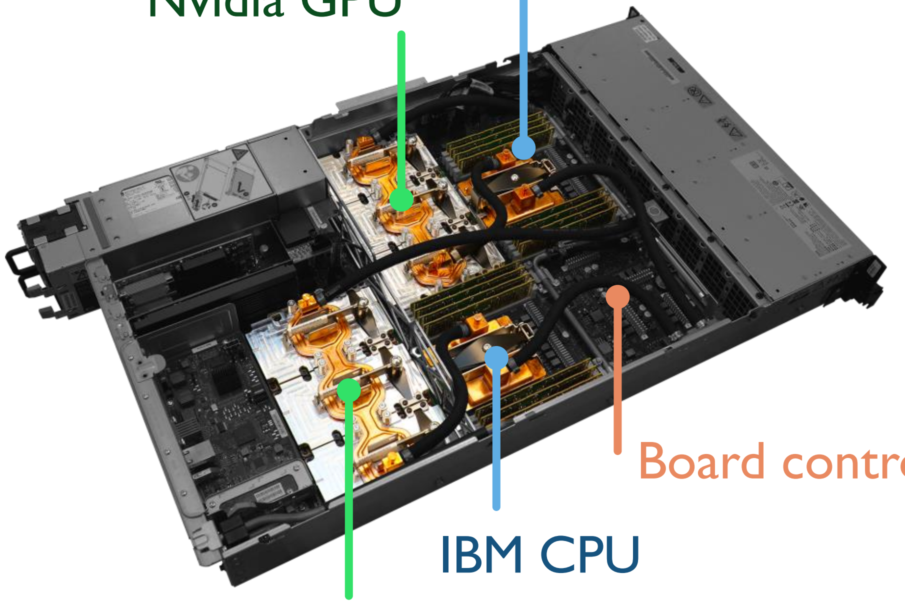
Tangram: Integrated Control of Heterogeneous Computers

Raghavendra Pradyumna Pothukuchi, Joseph L. Greathouse, Karthik Rao, Christopher Erb, Leonardo Piga, Petros Voulgaris, Josep Torrellas

ILLINOIS AMD

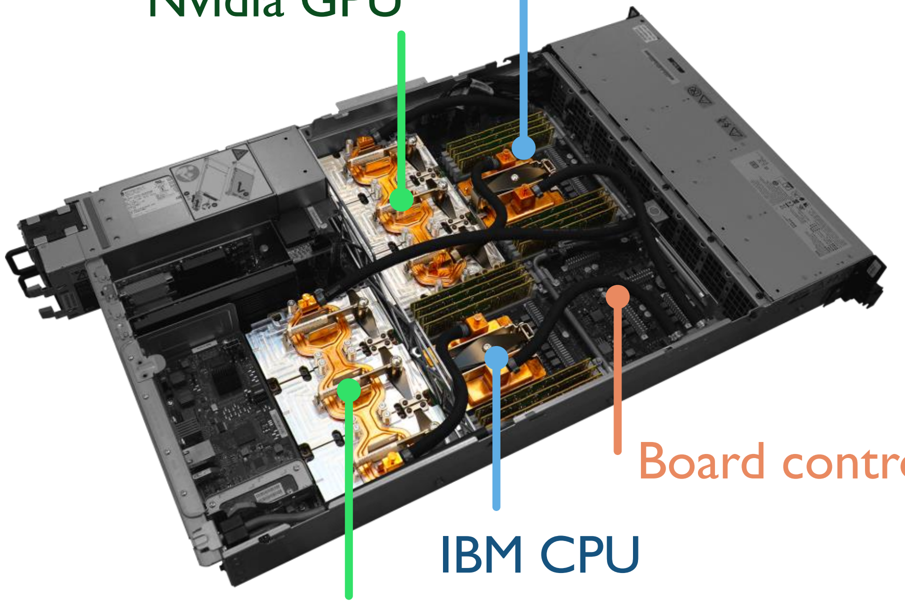
International Symposium on Microarchitecture (MICRO), October 2019

Heterogeneous Systems

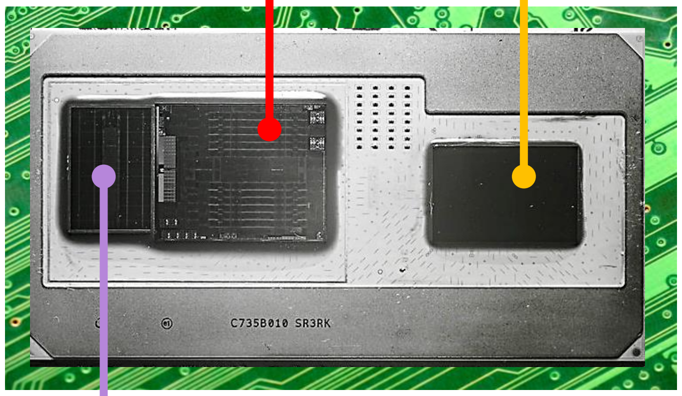


Discrete integration
(Image courtesy: NVIDIA)

Heterogeneous Systems

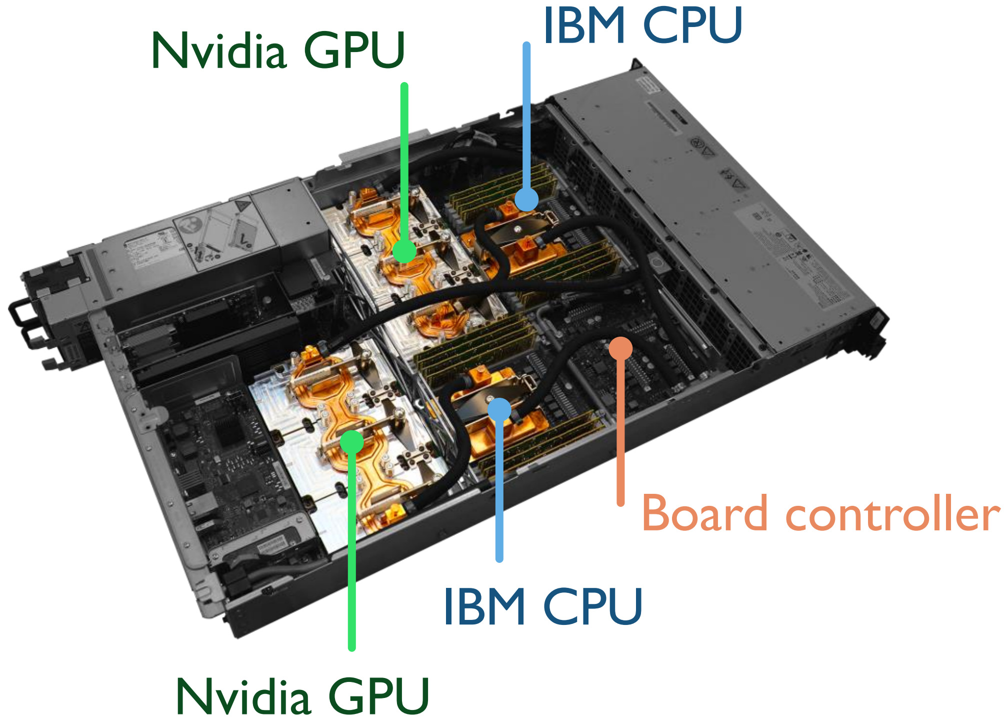


Discrete integration
(Image courtesy: NVIDIA)

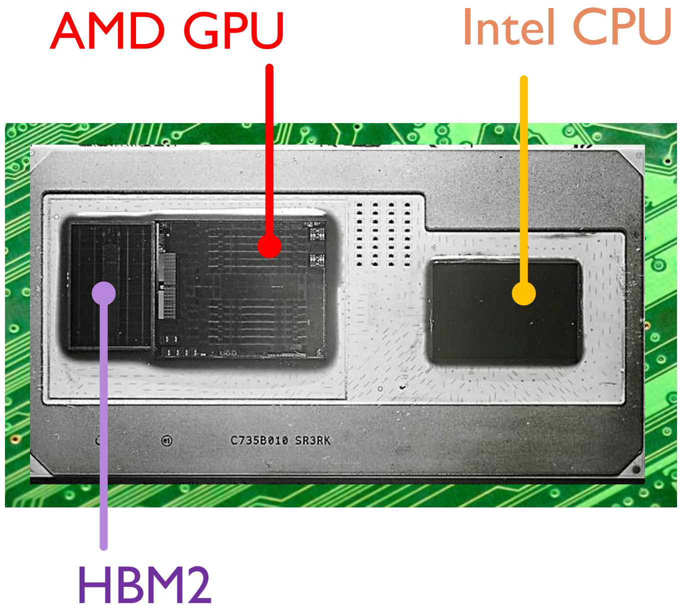


On-package integration

Heterogeneous Systems



Discrete integration
(Image courtesy: NVIDIA)



On-package integration

Challenge: System-wide efficiency with distributed resource control

State-of-the-Art Resource Controllers

Run as firmware on embedded microcontrollers

State-of-the-Art Resource Controllers

Run as firmware on embedded microcontrollers

Limitation 1: Many heuristic algorithms

Conflicting, and hard to reuse

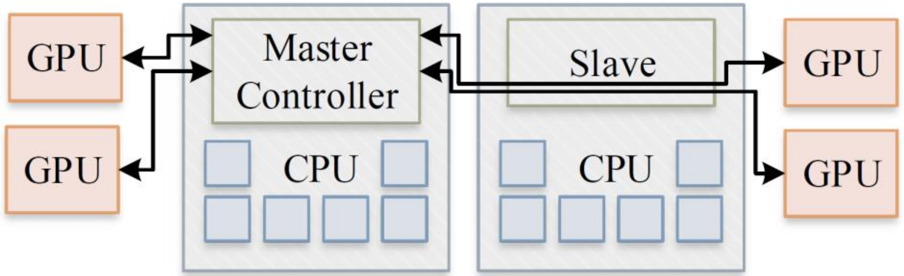
State-of-the-Art Resource Controllers

Run as firmware on embedded microcontrollers

Limitation 1: Many heuristic algorithms

Conflicting, and hard to reuse

Limitation 2: Centralized control



Slow, ineffective because of non-modularity. Hard to reuse.

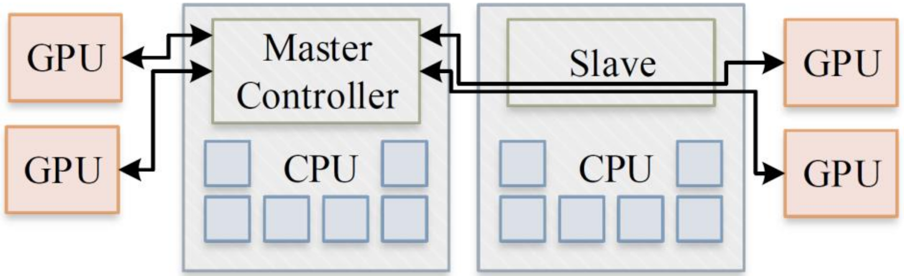
State-of-the-Art Resource Controllers

Run as firmware on embedded microcontrollers

Limitation 1: Many heuristic algorithms

Conflicting, and hard to reuse

Limitation 2: Centralized control



Slow, ineffective because of non-modularity. Hard to reuse.

Why is the state of the art so limited?

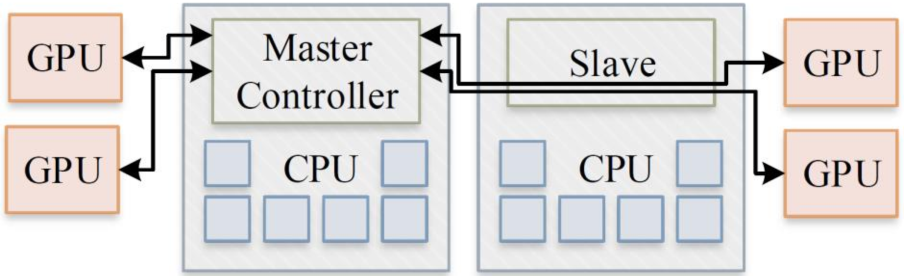
State-of-the-Art Resource Controllers

Run as firmware on embedded microcontrollers

Limitation 1: Many heuristic algorithms

Conflicting, and hard to reuse

Limitation 2: Centralized control



Slow, ineffective because of non-modularity. Hard to reuse.

Why is the state of the art so limited?

1. Distributed control is difficult
2. Heuristics appear intuitive

Contributions

- Tangram: a distributed control framework that is modular, fast, and globally coordinated

Contributions

- Tangram: a distributed control framework that is modular, fast, and globally coordinated

- A novel controller design that uses formal control, and with a standard interface

Contributions

- Tangram: a distributed control framework that is modular, fast, and globally coordinated
- A novel controller design that uses formal control, and with a standard interface
- Prototype on a multi-vendor heterogeneous server

Classifying Resource Control Goals

Safety

thermals, voltage droops

Immediate response
for protection

Classifying Resource Control Goals

Safety

thermals, voltage droops

Immediate response
for protection

Enhancement

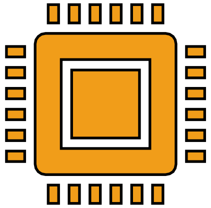
minimize Energy×Delay

Requires search

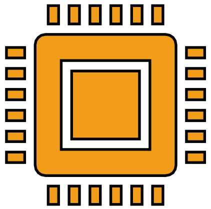
Classifying Resource Control Goals

Safety	Preconfigured	Enhancement
thermals, voltage droops	optimal for common patterns	minimize Energy×Delay
Immediate response for protection	Immediate response for optimality	Requires search

