Configuration Management Challenges

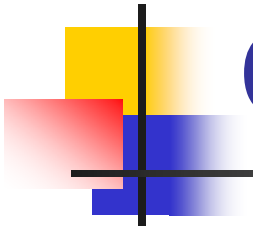
Composition of Adaptive Components

Tarek Abdelzaher

Dept. of Computer Science

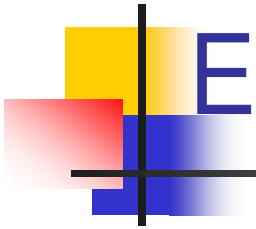
University of Illinois at Urbana Champaign

Composition of Adaptive Components



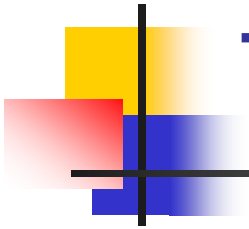
- Challenge:
  - Larger systems composed of a larger number of components
  - Modularity and separation of concerns → components are designed independently
  - Autonomy → adaptive behavior in many components
  - Composition of adaptive components can have unexpected adverse interactions
  - Configuration challenge: How to compose well-behaved systems from large numbers of adaptive components and prevent adverse interactions?

Datacenter Energy Management Example: *Policy Coordination*



Many policies are used to save energy:

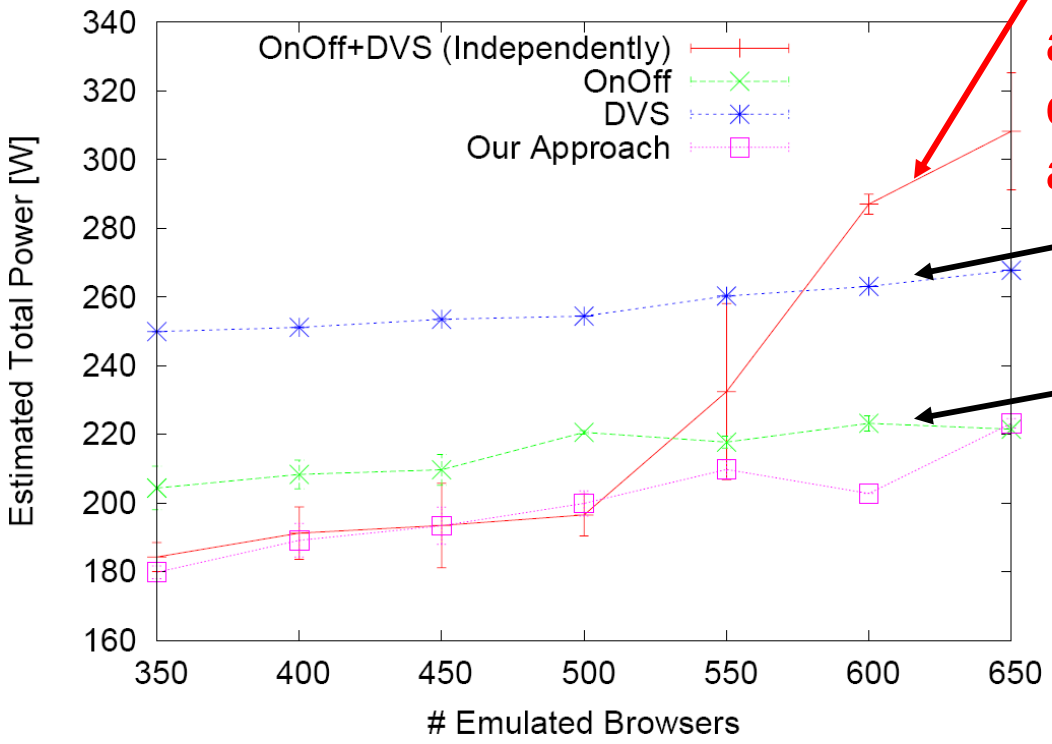
- Computing-side:
  - Sensors: Machine utilization, Delay, Throughput, ...
  - Policies: DVS-based, consolidation, turning machines On/Off
- Cooling-side:
  - Sensors: Temperature, air flow, ...
  - Policies: Control air-conditioning units, fans, ...
- Energy bill is 20-50% of total profit



