Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Integrating Adaptive Components

by Jin Heo, Dan Henriksson, Xue Liu, Tarek Abdelzaher (2007) [1] [2]

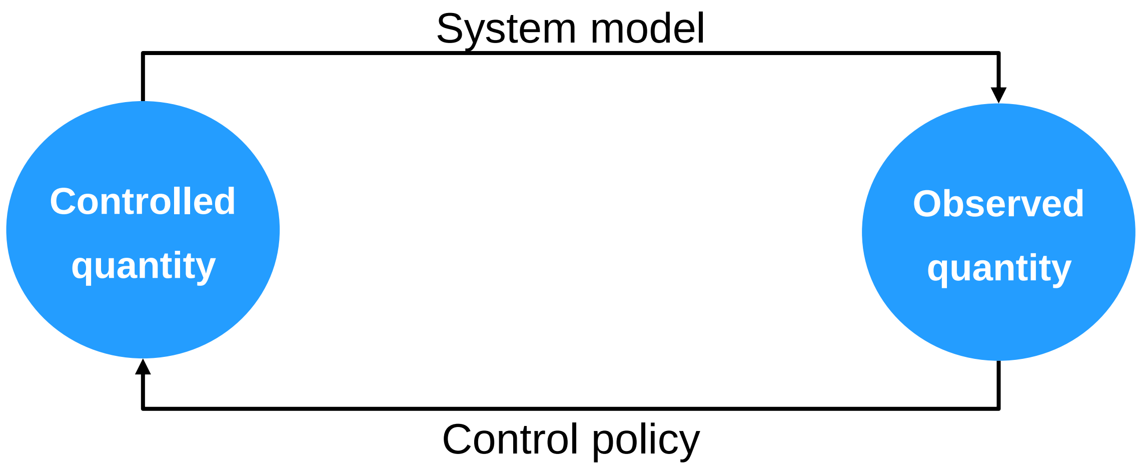
Gábor Sörös
Distributed Systems Seminar
17.05.2011

Problem

- Timing-sensitive and performance-sensitive systems (for example, data centers) are very complex
  - Manual tuning is imprecise, costly, and time-consuming
  - Hence: automation
- Automation calls for adaptive capabilities
  - Hence: adaptive components
  - Self-managing, self-calibrating, self-healing, self-tuning SW blocks
- Composition of locally stable adaptive components can lead to a globally unstable system

How to identify potential incompatibilities?

Adaptive components – Feedback loops

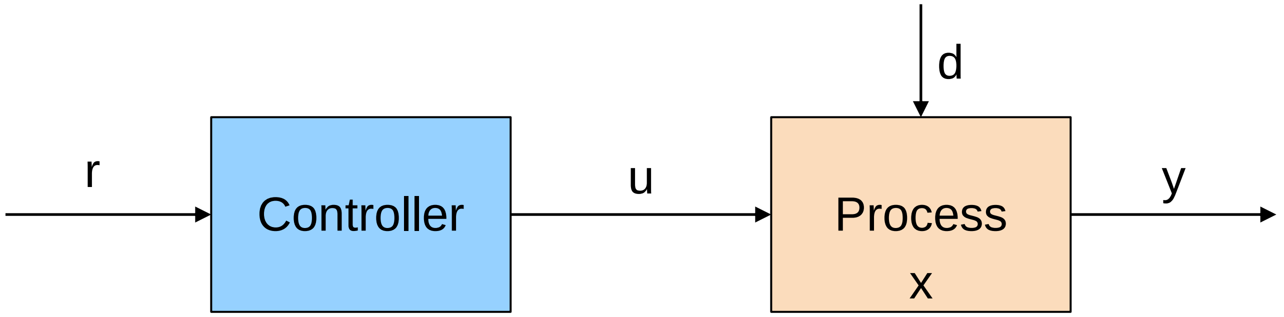


FEEDBACK CONTROL

Stability – Some insights

r: reference value
d: disturbance(s)
u: control variable(s)
x: state variable(s)
y: output variable(s)

Open-loop vs. closed loop control



Open-loop control:

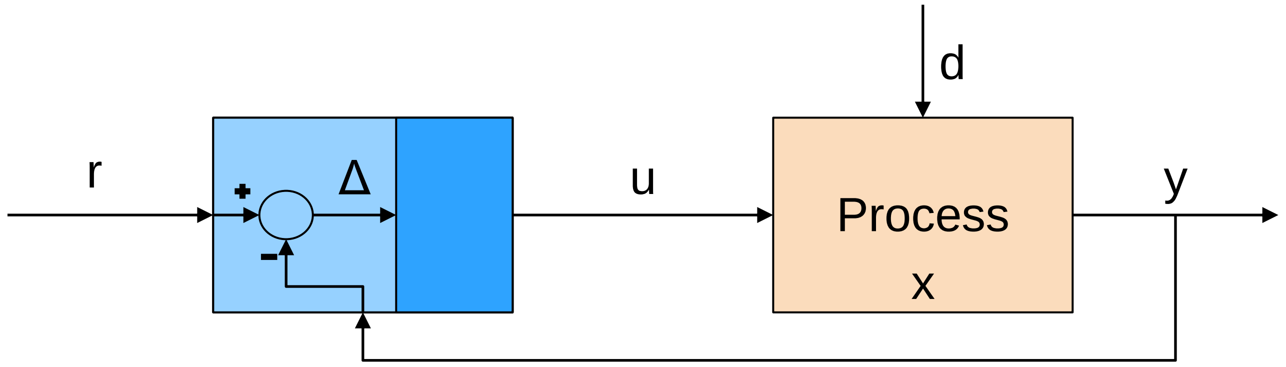
- without feedback
- the effects of known disturbances are compensated (unknowns not)
- in case of a stable process always stable

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Stability – Some insights (2)

r: reference value
d: disturbance(s)
u: control variable(s)
x: state variable(s)
y: output variable(s)

Open-loop vs. closed loop control



Closed-loop control:

- with feedback
- slower
- the effects of disturbances AND parameter changes of the process are fully compensated
- the loop can become unstable even in case of a stable process

Stability – Some insights (3)

Positive vs. negative feedback

- Positive feedback tends to strengthen the event that caused it (e.g., nuclear reaction)
- Negative feedback tends to reduce the input signal that caused it (e.g., heating control)