Enhancement: Robust Control


CPU Chip

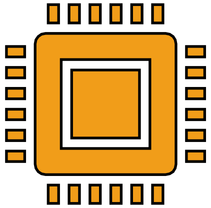
Enhancement: Robust Control



Enhancement: Robust Control

Inputs

DVFS
#cores



CPU Chip

Outputs

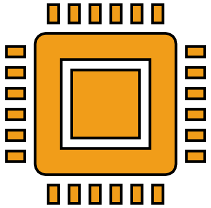
Performance
Power

Enhancement: Robust Control

Targets

Inputs

DVFS
#cores

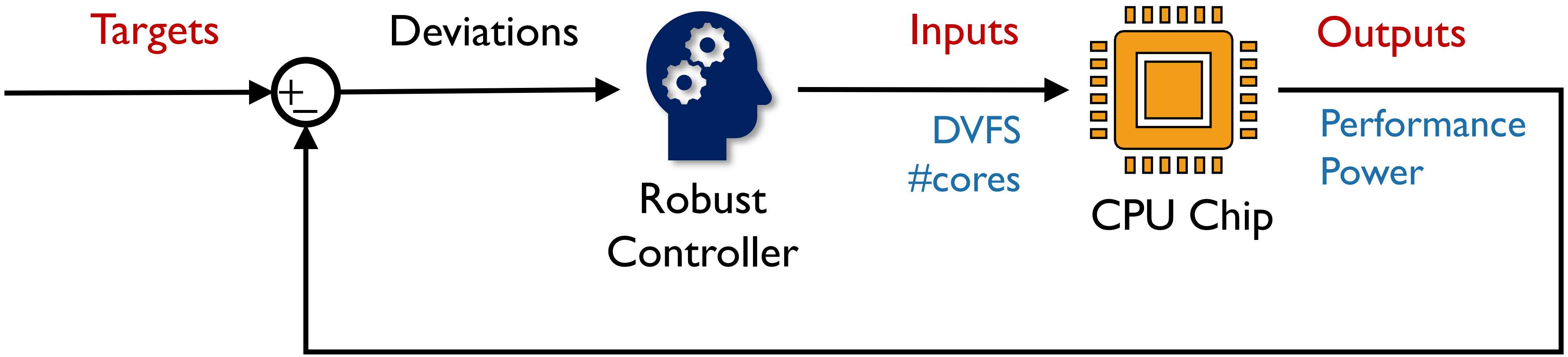


CPU Chip

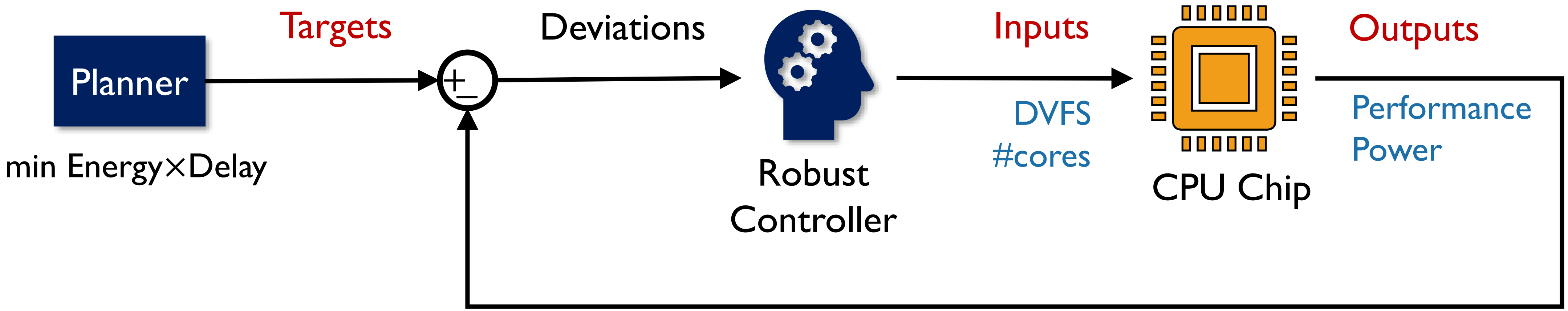
Outputs

Performance
Power

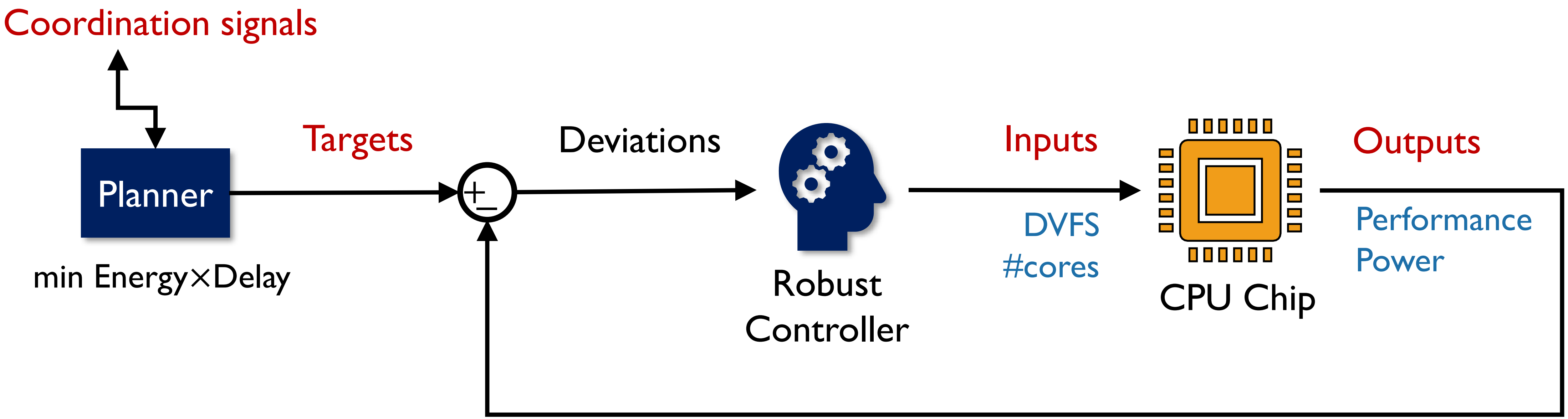
Enhancement: Robust Control



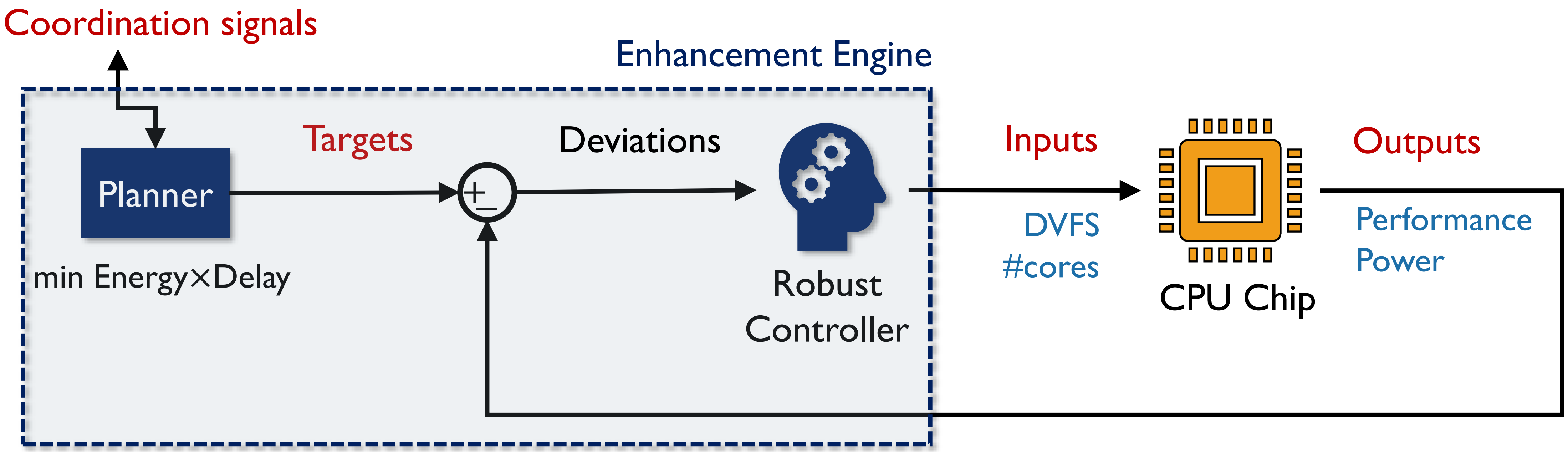
Enhancement: Robust Control



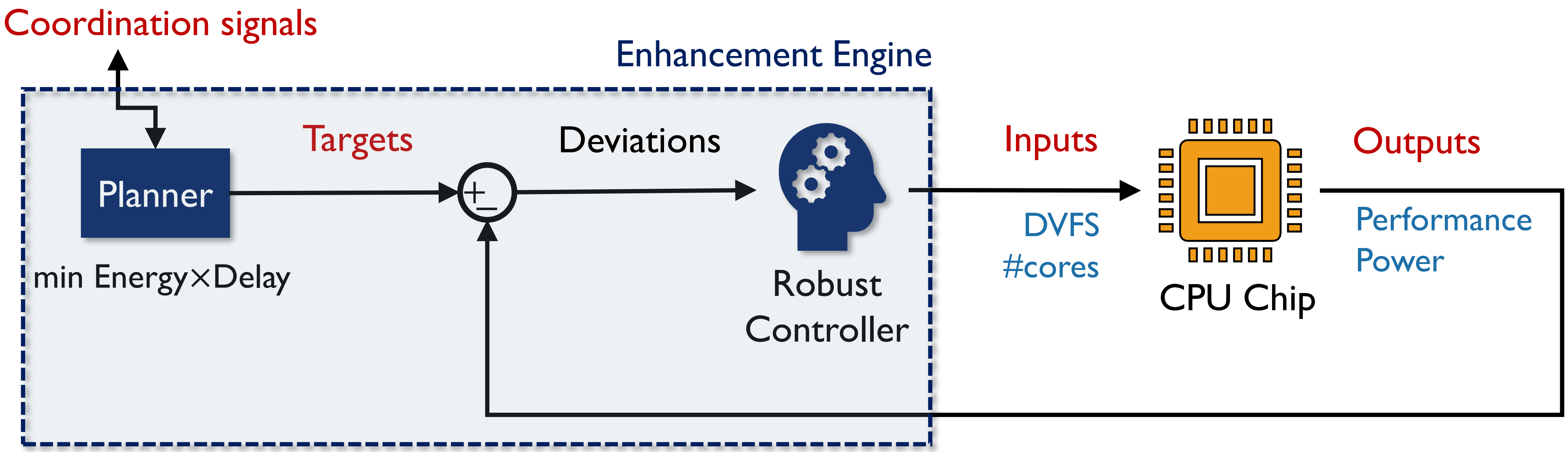
Enhancement: Robust Control



Enhancement: Robust Control

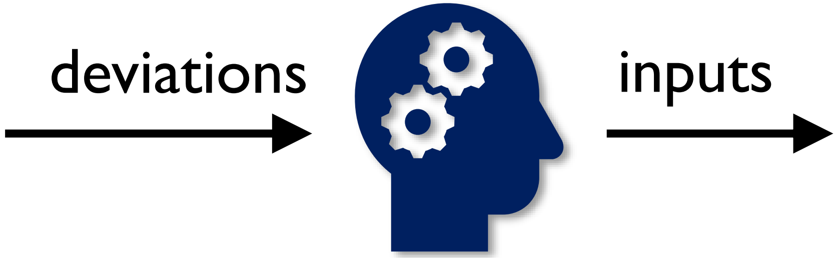


Enhancement: Robust Control

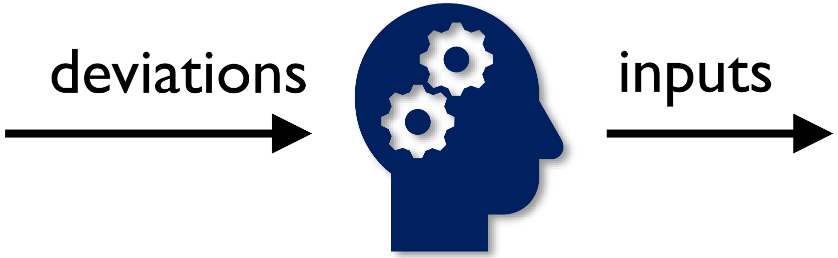


Formal, coordinated optimization replaces piecemeal heuristics

Robust Controller Operation

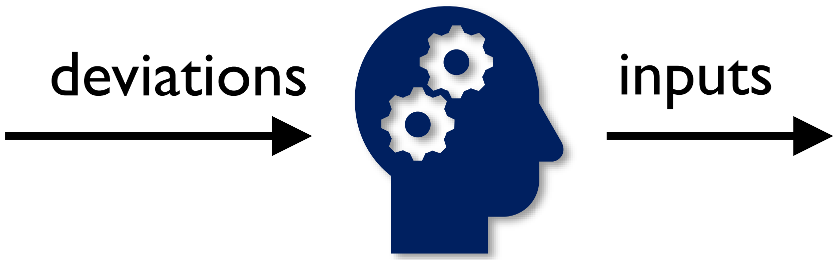


Robust Controller Operation



Matrices [**A, B, C, D**], and *state*

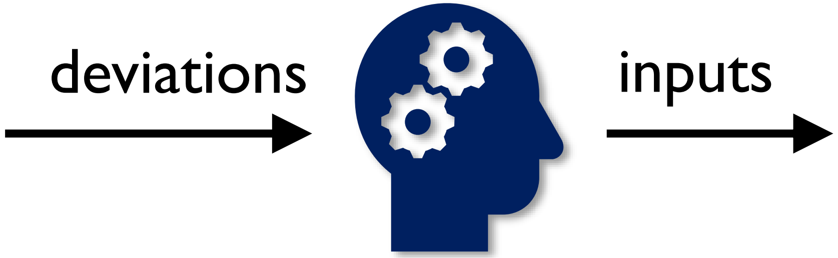
Robust Controller Operation



Matrices [**A, B, C, D**], and *state*

$$\text{inputs} = \mathbf{C} \times \text{state} + \mathbf{D} \times \text{deviations}$$

Robust Controller Operation

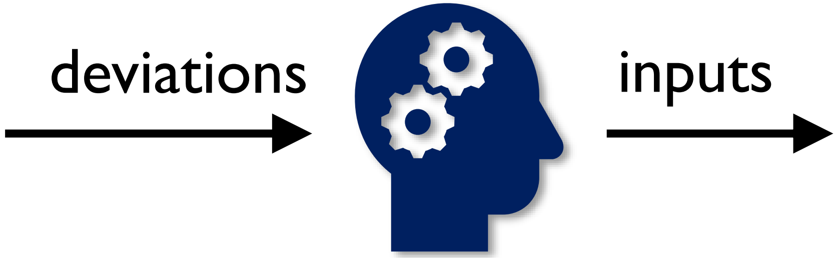


Matrices [**A, B, C, D**], and *state*

$$\text{inputs} = \mathbf{C} \times \text{state} + \mathbf{D} \times \text{deviations}$$

$$\text{state}_{\text{new}} = \mathbf{A} \times \text{state} + \mathbf{B} \times \text{deviations}$$

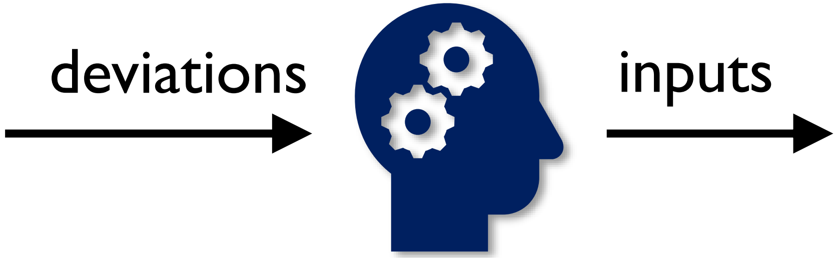
Robust Controller Operation



Matrices [**A, B, C, D**], and *state*

$$\left.\begin{aligned} \text{inputs} &= \mathbf{C} \times \text{state} + \mathbf{D} \times \text{deviations} \\ \text{state}_{\text{new}} &= \mathbf{A} \times \text{state} + \mathbf{B} \times \text{deviations} \end{aligned}\right\} \text{Every T time-units}$$

Robust Controller Operation

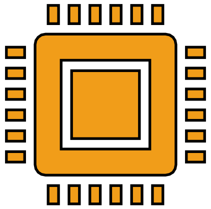


Matrices [**A, B, C, D**], and *state*

$$\left.\begin{aligned} \text{inputs} &= \mathbf{C} \times \text{state} + \mathbf{D} \times \text{deviations} \\ \text{state}_{\text{new}} &= \mathbf{A} \times \text{state} + \mathbf{B} \times \text{deviations} \end{aligned}\right\} \text{Every T time-units}$$