The Problem

Uncoordinated adaptation of multiple knobs can lead to instability or poor efficiency

Example
(A Tale of Two Policies)



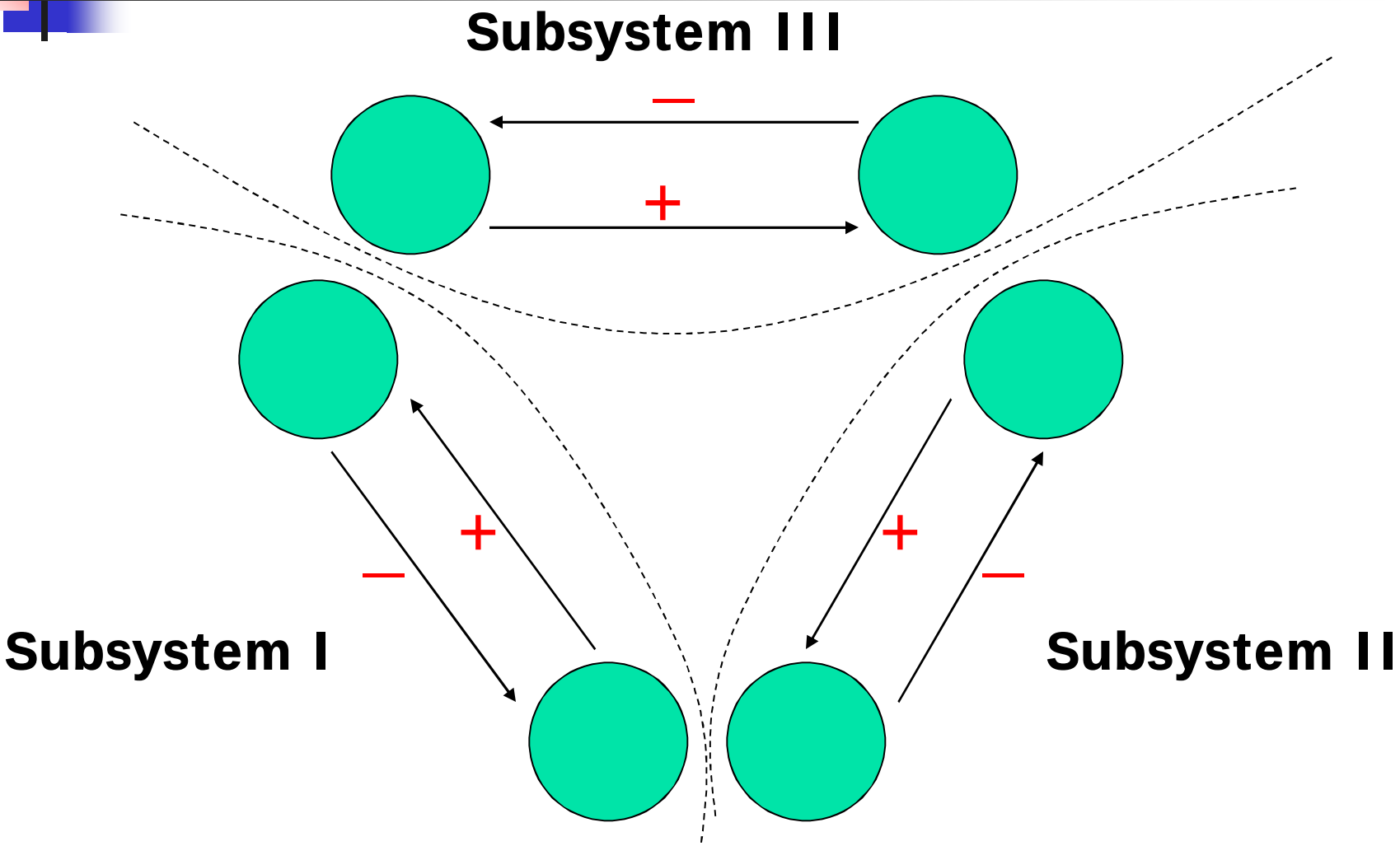
DVS + On/Off without co-adaptation: more energy consumption than DVS or On/Off alone!