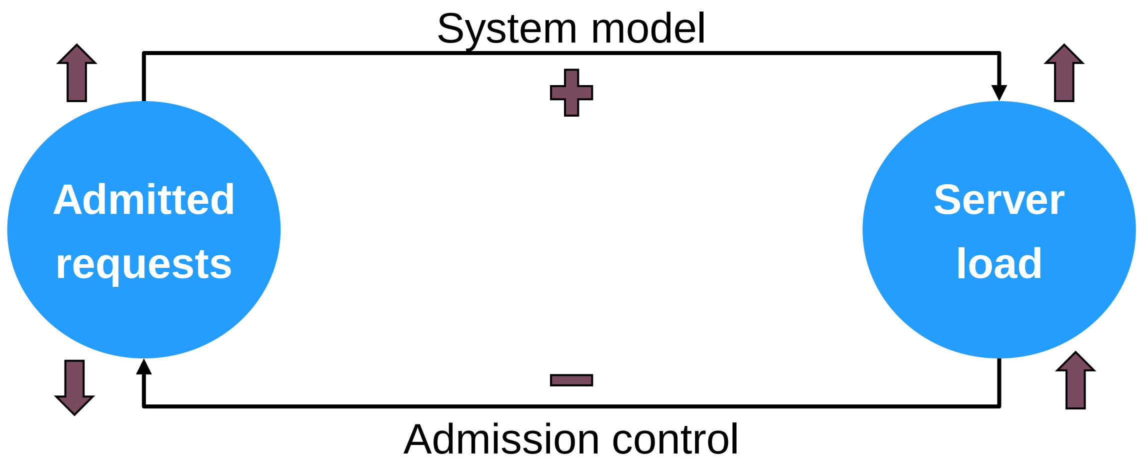
All stable feedback is negative

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

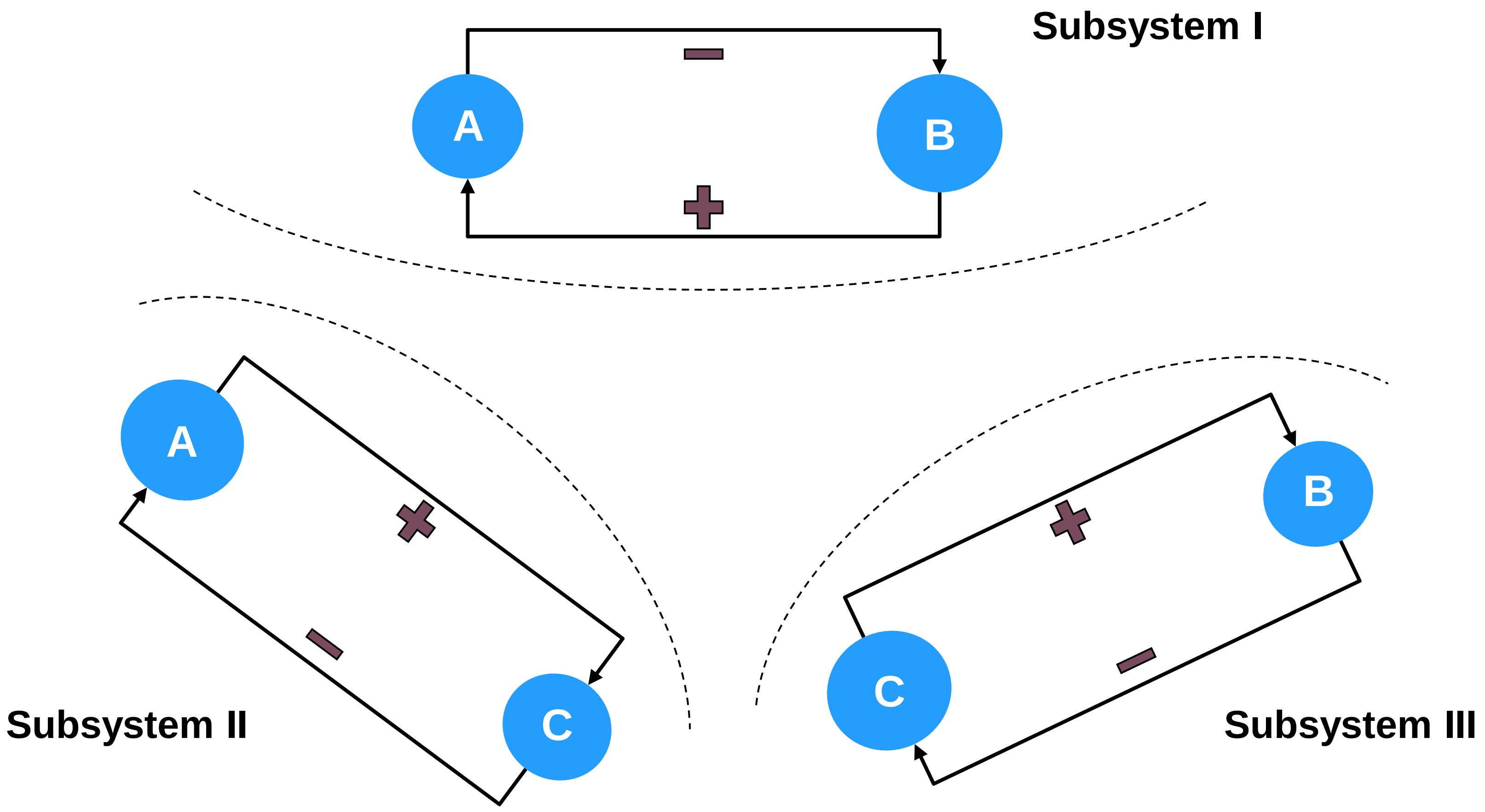
COMPOSITION OF ADAPTIVE POLICIES

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Adaptive policy – a feedback loop