Deviations are small even when runtime conditions differ from design conditions

Proposed Controller



Proposed Controller

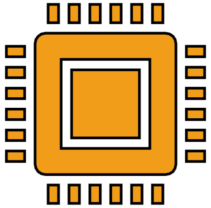
Proposed Controller

Coordination signals

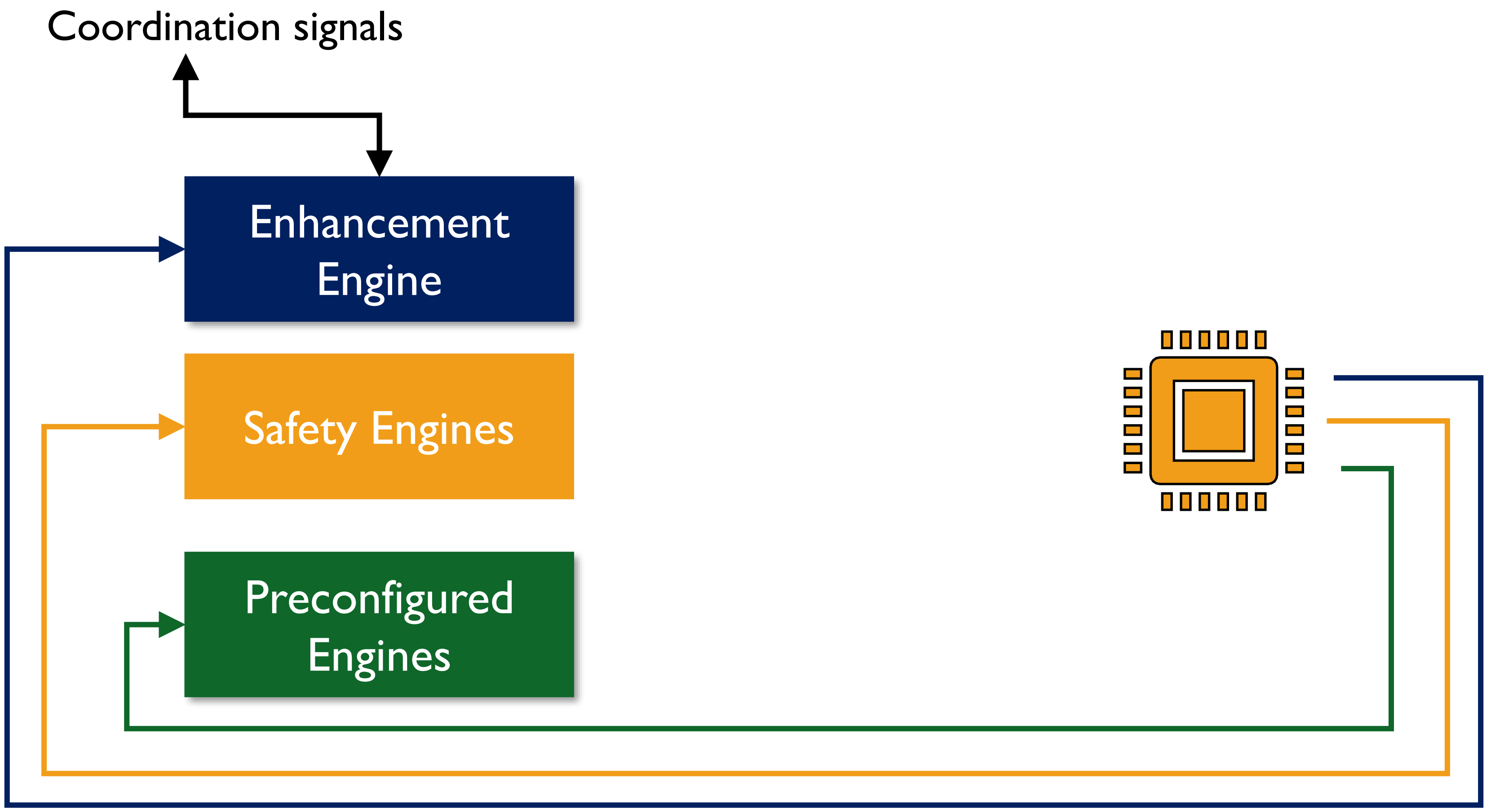
Enhancement Engine

Safety Engines

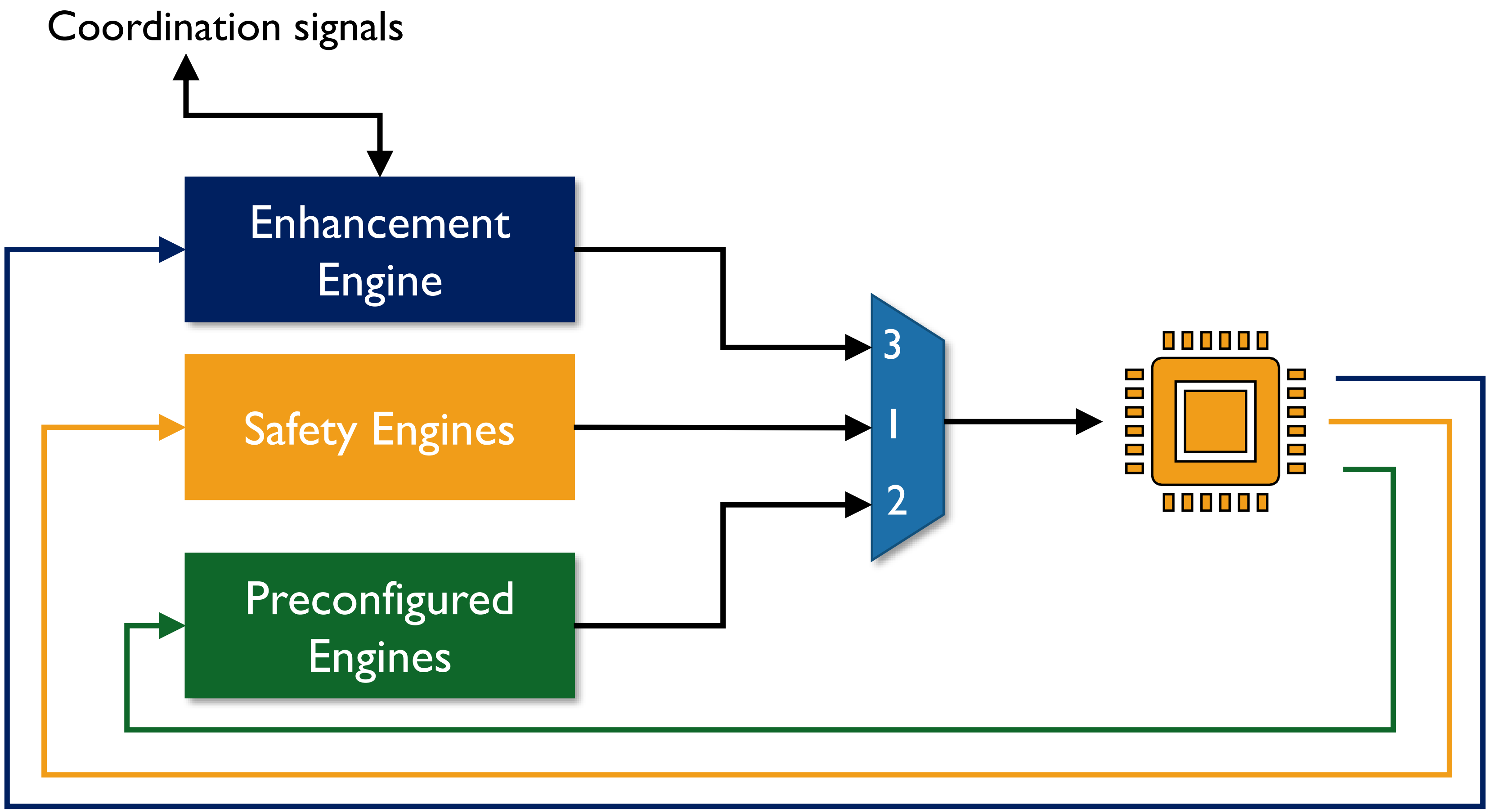
Preconfigured Engines



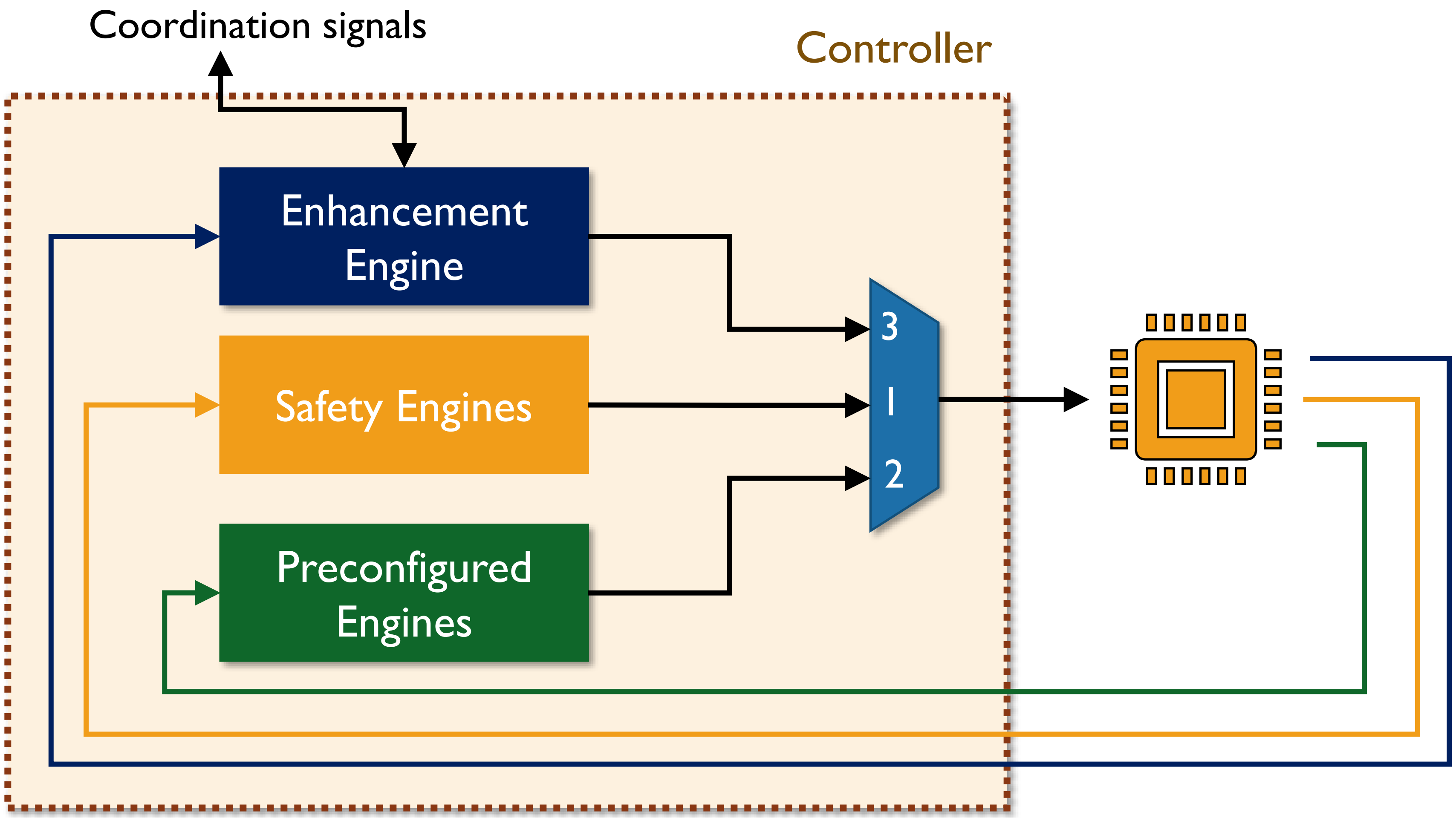
Proposed Controller



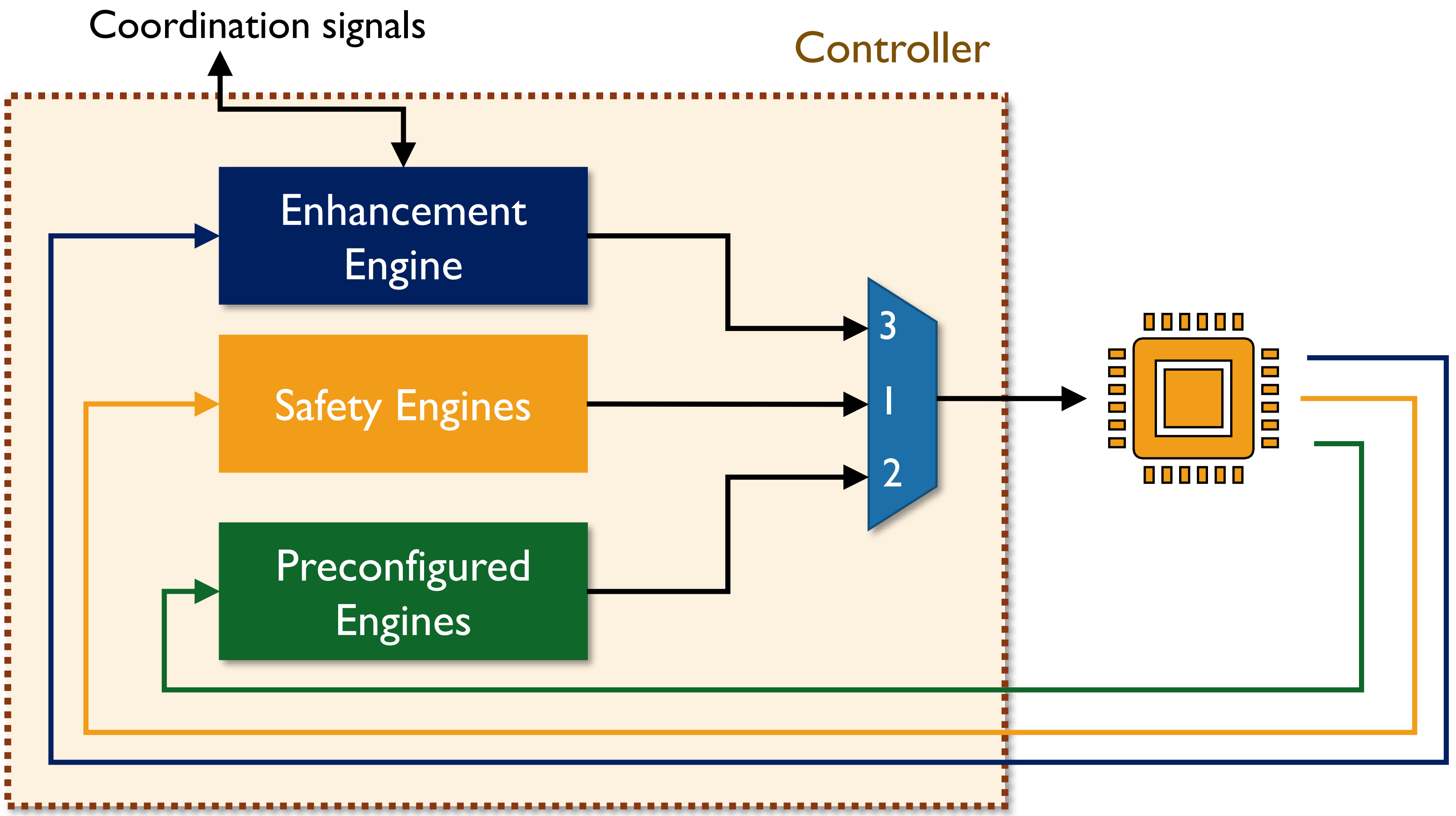
Proposed Controller



Proposed Controller

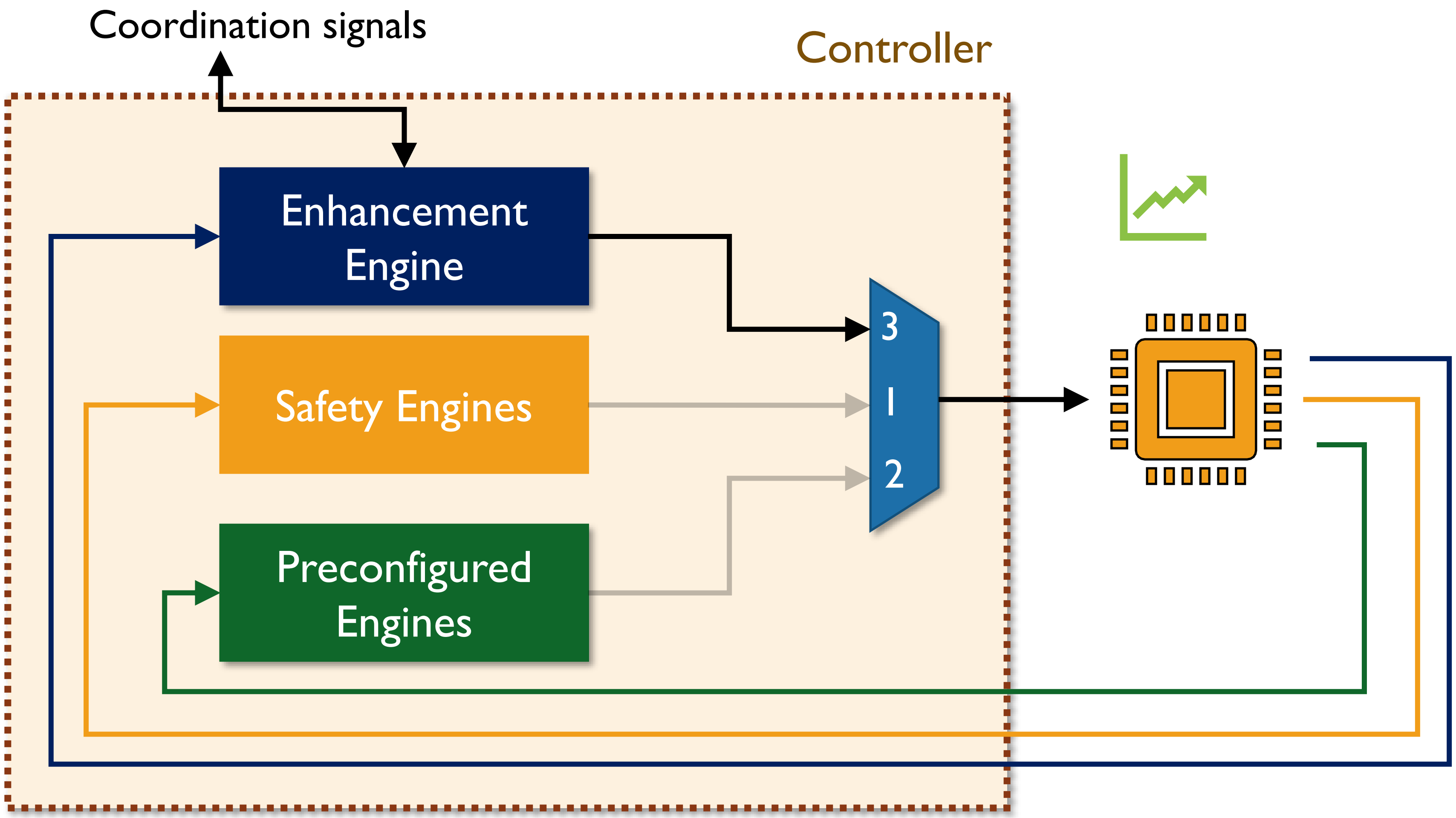


Proposed Controller



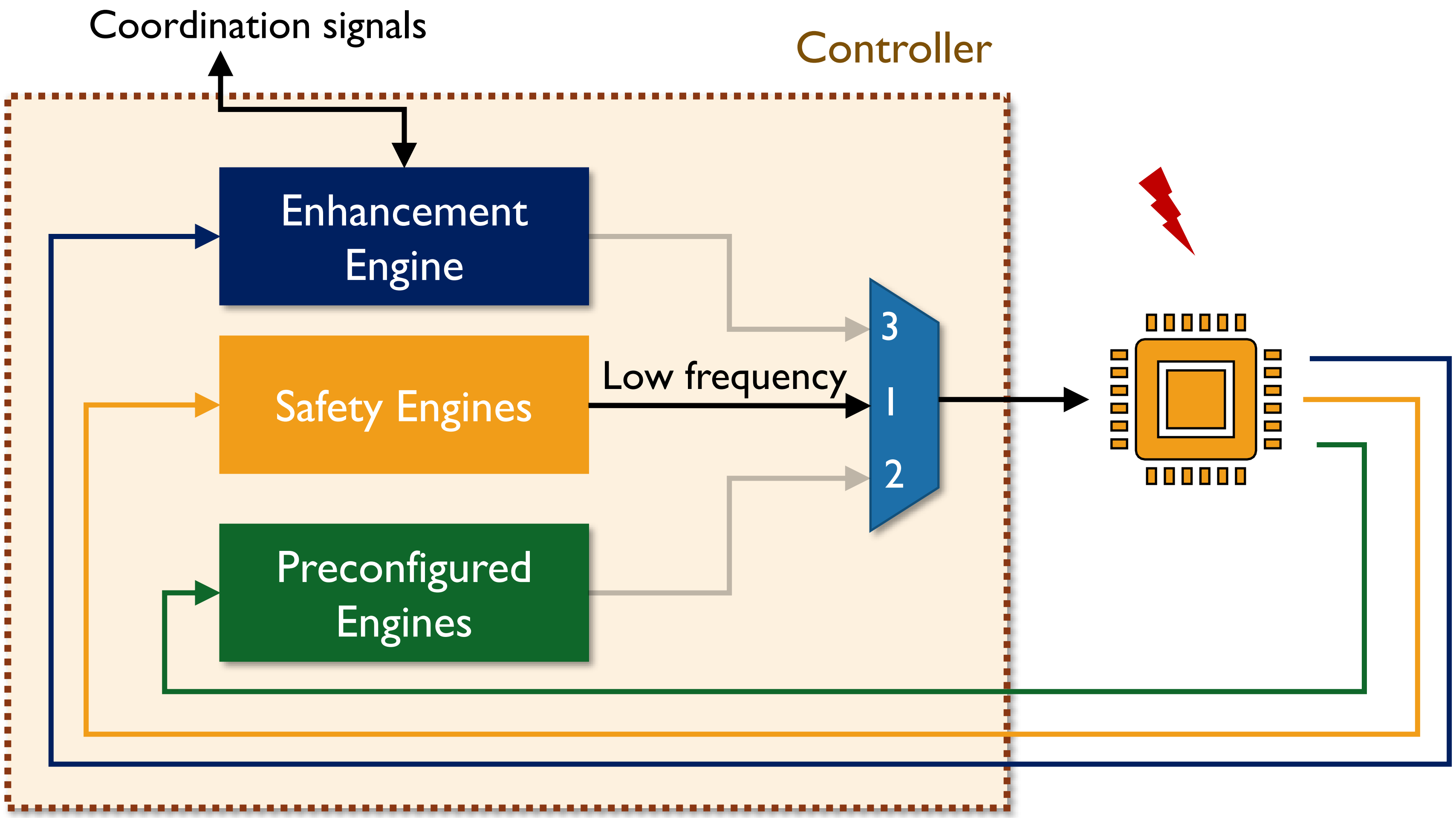
Enhancement engine is robust to interference from other engines, and subsystems

Proposed Controller



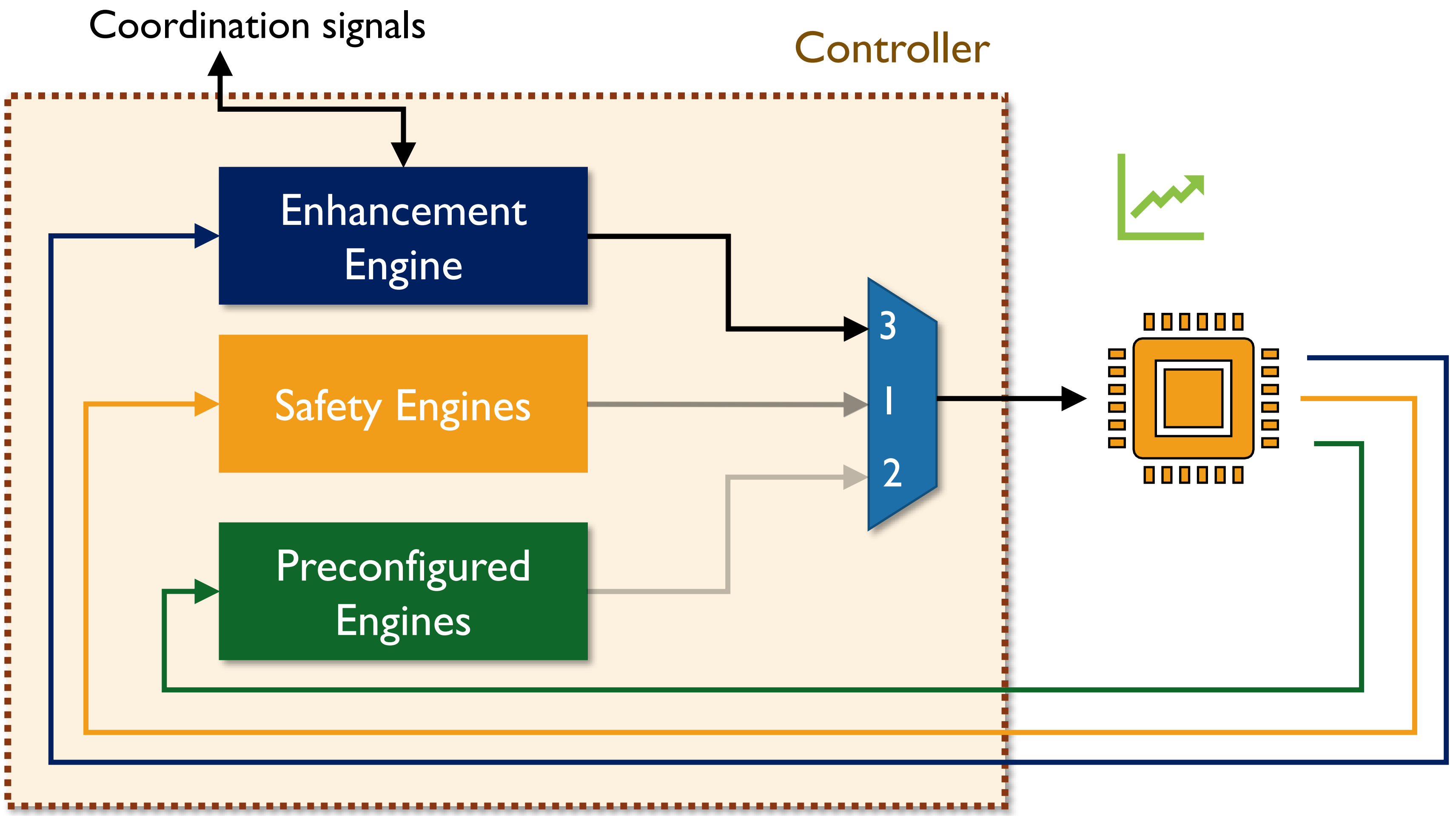
Enhancement engine is robust to interference from other engines, and subsystems