DVS alone

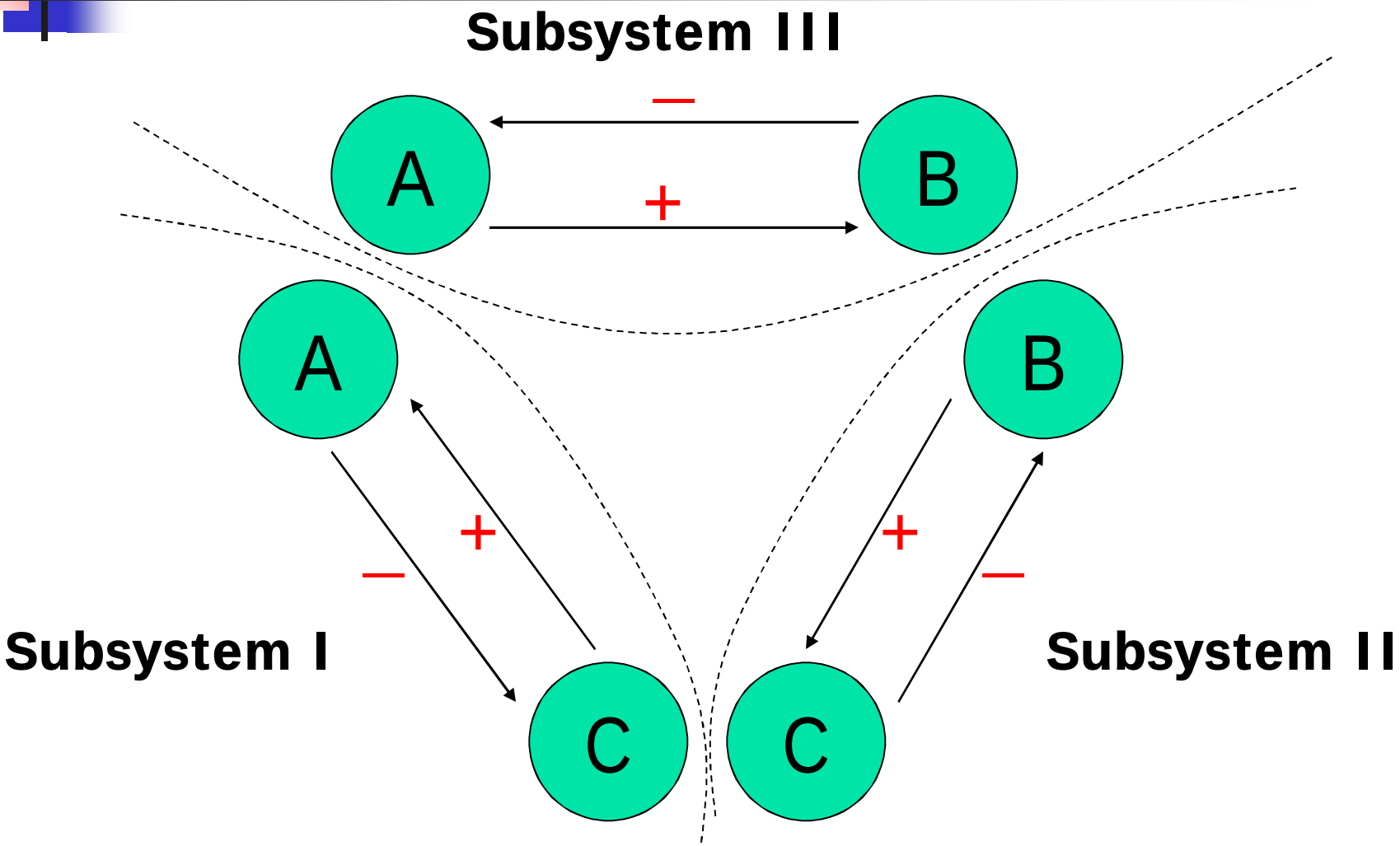
On/Off alone

Empirical measurements from a 30-machine 3-tier testbed