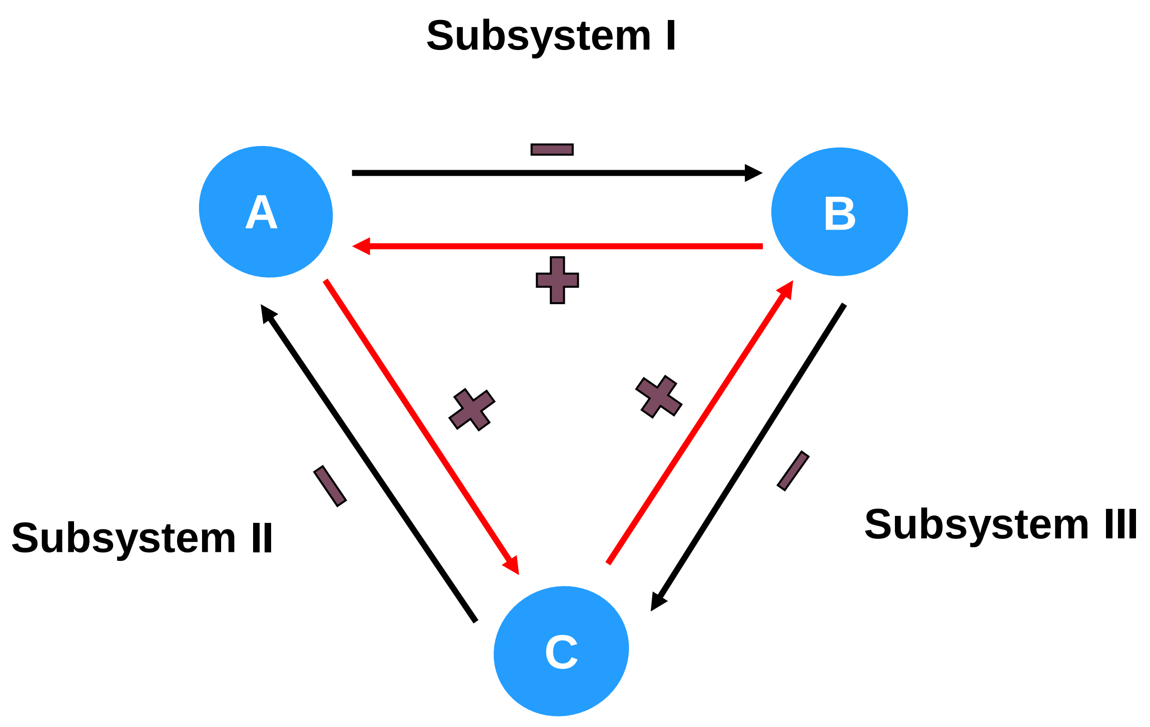


Composition of adaptive subsystems



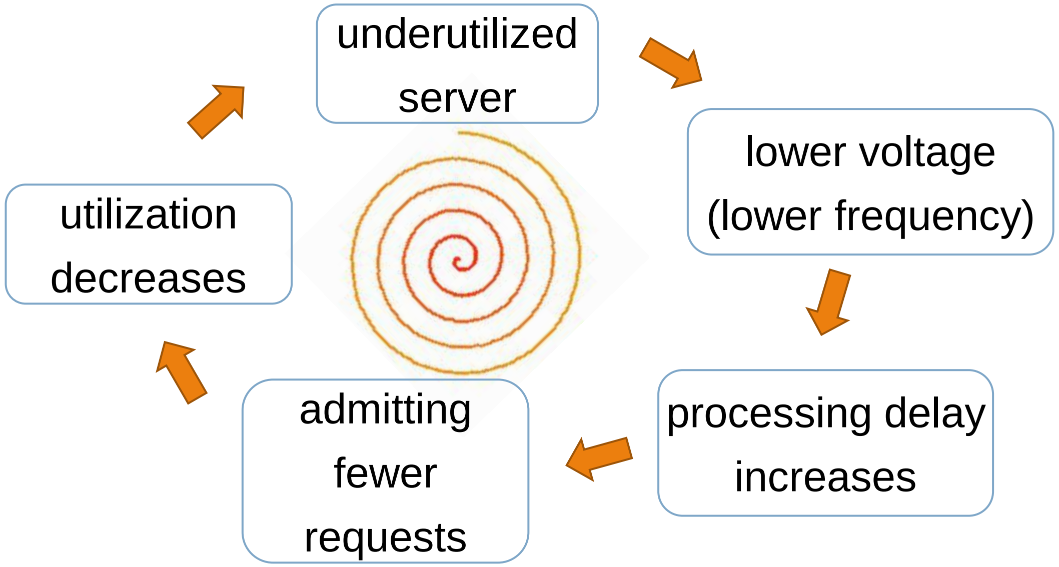
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Composition of adaptive subsystems