Proposed Controller



Enhancement engine is robust to interference from other engines, and subsystems

Proposed Controller



Enhancement engine is robust to interference from other engines, and subsystems

Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically

Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically

Subsystem, e.g., CPU Chip

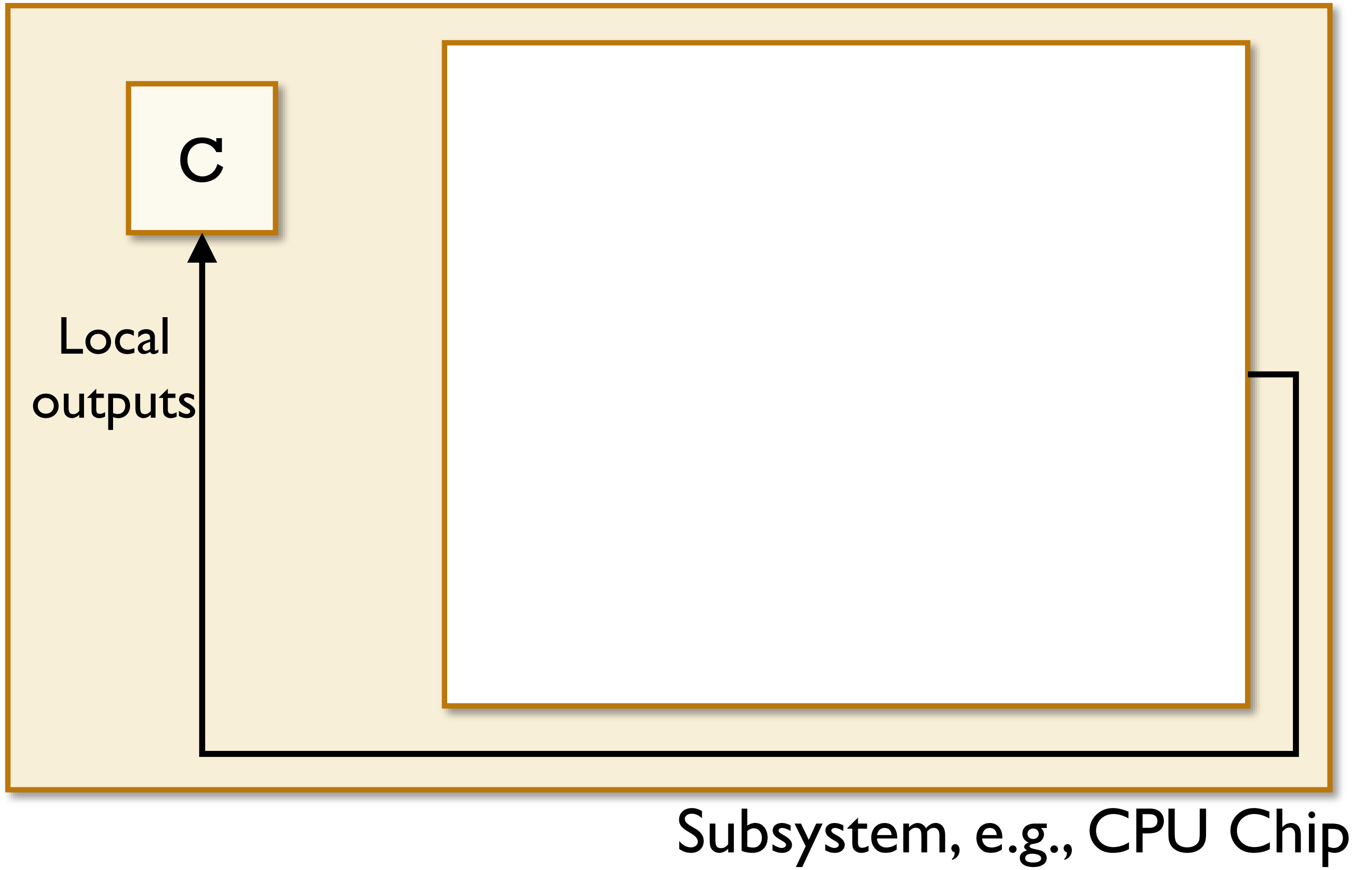
Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically

Subsystem, e.g., CPU Chip

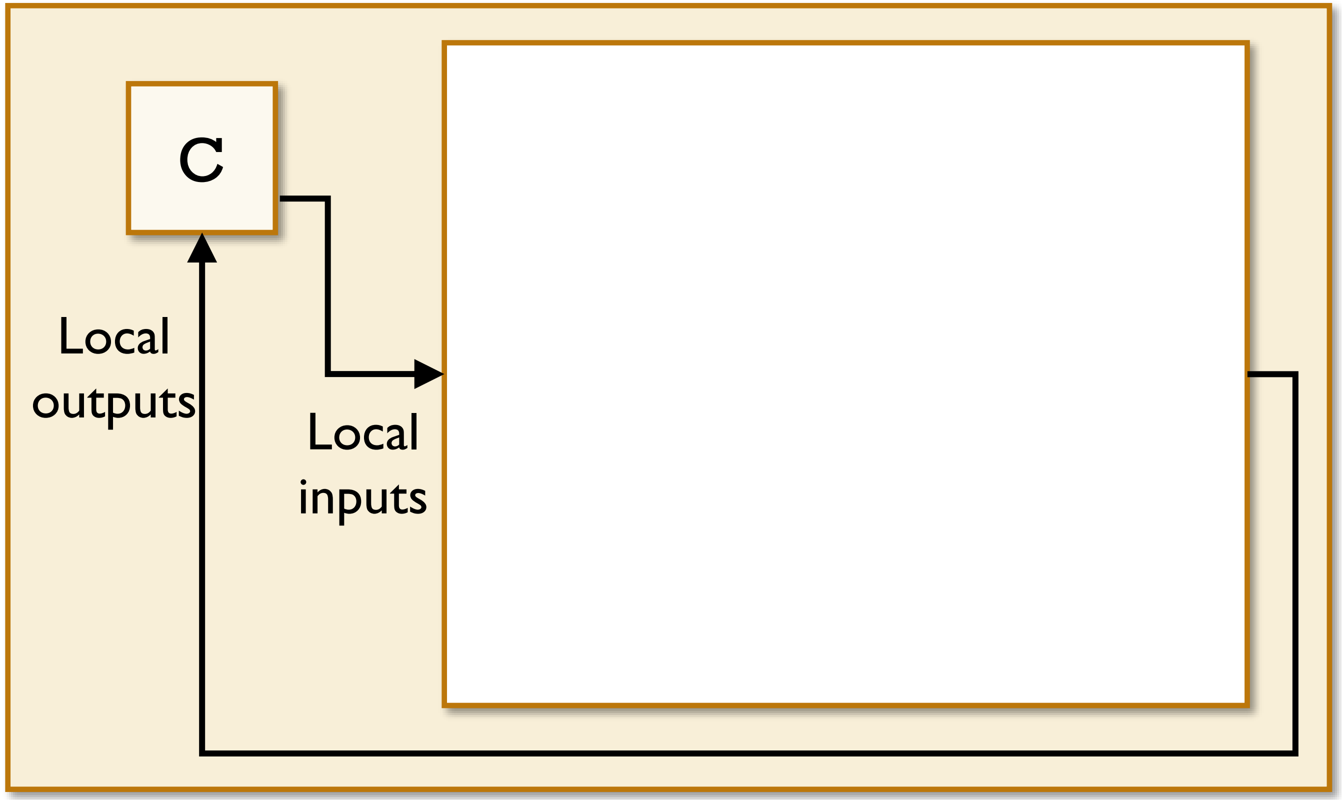
Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically



Proposed Controller Interface

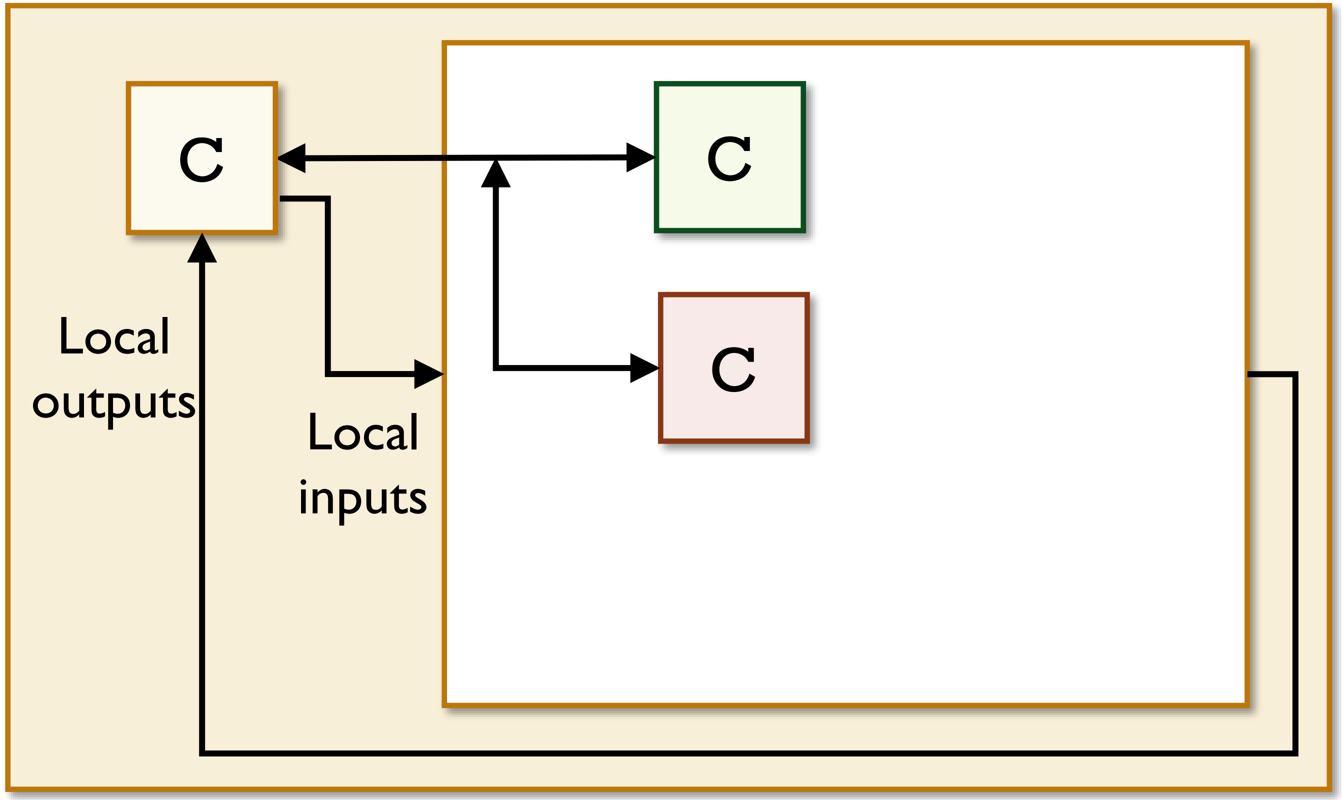
Optimize/control locally and propagate constraints hierarchically



Subsystem, e.g., CPU Chip

Proposed Controller Interface

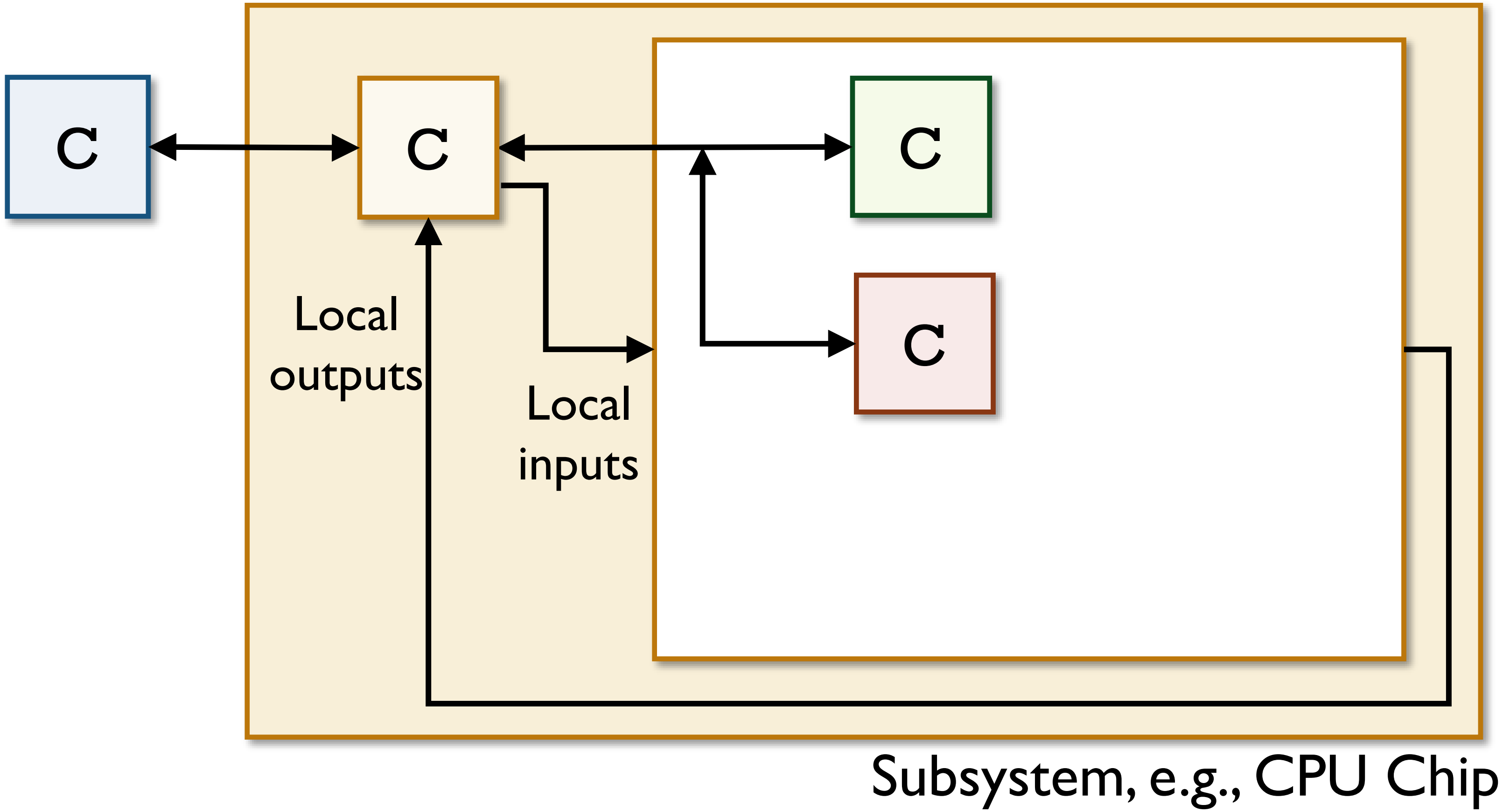
Optimize/control locally and propagate constraints hierarchically



Subsystem, e.g., CPU Chip

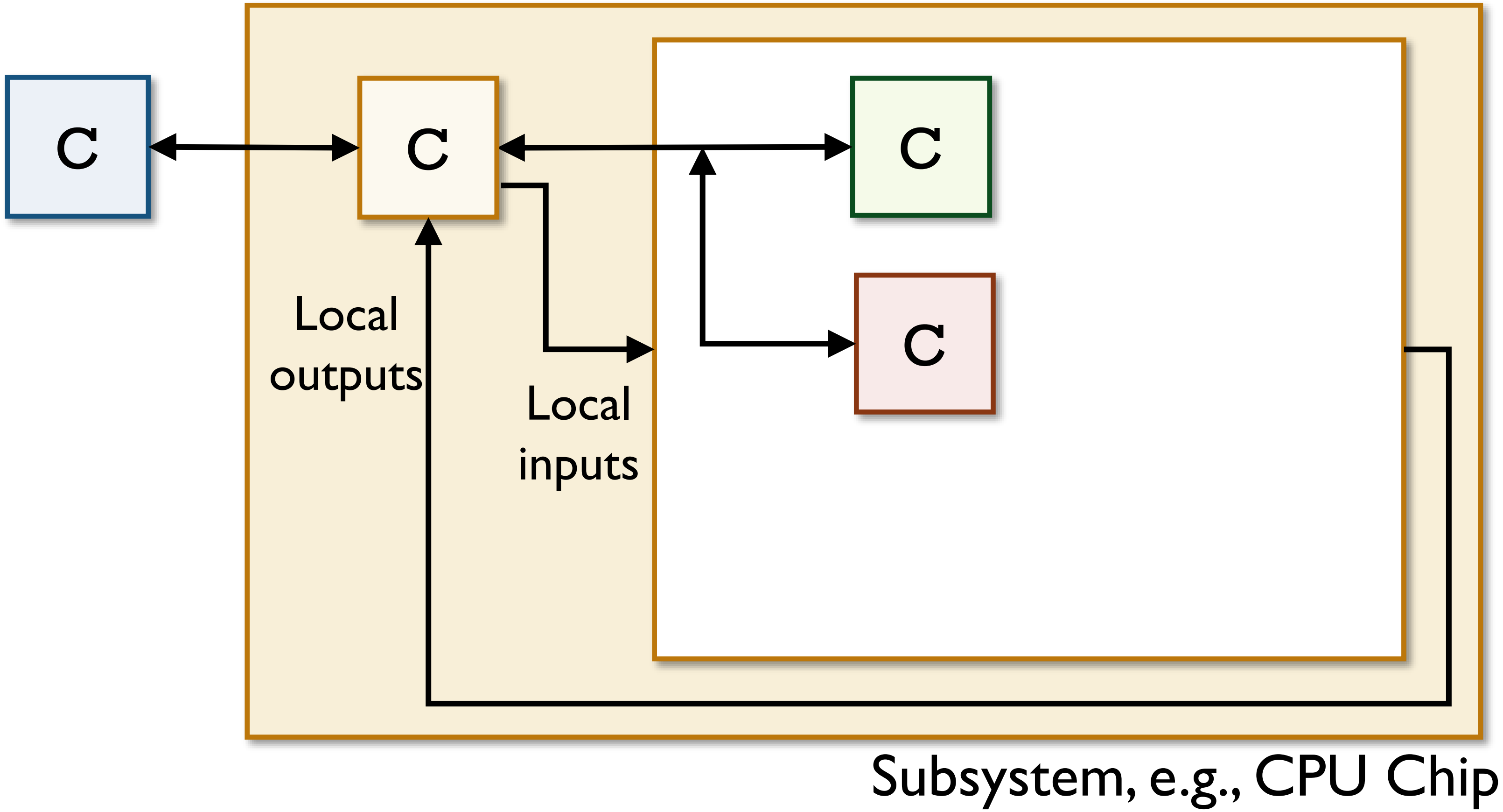
Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically



Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically

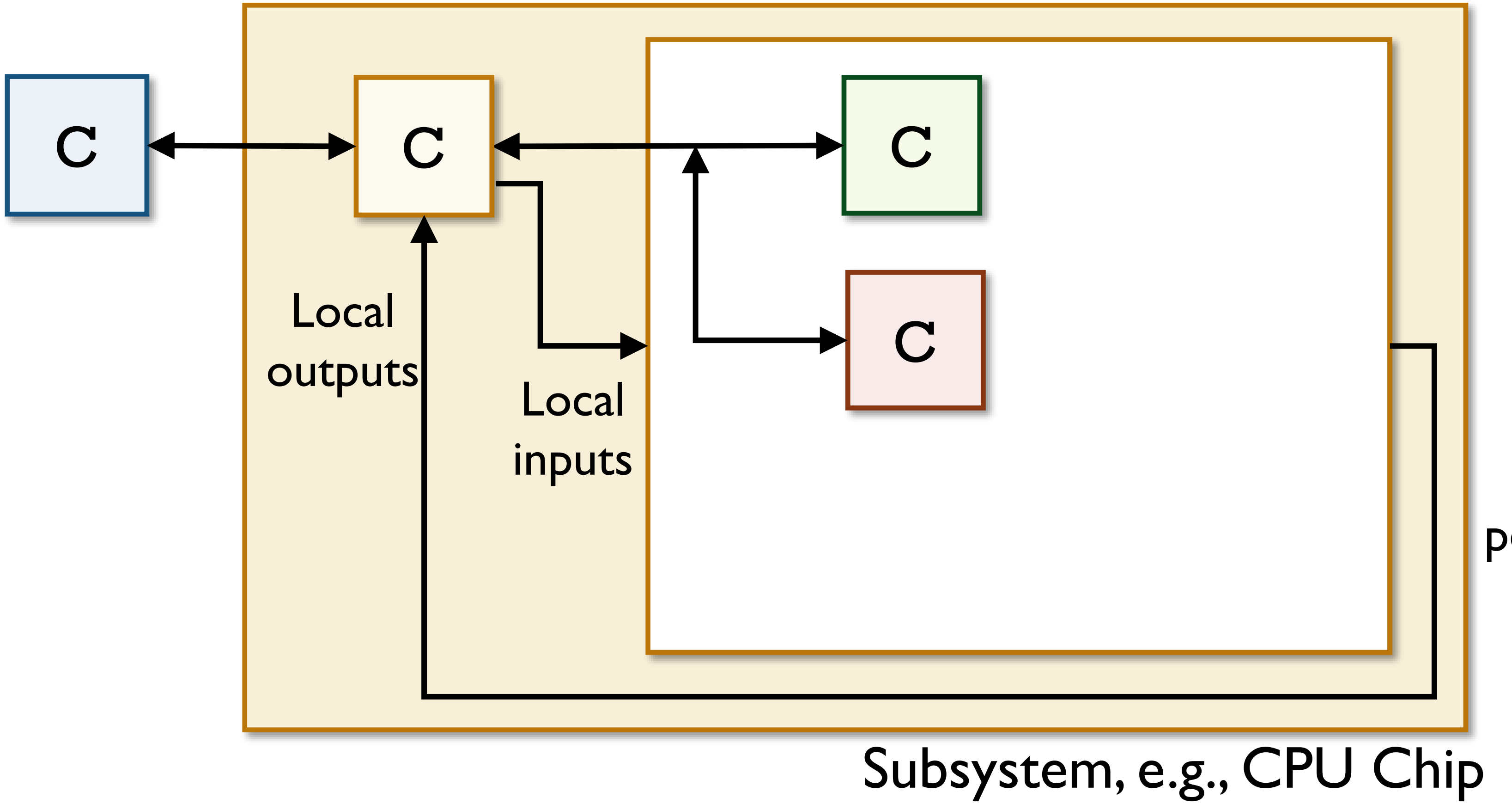


Parent to Child

max power,
min performance, etc.

Proposed Controller Interface

Optimize/control locally and propagate constraints hierarchically



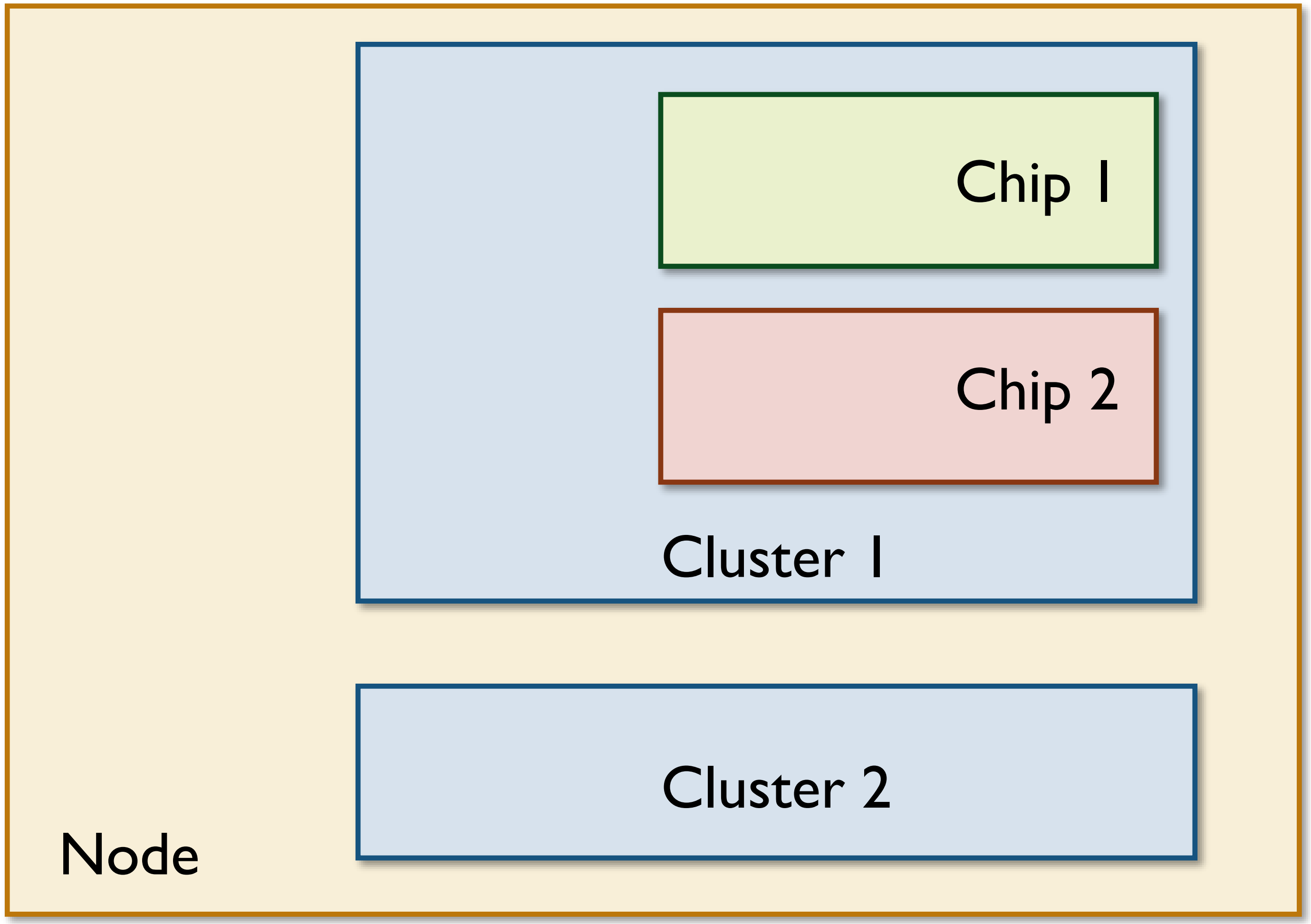
Parent to Child

max power,
min performance, etc.

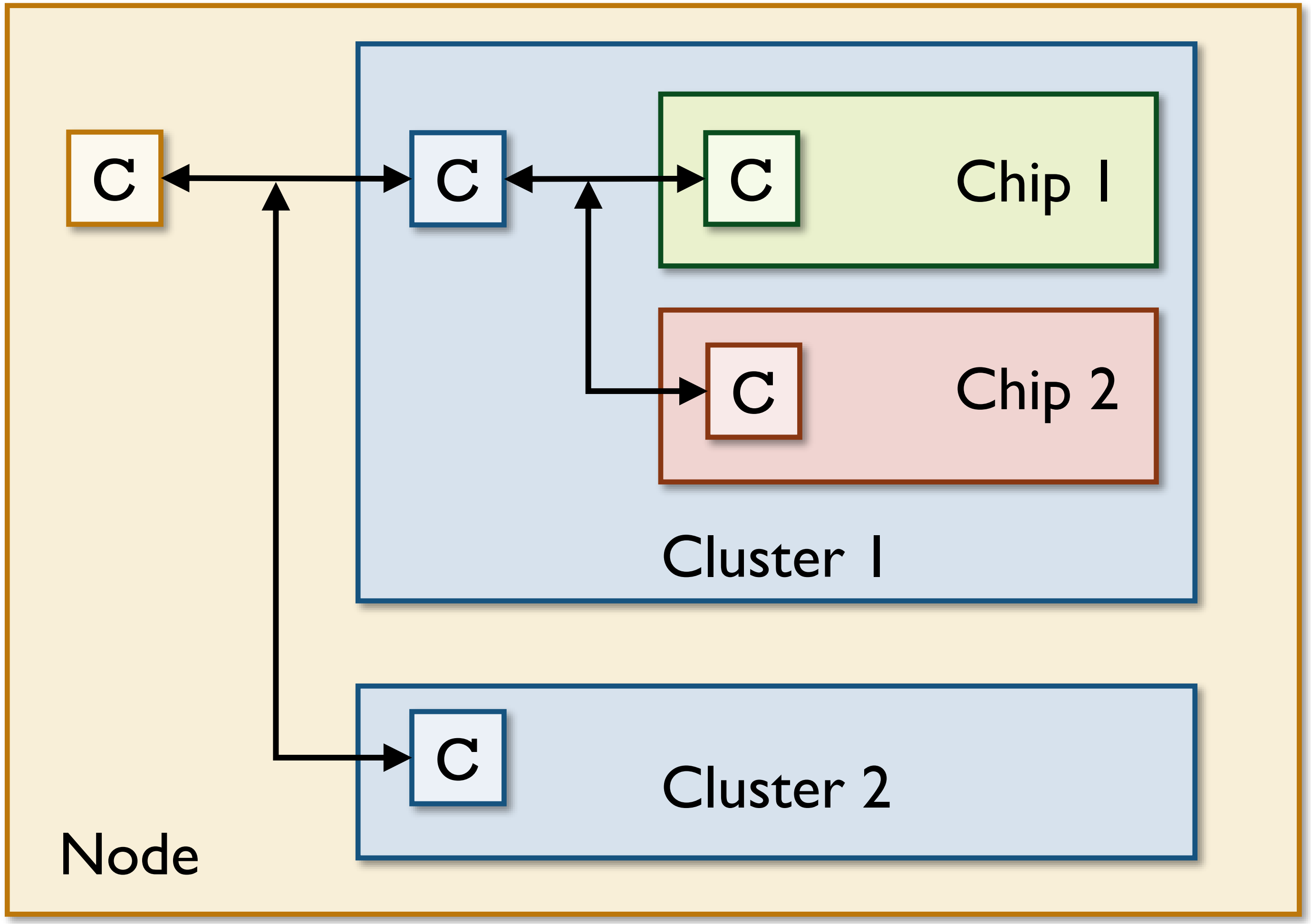
Child to Parent

power used,
performance achieved, etc.

Tangram Control Framework

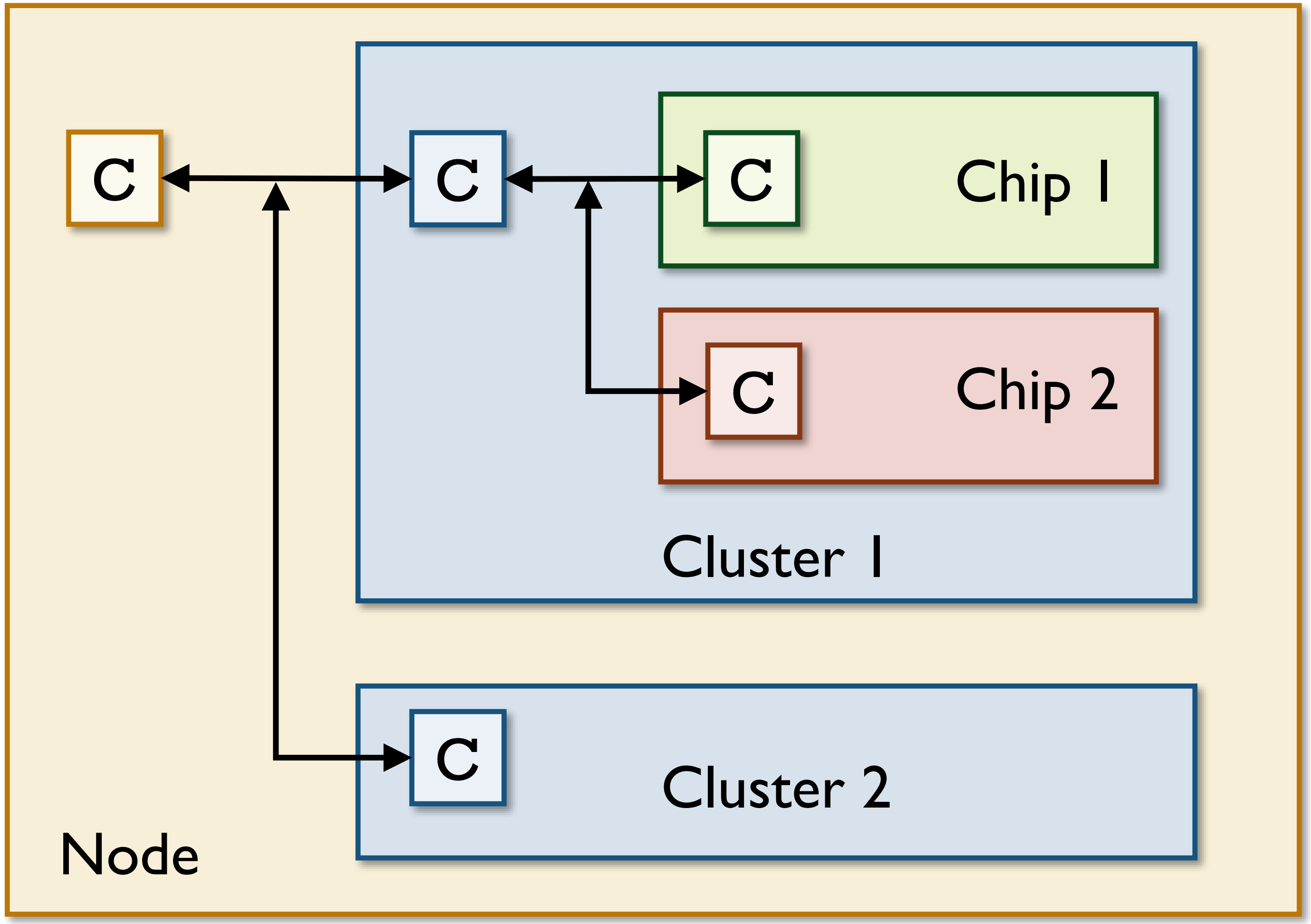


Tangram Control Framework



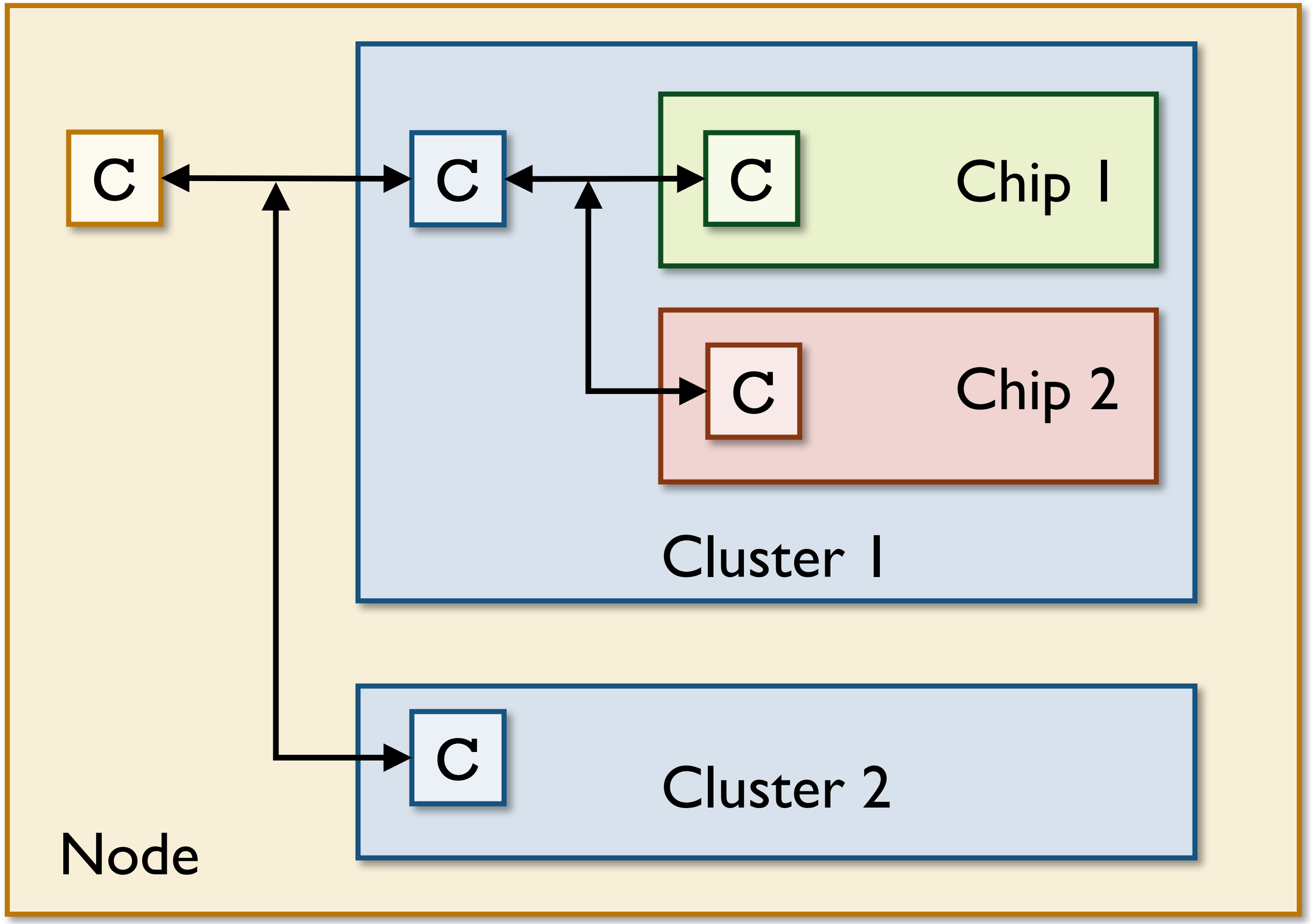
Tangram Control Framework

Control organization naturally fits the hierarchical organization of the computer



