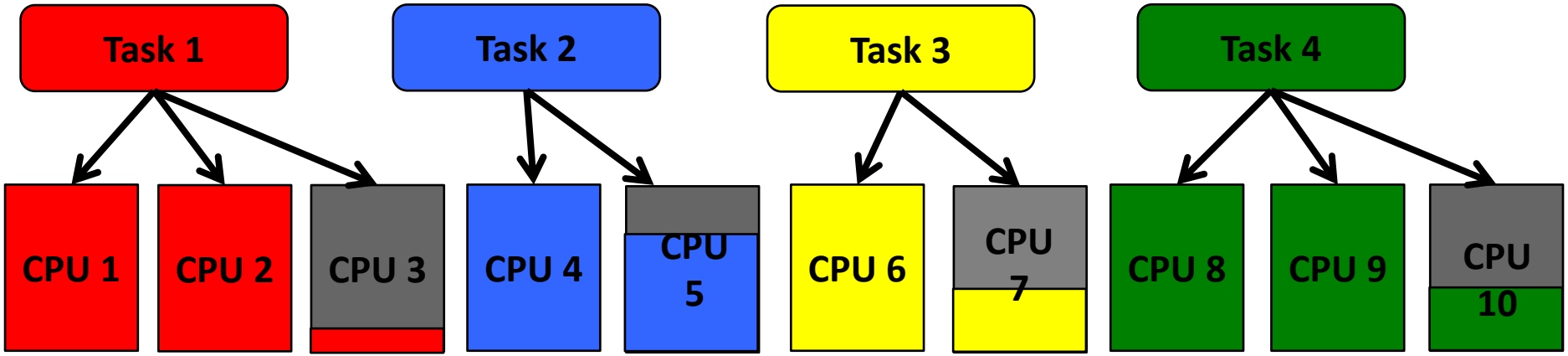
C. Gill, J. Orr, S. Harris, "Supporting Graceful Degradation Through Elasticity in Mixed-Criticality Scheduling," WMC at RTSS 2018

Backup Slide: Next Steps as Future Work

- Integration within next-generation RTHS experiments
  - » Moving between linear and non-linear control in different modes
  - » Adjusting resource utilizations rigorously/adaptively at run-time
- Removing current restrictions on the model
  - » Allow any task to be computationally ***and*** temporally elastic
  - » Allow a task to vary its utilization between < 1 to > 1 cores
- Adapting criticality level downward as soon as possible
  - » May reduce amounts and durations of degradation
  - » Need to consider how virtual deadlines are (re-)assigned

Backup Slide: Federated Scheduling Model

- In general a parallel task requires $\frac{C_i - L_i}{D_i - L_i} = \mathbf{A_i} + \boldsymbol{\varepsilon_i}$ (A_i is integer, $0 \le \varepsilon_i < 1$) CPUs to guarantee completion
- Federated scheduling allocates $\left\lceil \frac{C_i - L_i}{D_i - L_i} \right\rceil = \begin{cases} A_i & \varepsilon_i = 0 \\ 1 + A_i & \varepsilon_i > 0 \end{cases}$ CPUs



J. Li et al., "Analysis of Federated and Global Scheduling for Parallel Real-Time Tasks," 2014 26th Euromicro Conference on Real-Time Systems, Madrid, 2014, pp. 85-96.