Adverse policy interactions – an example of a Web server

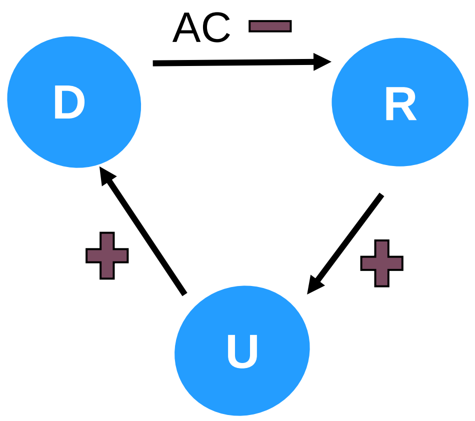
- Dynamic Voltage Scaling (DVS)
  - in an underutilized server, DVS decreases frequency, hence increasing delay
- Admission Control (AC)
  - responds to increased delay by admitting fewer requests



DETECTION OF POSSIBLE CONFLICTS

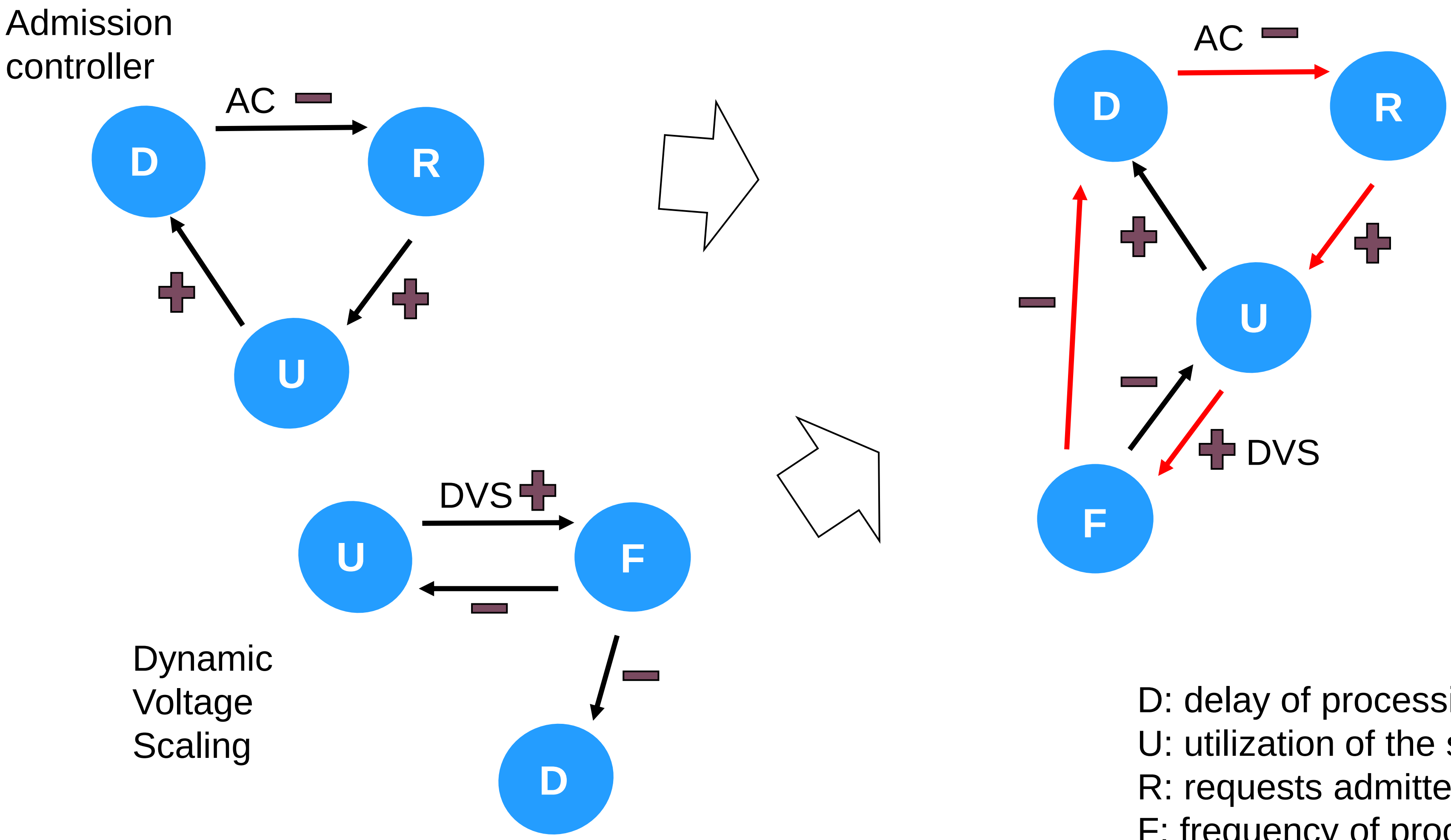
Adaptation Graph Analysis

- Nodes – key variables in the system
- Arcs – direction of causality
  - "+" arcs – changes in the same direction
  - "-" arcs – changes in the opposite direction
  - normal arcs – natural relationship
  - policy arcs – programmed behavior