Tangram Control Framework

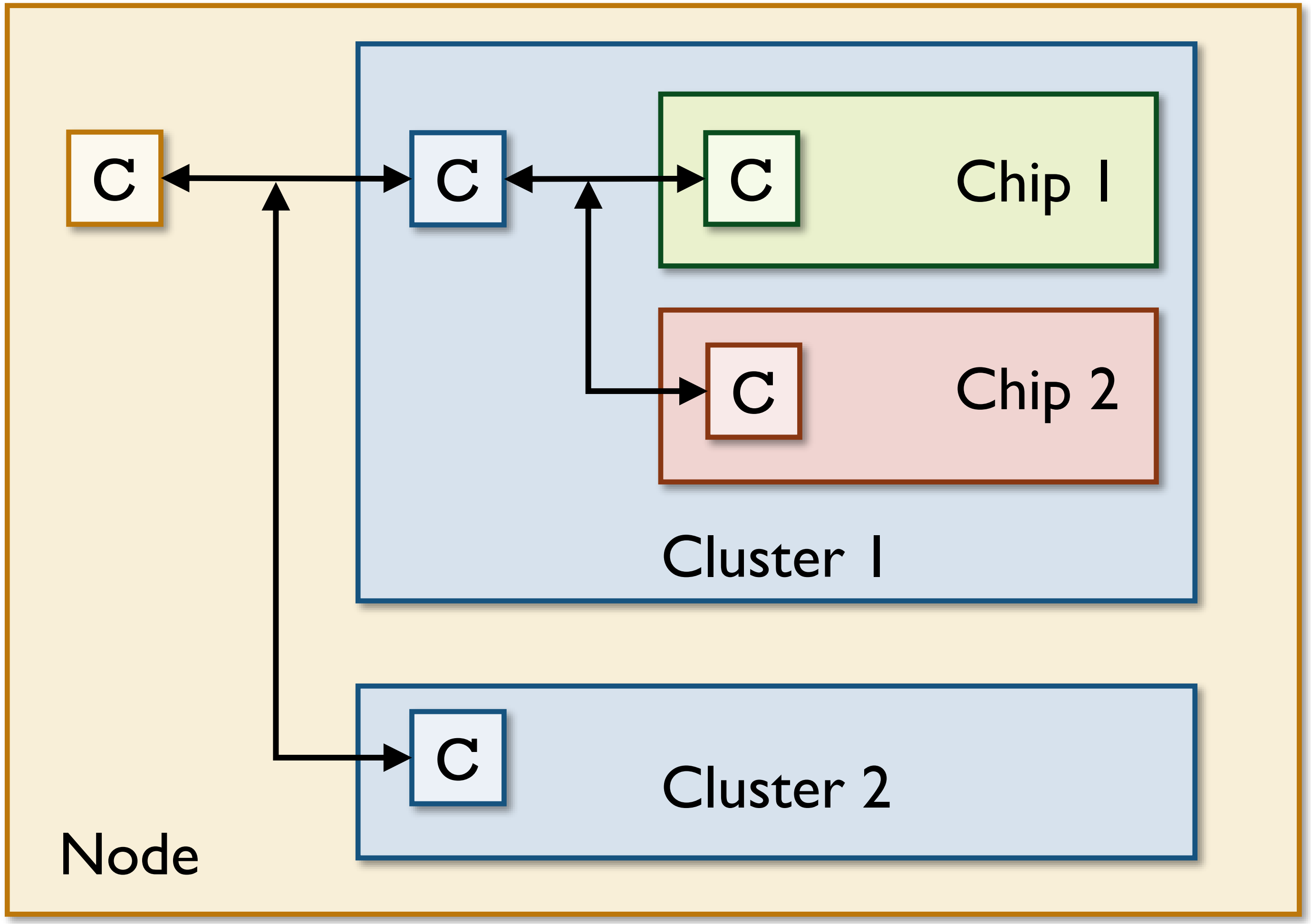
Control organization naturally fits the hierarchical organization of the computer



- Fast

Tangram Control Framework

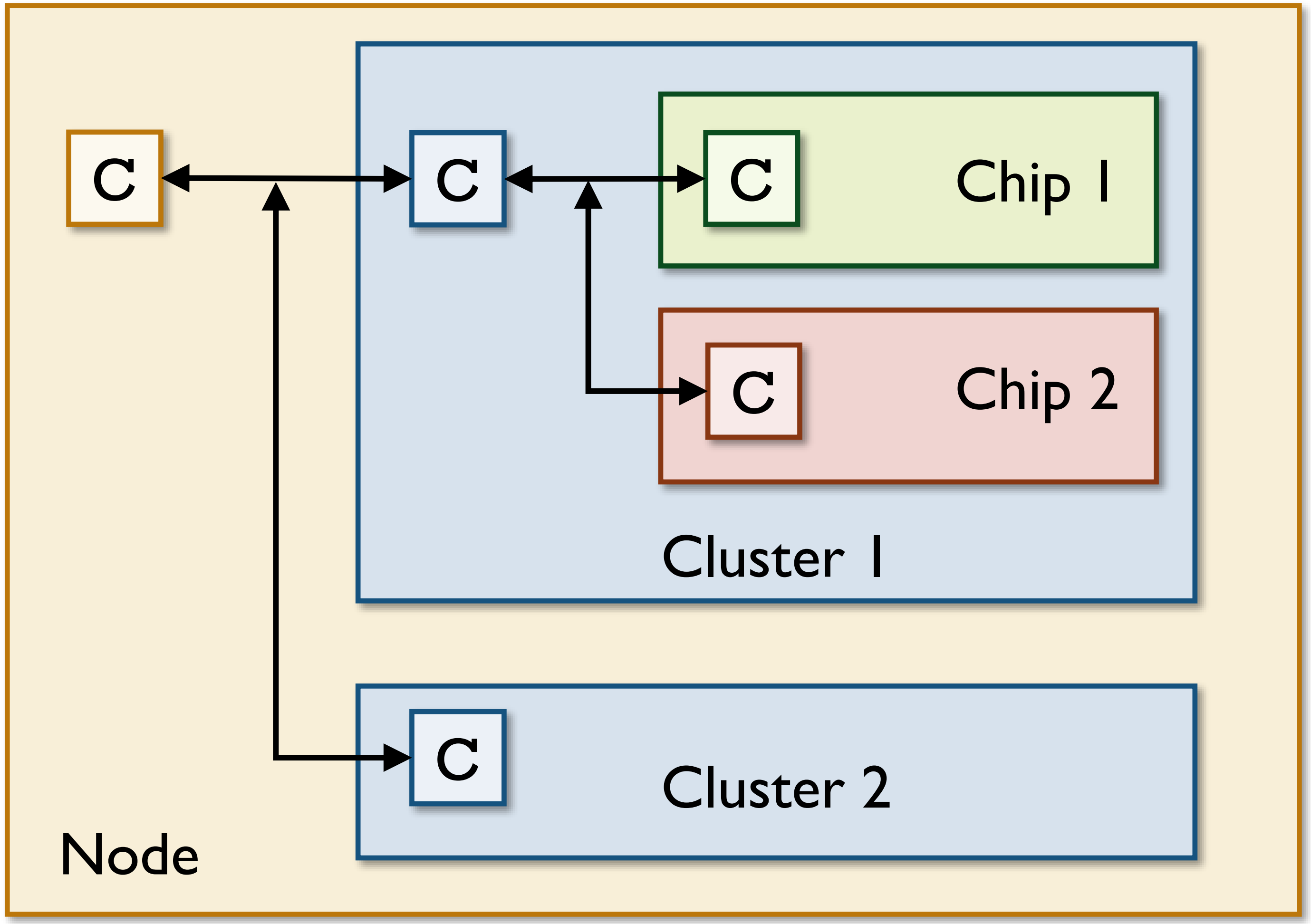
Control organization naturally fits the hierarchical organization of the computer



- Fast
- Globally coordinated

Tangram Control Framework

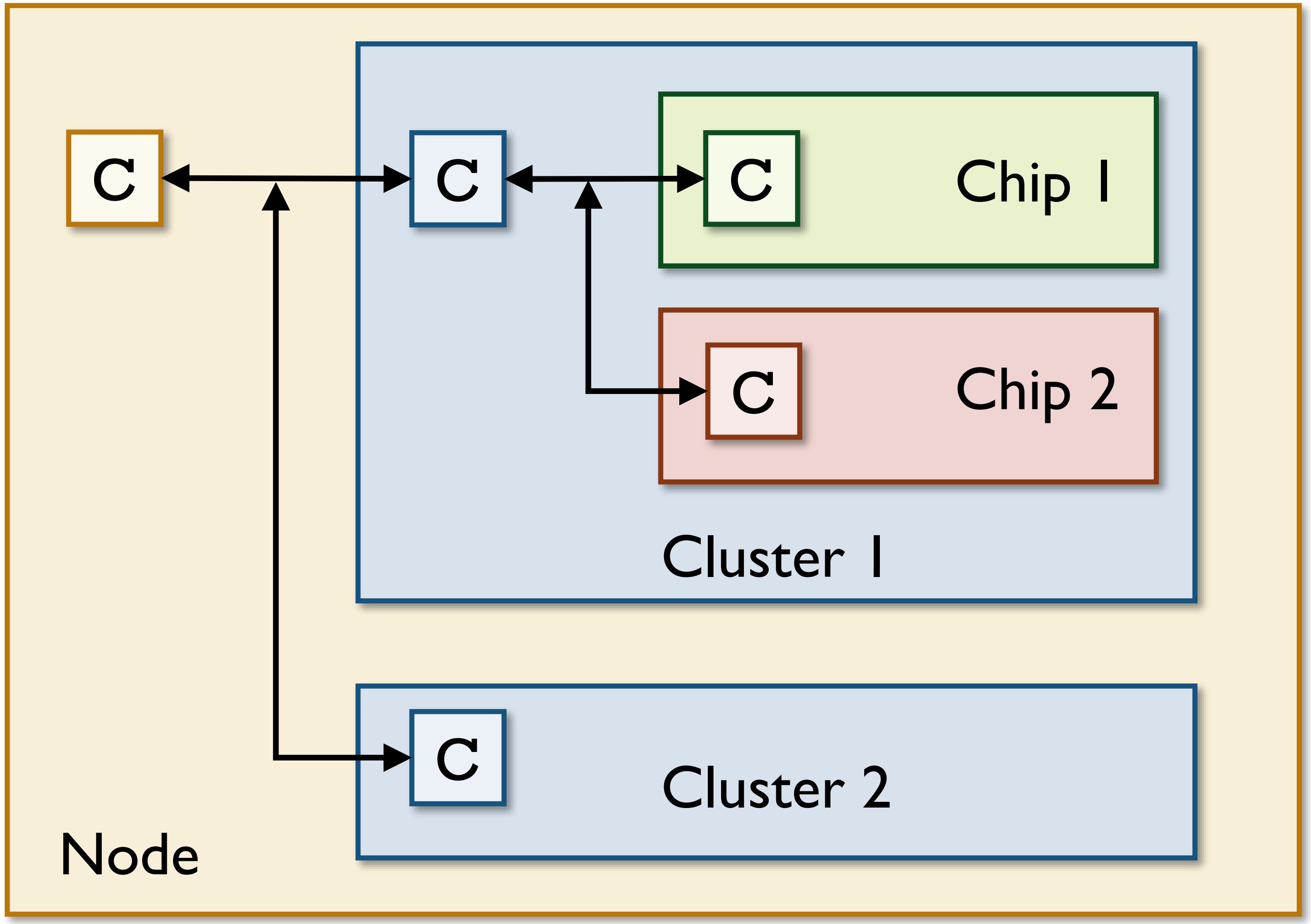
Control organization naturally fits the hierarchical organization of the computer



- Fast
- Globally coordinated
- Modular

Tangram Control Framework

Control organization naturally fits the hierarchical organization of the computer



- Fast
- Globally coordinated
- Modular

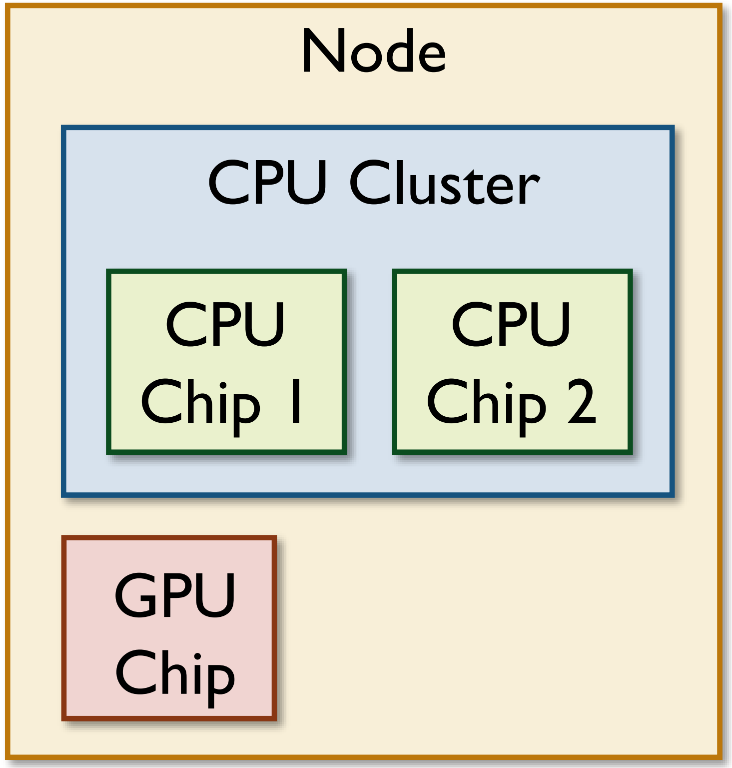
Obvious and simple:
Thanks to formal control!