of a shopping site

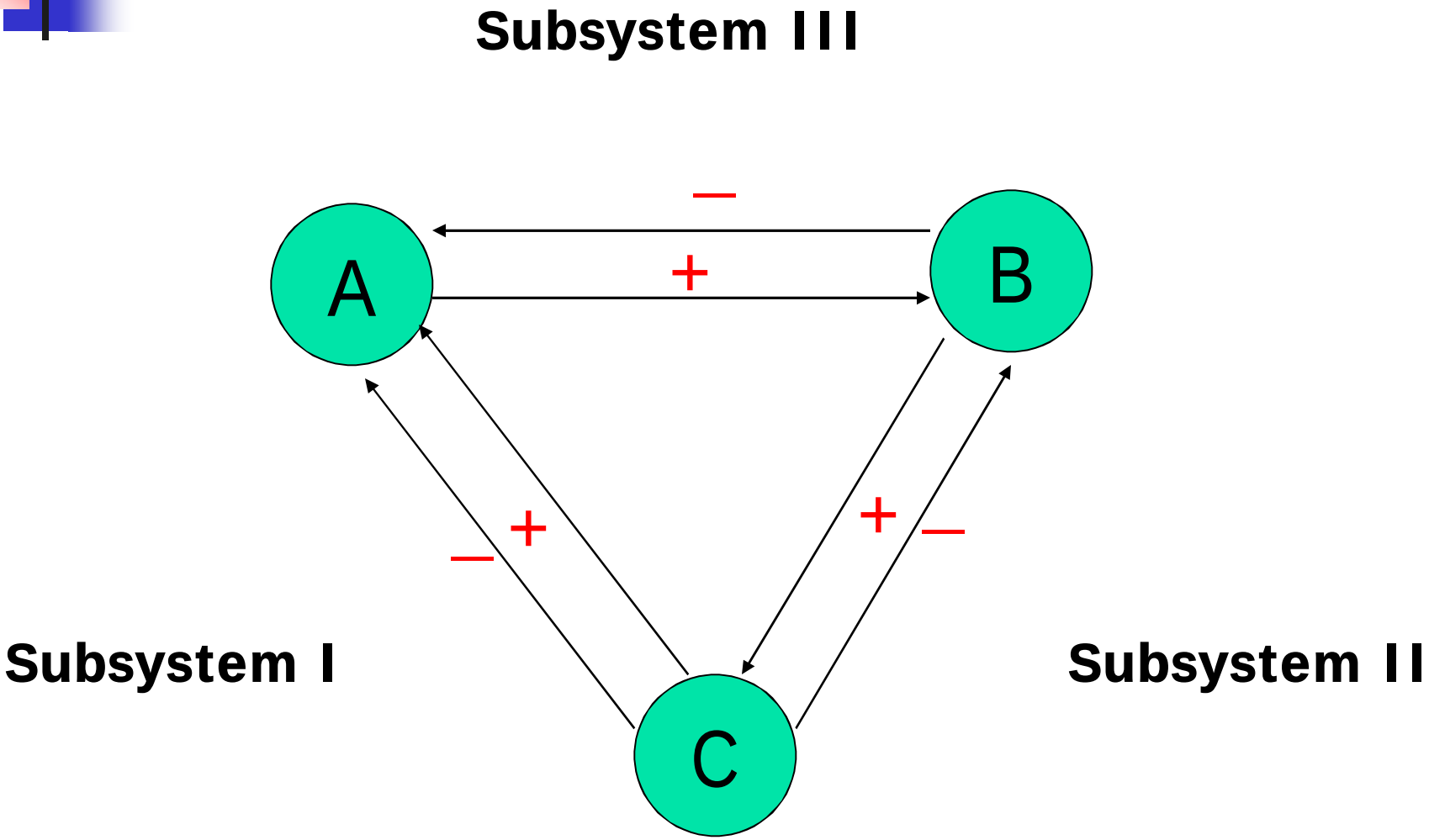
Preliminary Insights: Composition of Adaptive Systems