- Sign of a directed circle – multiplication of the signs of all edges



Adaptation graphs determine which adaptive policies conflict (if they do)

Checks for potential incompatibility

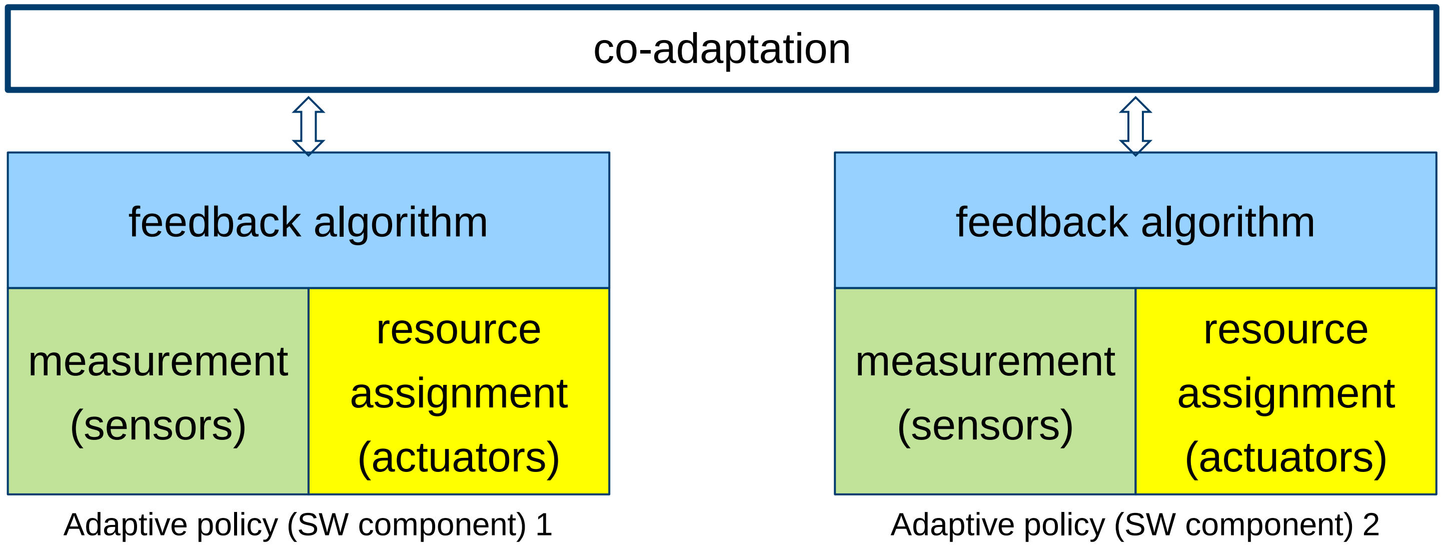


D: delay of processing
U: utilization of the server
R: requests admitted
F: frequency of processing

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DESIGN METHODOLOGY FOR INTEGRATING ADAPTIVE POLICIES

Co-adaptation


co-adaptation
feedback algorithm
measurement (sensors)
resource assignment (actuators)
Adaptive policy (SW component) 1
feedback algorithm
measurement (sensors)
resource assignment (actuators)
Adaptive policy (SW component) 2