Prototype

CPUs: AMD Ryzen™ 1800X
GPU: AMD Radeon™ RX 580

Controllers built as privileged processes

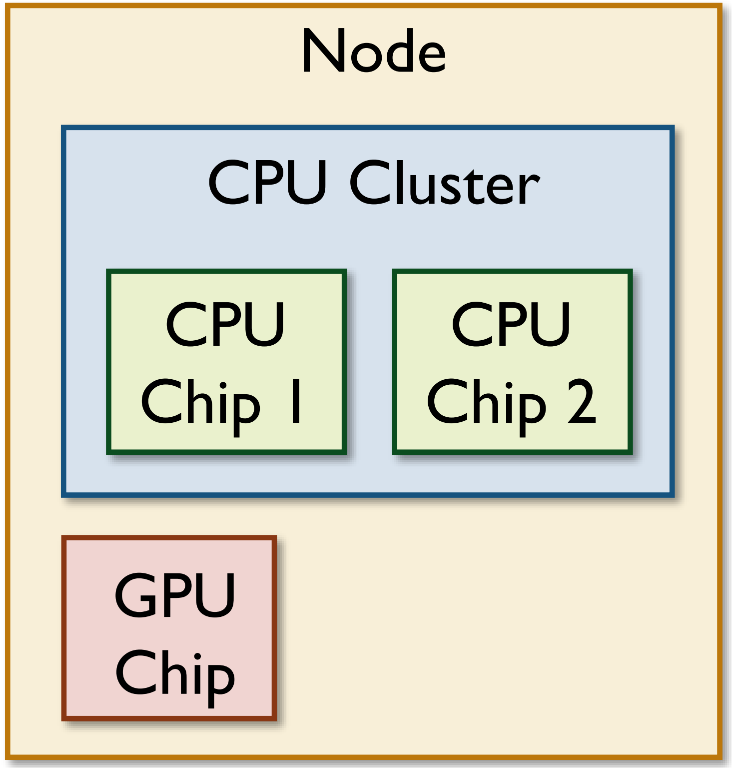
Prototype



CPUs: AMD Ryzen™ 1800X
GPU: AMD Radeon™ RX 580

Controllers built as privileged processes

Prototype

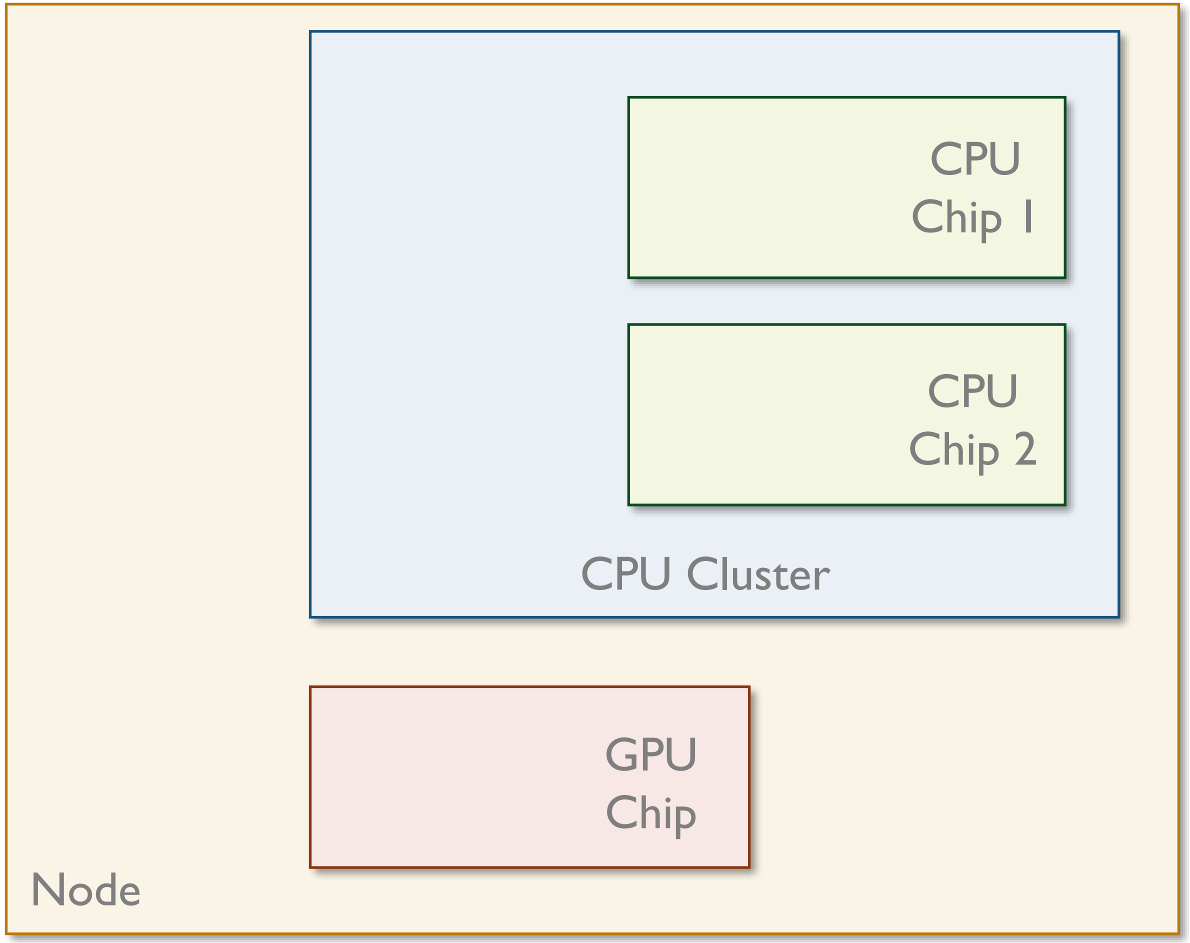


Minimize Energy×Delay with power constraints
Current and temperature safety

CPUs: AMD Ryzen™ 1800X
GPU: AMD Radeon™ RX 580

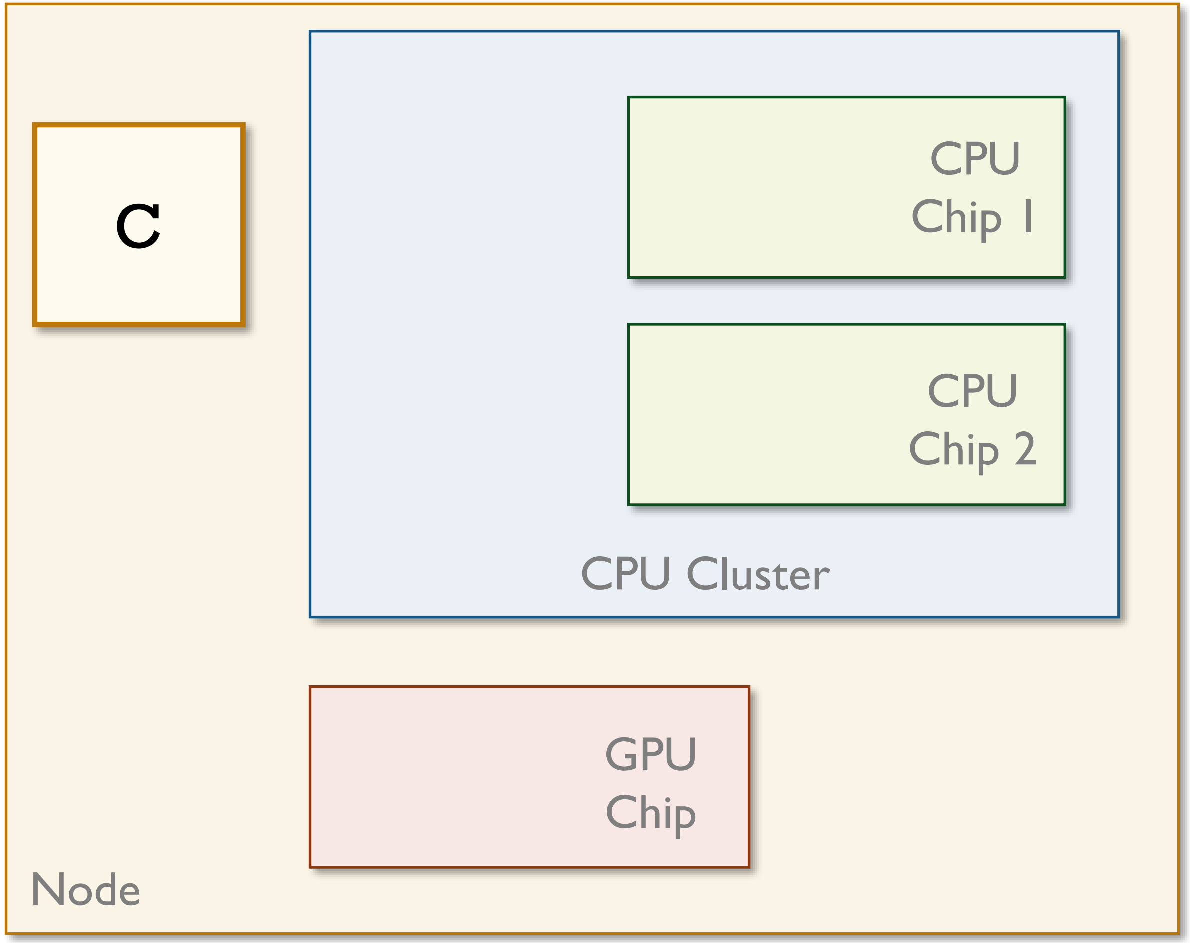
Controllers built as privileged processes