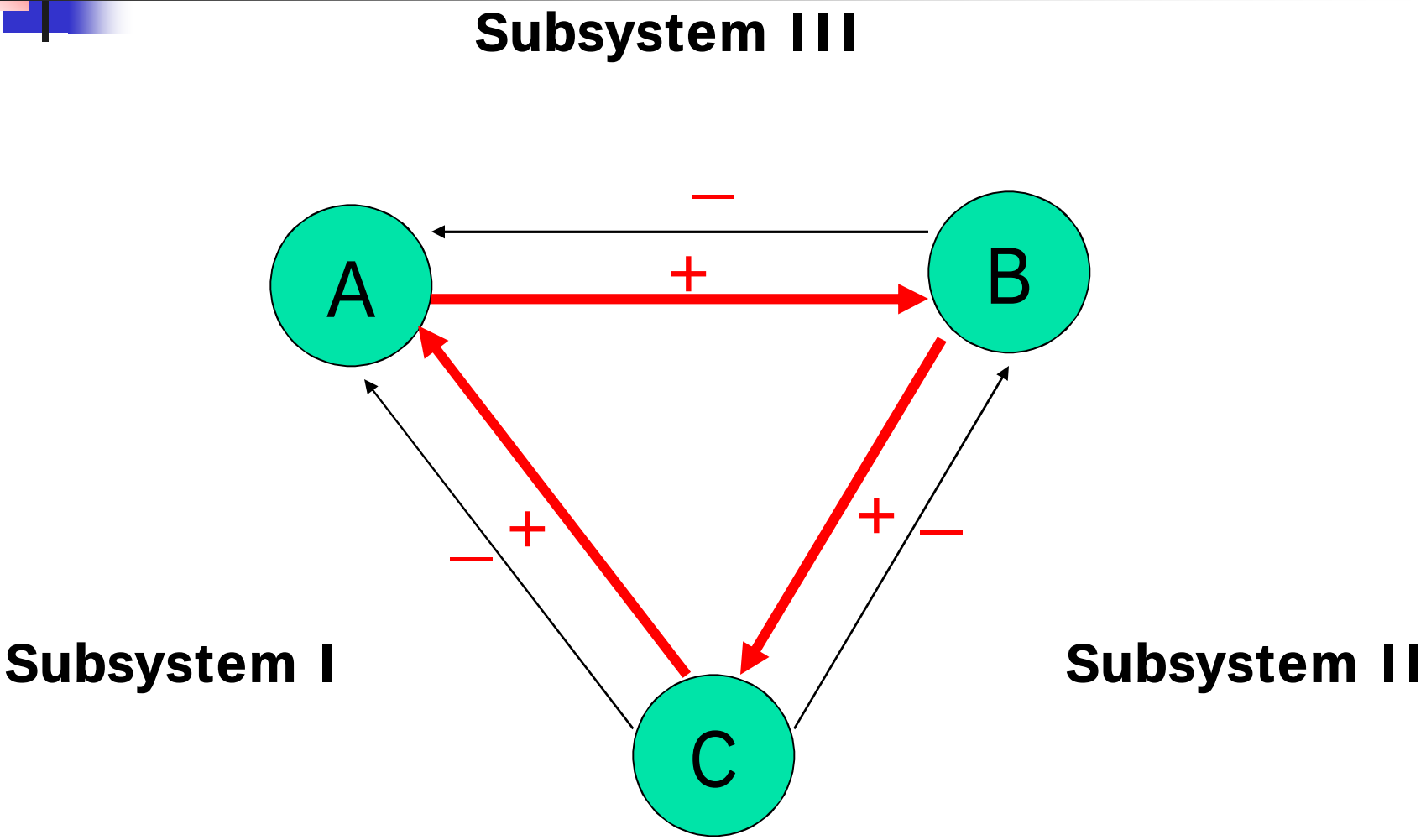
Preliminary Insights: Composition of Adaptive Systems