- co-adaptation guides the design of a combined module – it outputs jointly optimized knob settings that increase utility
- constrained optimization (necessary condition) + feedback control

Step 1: Casting the objective

- Find a common objective function – minimize cost or maximize utility
  - Same objective → ok
  - Different objective functions → Find common function

Step 2: Formulating the optimization problem

- Decision variables – settings of adaptation knobs
- Subject to two types of constraints:
  - Performance specifications
  - Resource constraints

$$\begin{aligned} \min_{x_1,\ldots,x_n} \quad & f(x_1,\ldots,x_n) \\ \text{subject to} \quad & g_j(x_1,\ldots,x_n) \leq 0, \quad j = 1,\ldots,m \end{aligned}$$

where f is the common objective function,
x_i are the set of adaptation knobs for policy i (these sets may overlap)
g_j are the constraints

Step 3: Necessity conditions for optimality

- Lack of accurate model for computing systems
- Augmented by feedback loops to move closer to the point that increases utility
- Use the Karush-Kuhn-Tucker (KKT) optimality condition

$$\Gamma x_i \leftarrow \frac{\partial f(x_1,\ldots,x_n)}{\partial x_i} + \sum_{j=1}^{m} \nu_j \frac{\partial g_j(x_1,\ldots,x_n)}{\partial x_i} = 0$$

- Necessary condition:

$$\Gamma_{x_1} = \ldots = \Gamma_{x_n}$$

  - Define $\Gamma_x = (\Gamma_{x_1} + \ldots + \Gamma_{x_n})/n$

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Step 4: Feedback control

- Periodic measurements to estimate Γx_i

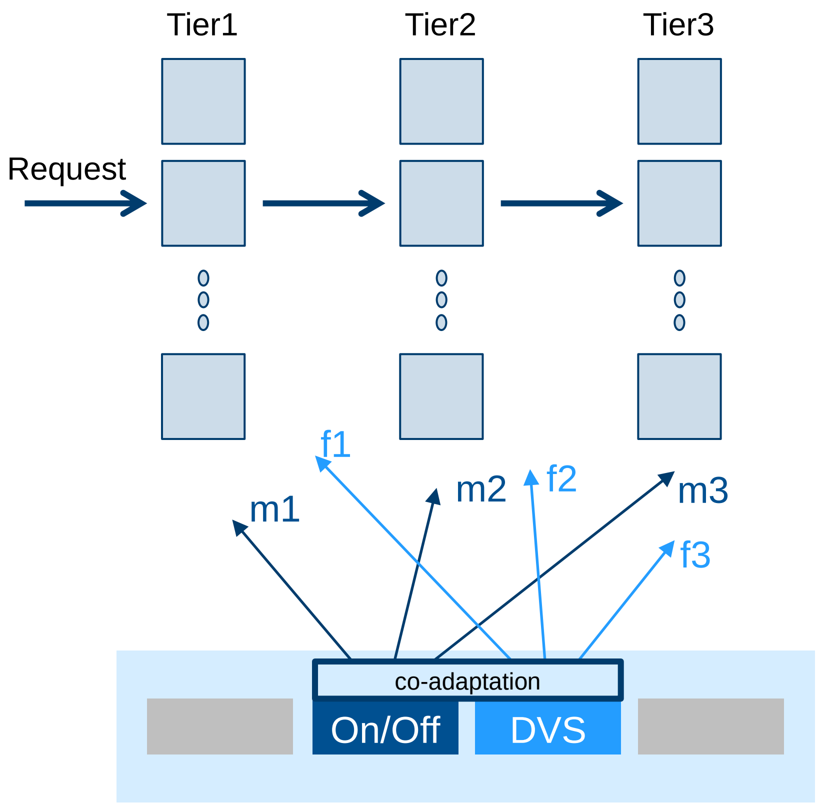
- Try to meet the necessary condition $\Gamma x_1 = \ldots = \Gamma x_n$ by hill climbing
  - Pick one with the largest or smallest value of Γx_i
  - Search through the neighboring knob settings (values of x_i)
    - Reduce the error $\Gamma x - \Gamma x_i$
    - Maximum increase in utility

x_i are the sets of adaptation knobs for policy i

A SERVER FARM CASE STUDY

Energy minimization in server farms

Energy minimization in server farms

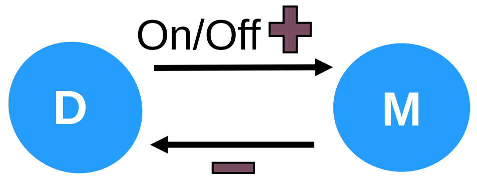


- Two adaptive policies in conflict
  - On/Off policy
  - DVS policy
- Co-adaptation finds the knob settings (m1,m2,m3,f1,f2,f3) such that energy consumption is reduced