Prototype Response Times



Prototype Response Times

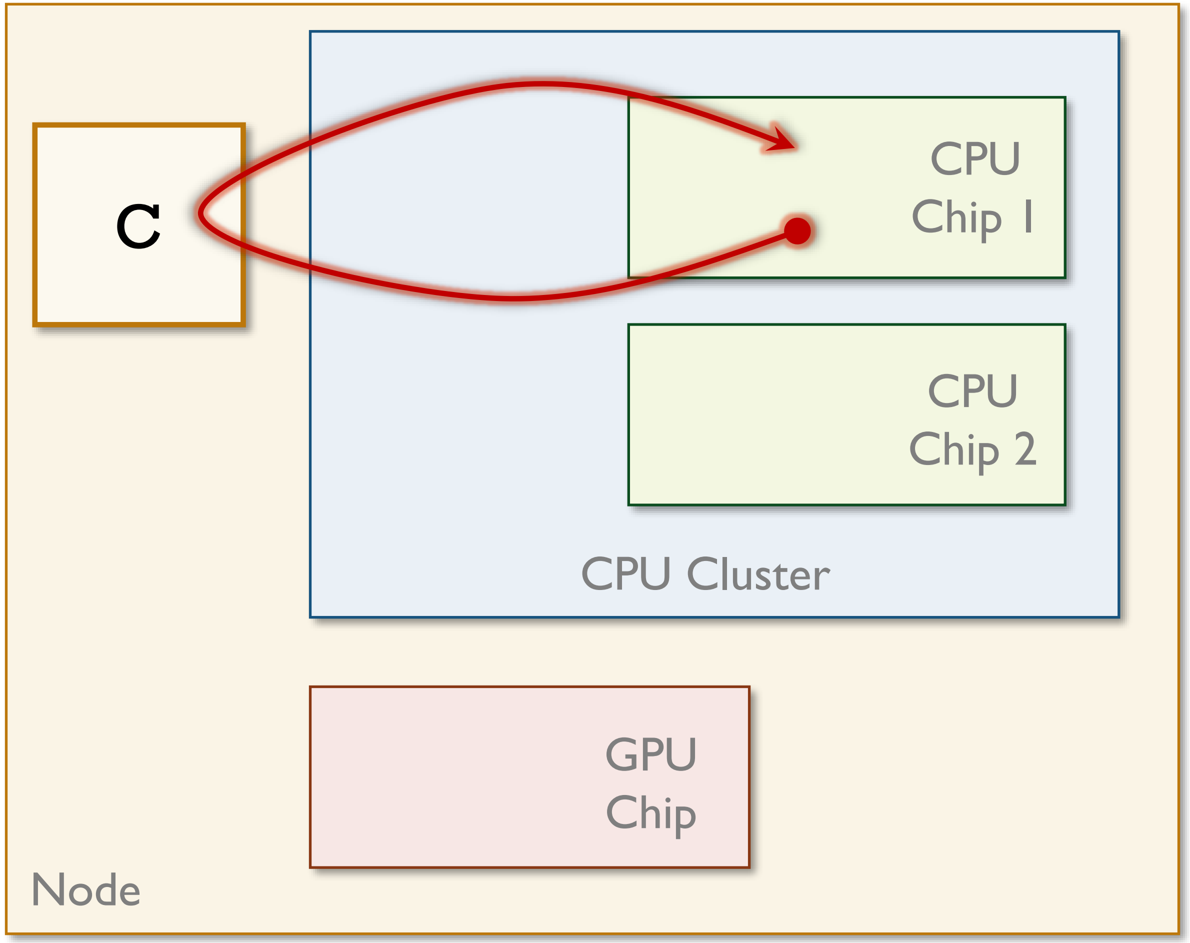
Centralized



Prototype Response Times

Centralized

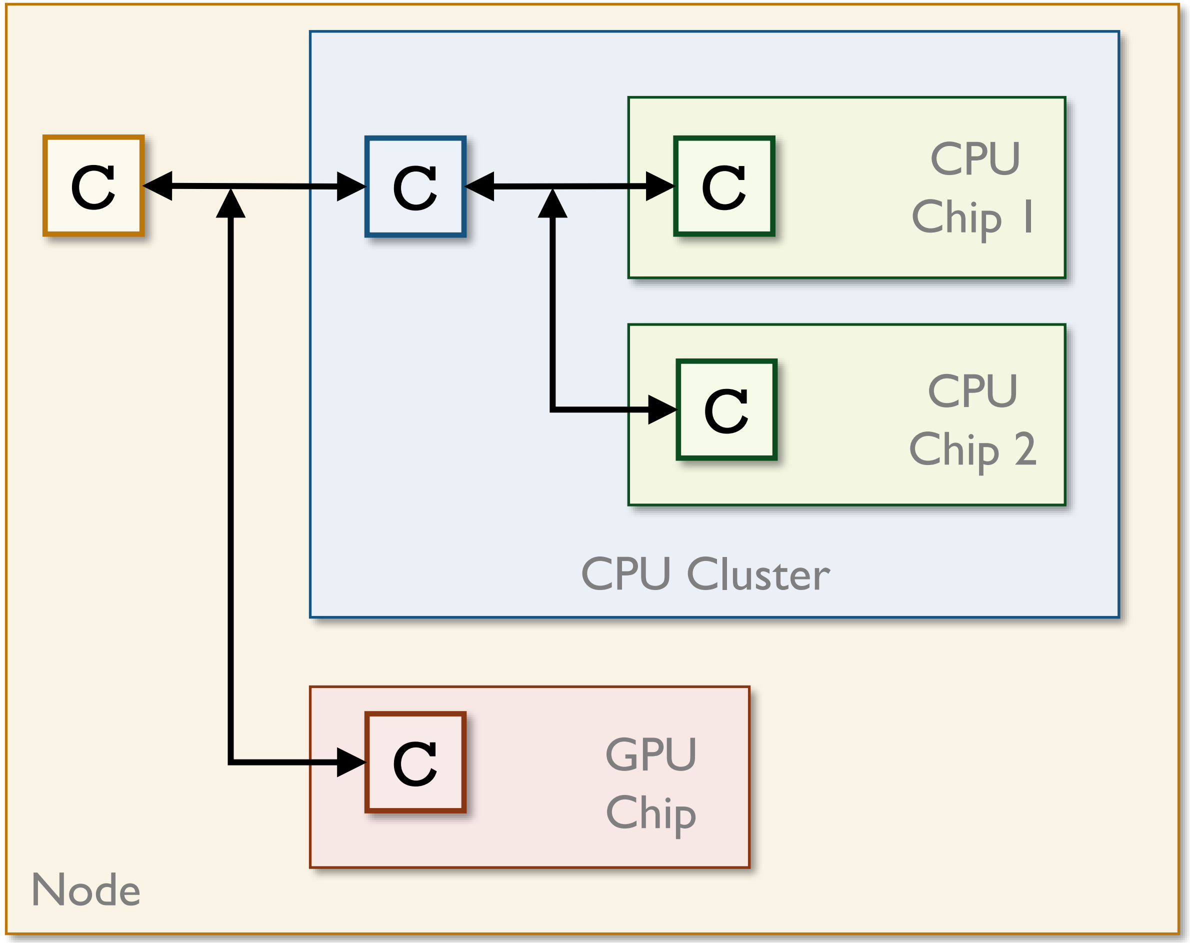
Always 200 ms



Prototype Response Times

Centralized

Always 200 ms

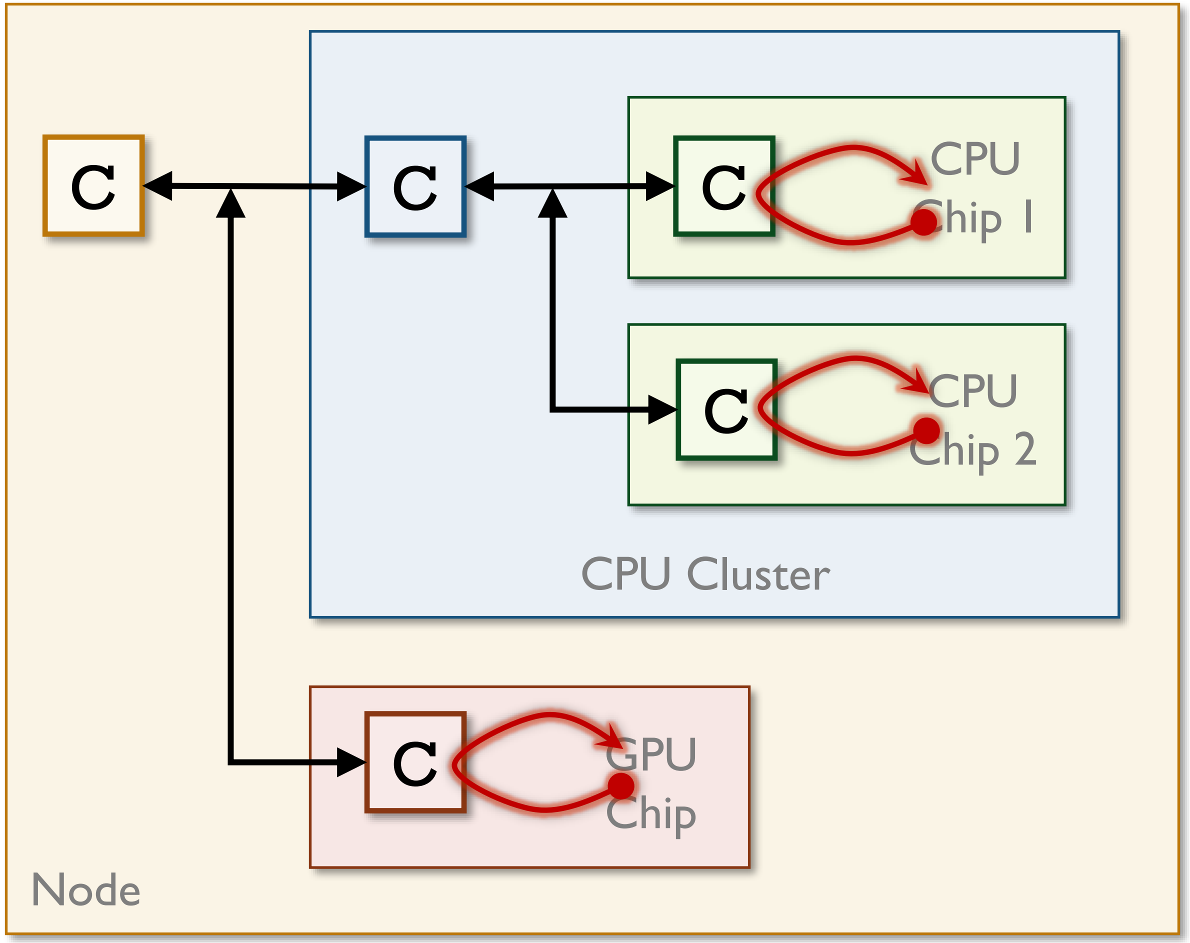


Tangram

Prototype Response Times

Centralized

Always 200 ms



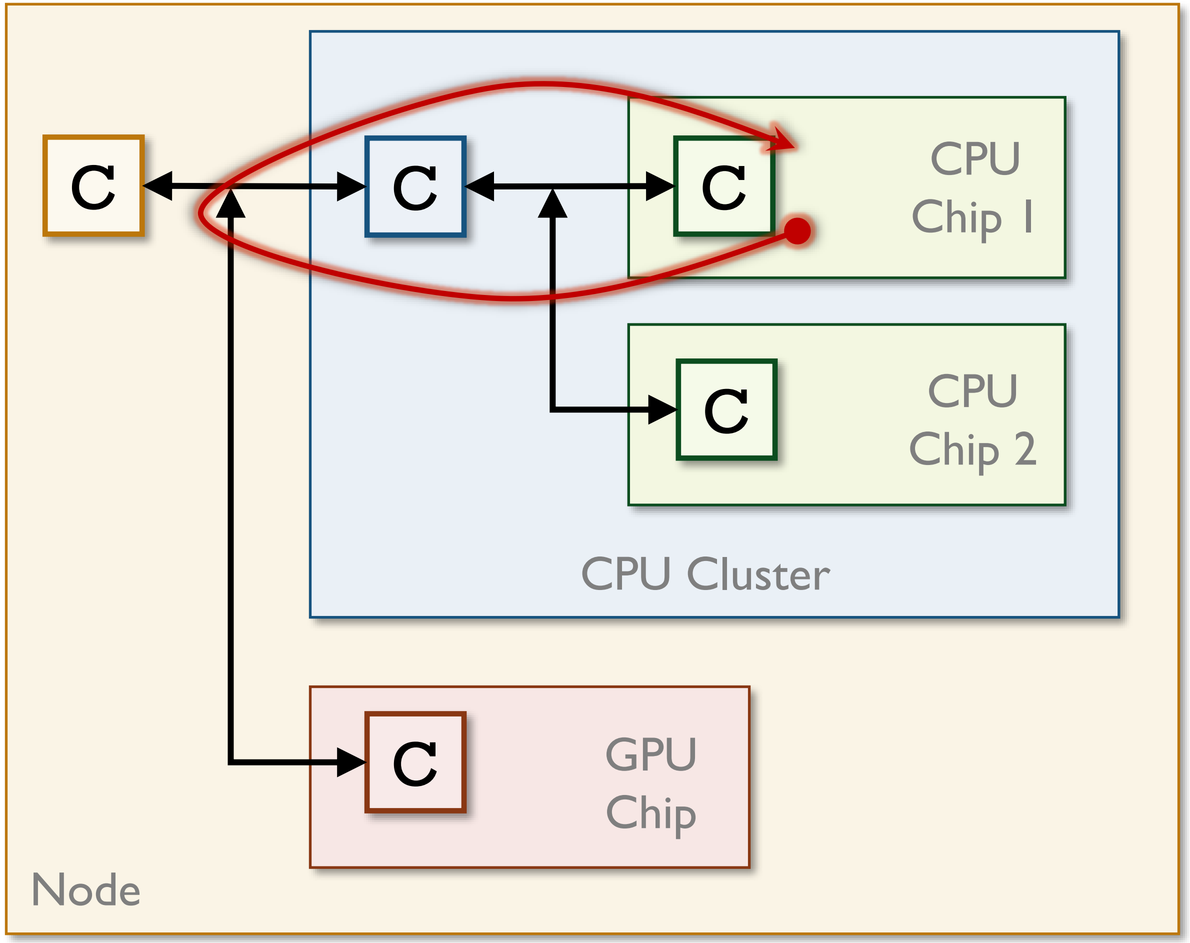
Tangram

Common case: 15 ms

Prototype Response Times

Centralized

Always 200 ms



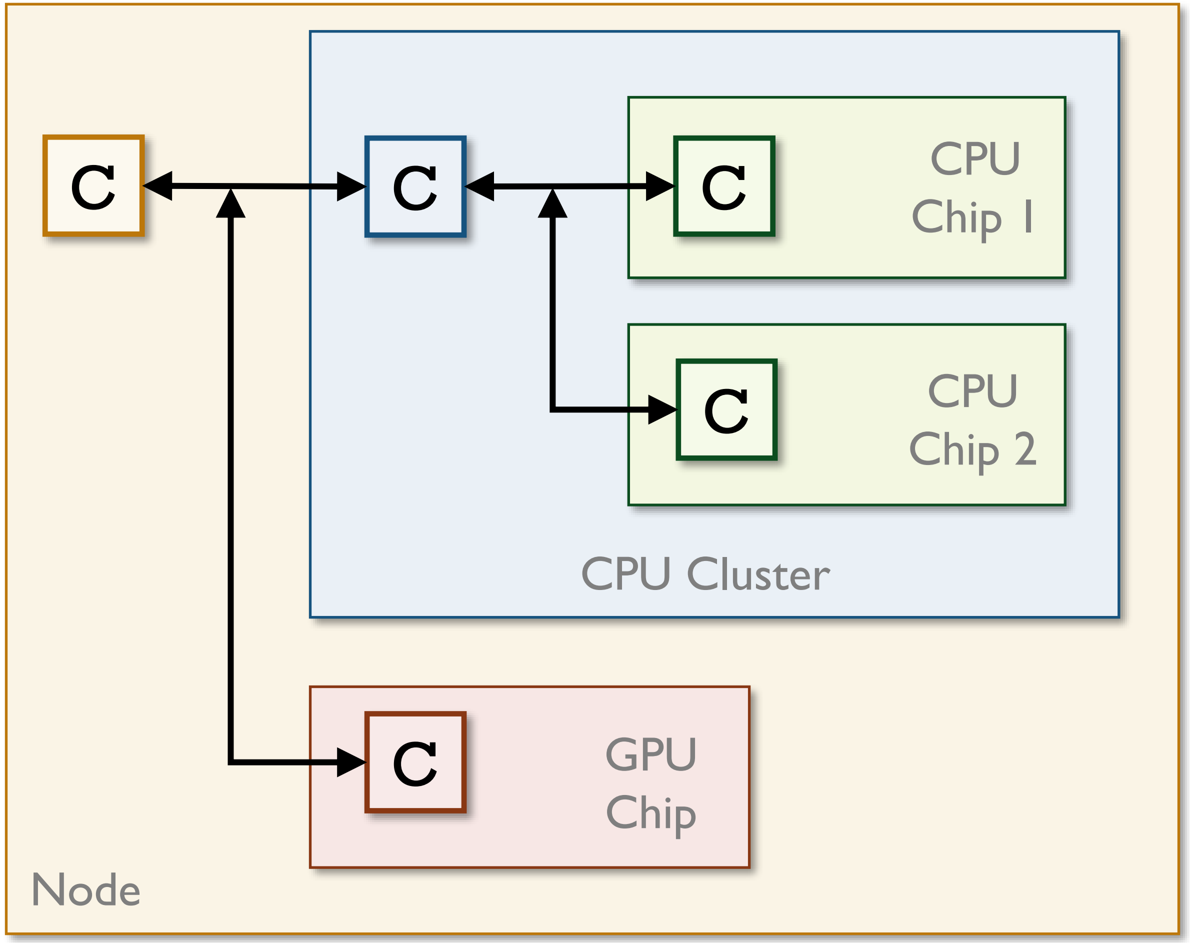
Tangram

Common case: 15 ms
Worst case: 515 ms

Prototype Response Times

Centralized

Always 200 ms



Tangram

Common case: 15 ms
Worst case: 515 ms

Preconfigured and Safety response is ≈ 2 ms

Prototype Evaluation Highlights

Control frameworks

- Research state-of-the-art
- Tangram with heuristic
- Tangram

Prototype Evaluation Highlights

Control frameworks

- Research state-of-the-art
- Tangram with heuristic
- Tangram

Chai applications

Fine grained concurrent CPU-GPU execution

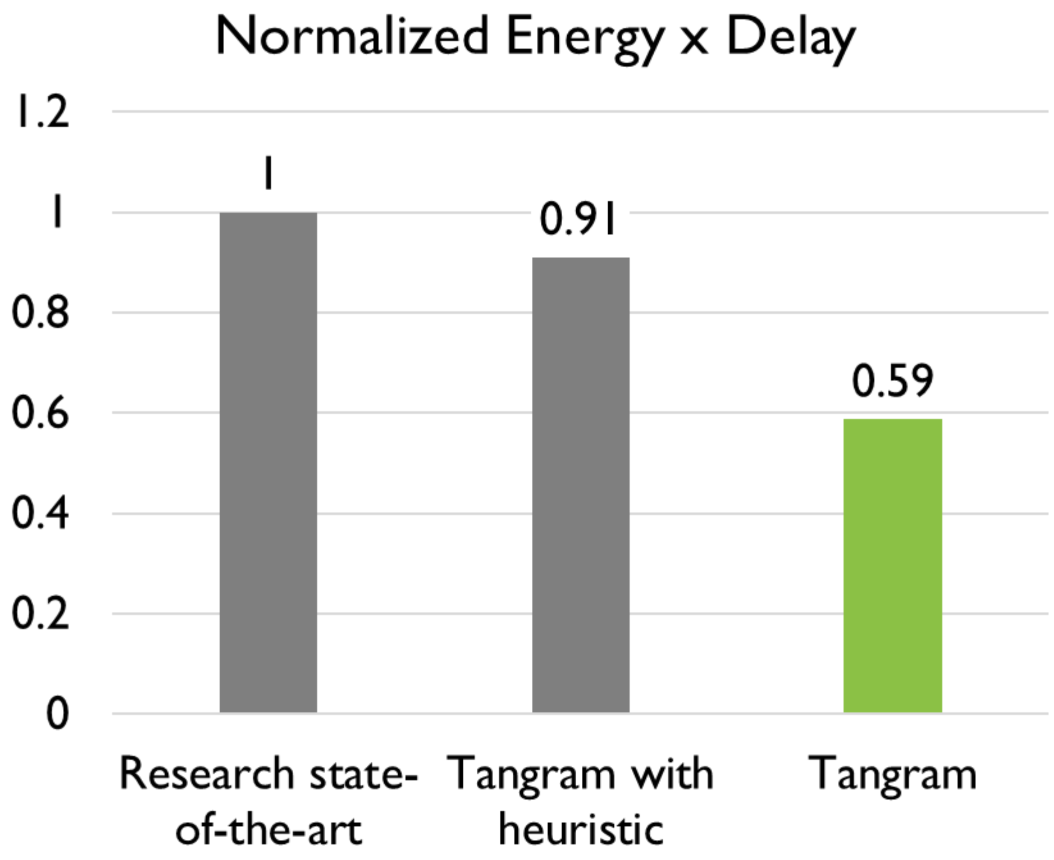
Prototype Evaluation Highlights

Control frameworks

- Research state-of-the-art
- Tangram with heuristic
- Tangram

Chai applications

Fine grained concurrent CPU-GPU execution



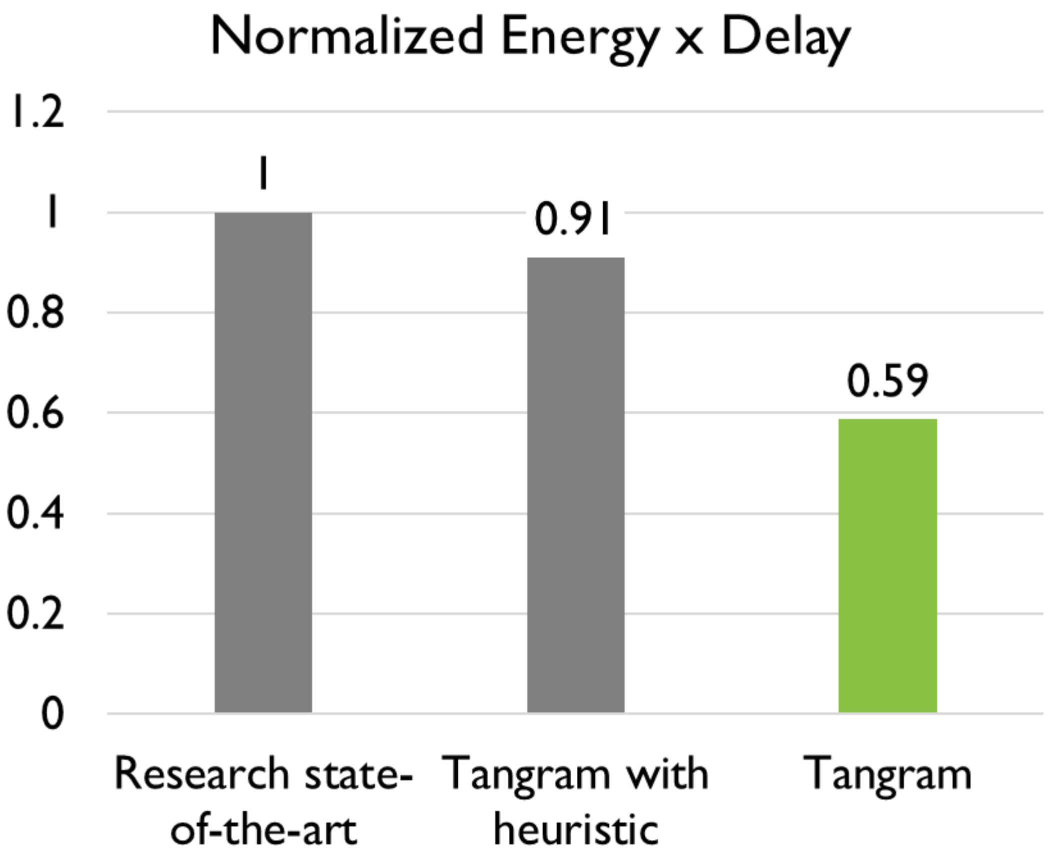
Prototype Evaluation Highlights

Control frameworks

- Research state-of-the-art
- Tangram with heuristic
- Tangram

Chai applications

Fine grained concurrent CPU-GPU execution



Tangram: 32% faster with 13% lower energy ⇒ 41% better EDP

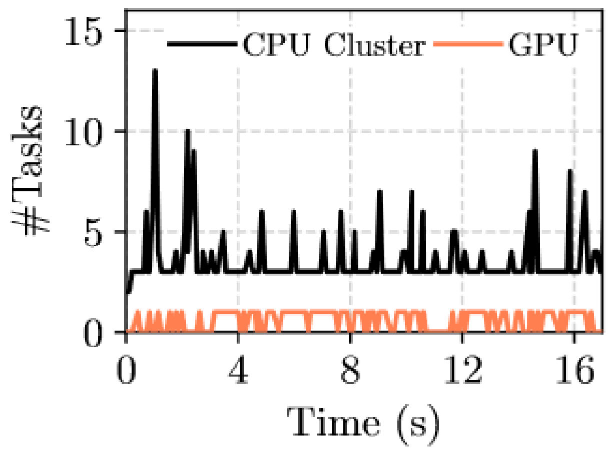
Collaborative Application: *rsct* from Chai

Research state-of-the-art

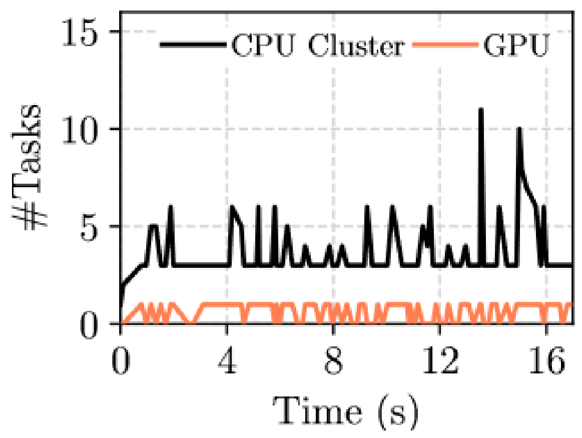
Tangram

Collaborative Application: *rsct* from Chai

Research state-of-the-art

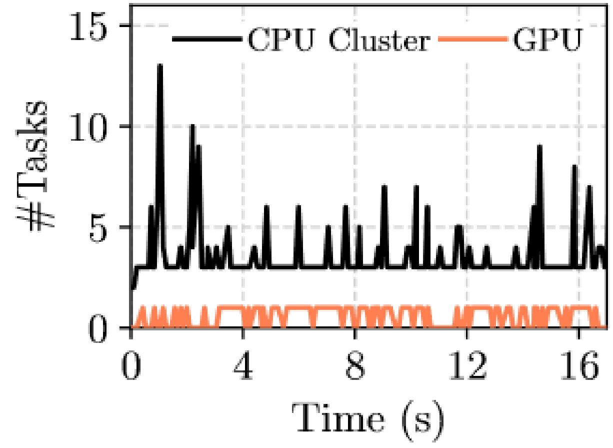


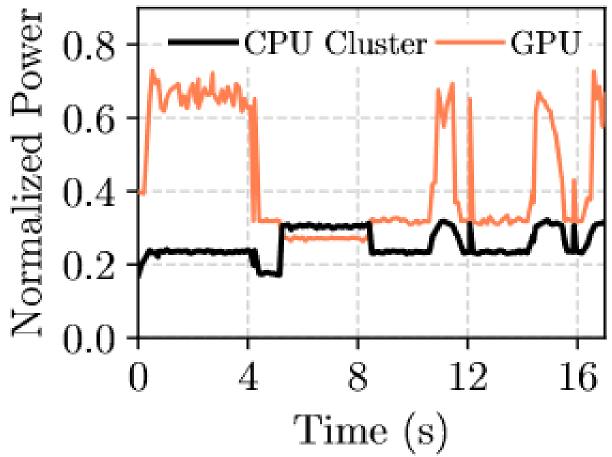
Tangram



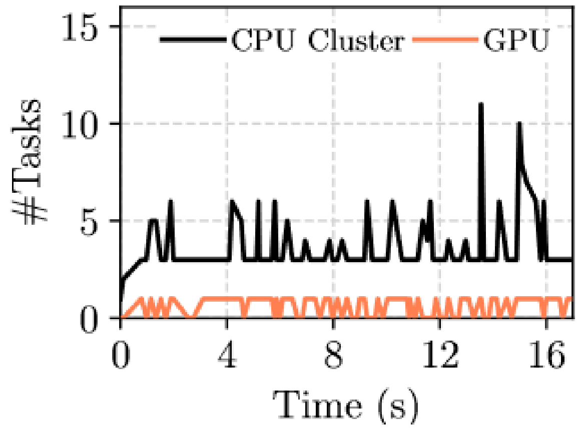
Collaborative Application: *rsct* from Chai

Research state-of-the-art



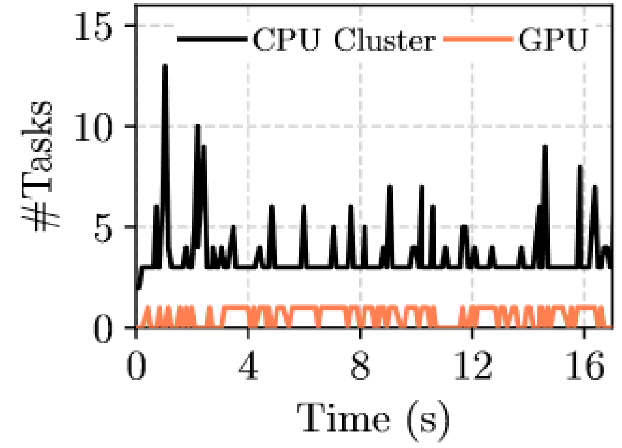


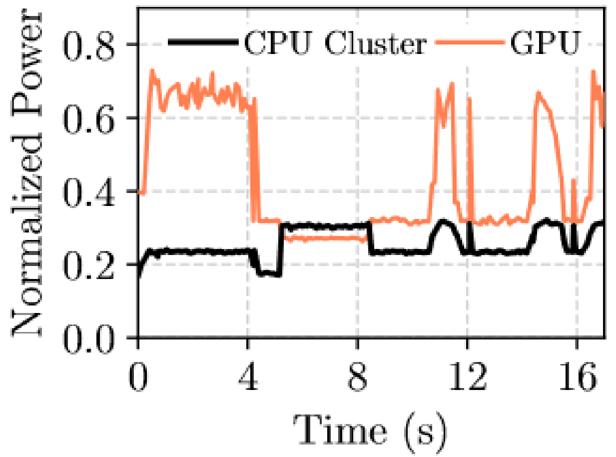
Tangram



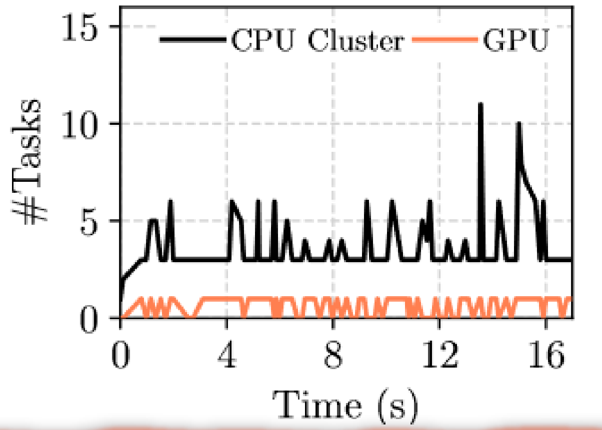
Collaborative Application: *rsct* from Chai

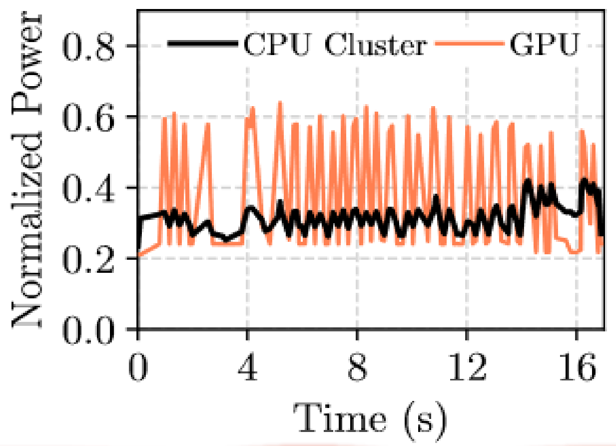
Research state-of-the-art





Tangram





Fine-grained power re-assignment

Summary

Summary

- Heterogeneous hardware is ubiquitous
  - Need to rethink resource control for efficiency and safety

Summary

- Heterogeneous hardware is ubiquitous
  - Need to rethink resource control for efficiency and safety
- Tangram is a fast, coordinated, and modular resource control framework
  - Uses formal control, multiple engines per controller, and a standard interface

Summary

- Heterogeneous hardware is ubiquitous
  - Need to rethink resource control for efficiency and safety
- Tangram is a fast, coordinated, and modular resource control framework
  - Uses formal control, multiple engines per controller, and a standard interface
- Prototype shows significant benefits
  - 41% lower EDP over state-of-the-art research
  - Modular

Summary

- Heterogeneous hardware is ubiquitous
  - Need to rethink resource control for efficiency and safety
- Tangram is a fast, coordinated, and modular resource control framework
  - Uses formal control, multiple engines per controller, and a standard interface
- Prototype shows significant benefits
  - 41% lower EDP over state-of-the-art research
  - Modular

Also in the paper:

Resource control from leading chip-makers
Detailed design of Tangram
Prototype implementation
Complete evaluation

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD Arrow logo, Radeon, Ryzen, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other names used herein are for identification purposes only and may be trademarks of their respective companies.

Image sources

Tangram image on the cover slide by Unknown Author is used under license by CC BY

Kaby Lake G image (Slide 2) by Fritzchens Fritz is used from public domain

Circuit background (Slide 2) from http://creativity103.com is used under license by CC BY