Preliminary Insights: Composition of Adaptive Systems



Preliminary Insights: Composition of Adaptive Systems



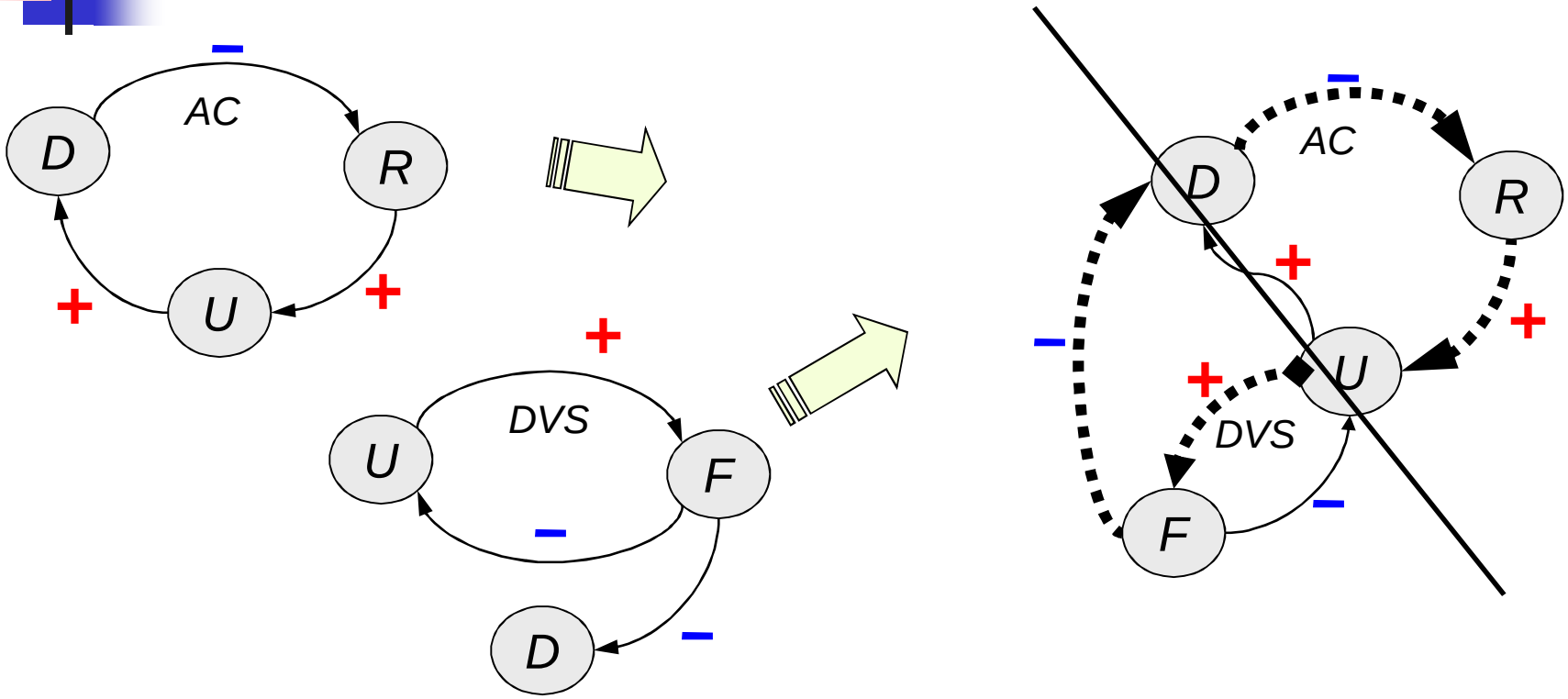
Composability of Adaptive Behavior

- Many adaptive policies may perform well in isolation, but conflict when combined
- Example:

Loop

- Excess capacity → DVS lowers frequency → Utilization increases
- High utilization → Another machine is turned on

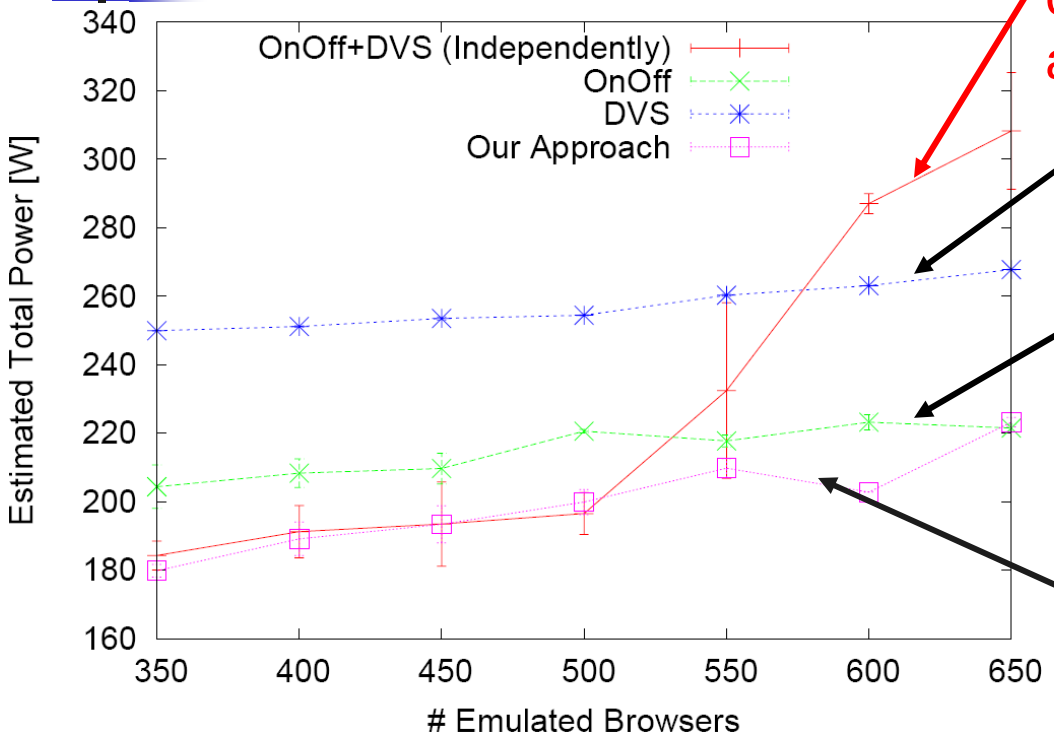
Example: DVS-enabled QoS-aware Web Server



Individual adaptation loop for each policy is stable (negative)

Combined together, unstable positive loop across policy boundaries

Evaluation



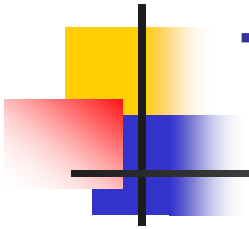
DVS + On/Off without co-adaptation: more energy consumption than DVS or On/Off alone!

DVS alone

On/Off alone

Joint tuning successfully resolves conflict!!

How to prevent adverse interactions in large systems?



Configuration Troubleshooting Tools: *Control-theoretic*

Software dynamics offer control stability challenges:

- Output of component A depends on past output of B (delay)
- Output of component A depends on integral of B (e.g., data queue is an integral of difference in data flow rates)
- Filters (averaging, moving average)
- Sampling
- Dynamics can be modeled in a control-theory context to determine stability

Troubleshooting: *Data Mining*