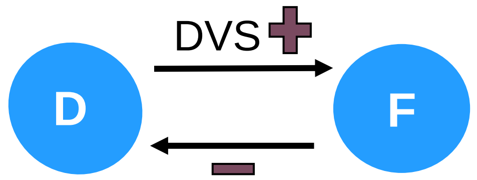
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

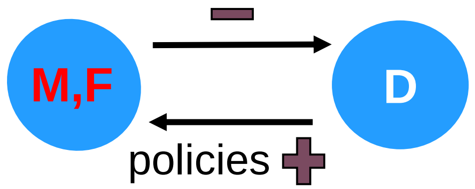
Step 0: Incompatibility detection

On/Off policy



Dynamic Voltage Scaling policy





Combined together, the loop is still negative.

But potentially unstable: possible interference in the control of D!

Step 1: Casting the objective

- In this case, both policies have the objective to minimize energy consumption.
  - Hence, energy is the common cost function

Step 2: Formulating the optimization problem

- Decision variables:
  - Number of active machines in each tier
  - Frequency level of the machines in each tier
    (or, equivalently, the utilization of each tier)
- Approximations:
  - M/M/1 queuing model
  - Number of computers in each tier is a real number
    (relaxation of the integer problem)
  - Power consumption is equal for all machines in a tier
  - Machines are load-balanced (same U, same F) inside the tier

Step 2: Formulating the optimization problem

$$P_i(f_i) = A_i \cdot f_i^p + B_i$$

Power estimation of a machine at tier i

$$U_i = \frac{\lambda}{\mu} = \frac{\lambda_i / m_i}{f_i} = \frac{\lambda_i}{m_i f_i}$$

Queuing equation using number of machines and arrival rate

$$P_i(U_i, m_i) = A_i \cdot \left(\frac{\lambda_i}{U_i m_i}\right)^p + B_i = \frac{A_i \lambda_i^p}{U_i^p m_i^p} + B_i$$

Power estimation function of a machine at tier i

$$\min_{U_i \geq 0,\ m_i \geq 0} \quad P_{tot}(U_i, m_i) = \sum_{i=1}^{3} m_i \left(\frac{A_i \lambda_1^3}{U_i^3 m_i^3} + B_i \right)$$

$$\text{subject to} \quad \sum_{i=1}^{3} \frac{m_i}{\lambda_i} \cdot \frac{U_i}{1 - U_i} \leq K, \qquad \sum_{i=1}^{3} m_i \leq M$$

Formulation of the problem:

find best composition of

$(m_1, m_2, m_3, U_1, U_2, U_3)$

Constraint: end-to-end delay

Constraint: No. of machines in a tier

Step 3: Necessity conditions for optimality

- Karush-Kuhn-Tucker condition

$$\frac{\lambda_1^4(1-U_1)^2}{m_1^3U_1^4} = \frac{\lambda_2^4(1-U_2)^2}{m_2^3U_2^4} = \frac{\lambda_3^4(1-U_3)^2}{m_3^3U_3^4}$$

$$\Gamma(m_1, U_1) = \Gamma(m_2, U_2) = \Gamma(m_3, U_3)$$

Try to find (m_1, m_2, m_3, U_1, U_2, U_3) tuple that balance this condition

Step 4: Feedback control

- Goal: to balance the necessary condition in the direction to reduce energy consumption
- When delay constraint violated
  - Pick the most overloaded tier – the one with lowest $\Gamma(m_i,U_i)$
- Choose (m_i,U_i) pair that makes the error within a bound and yields the lowest total energy
  - Error = $\Gamma_x - \Gamma(m_i,U_i)$, where Γ_x is the average of $\Gamma(m_i,U_i)$

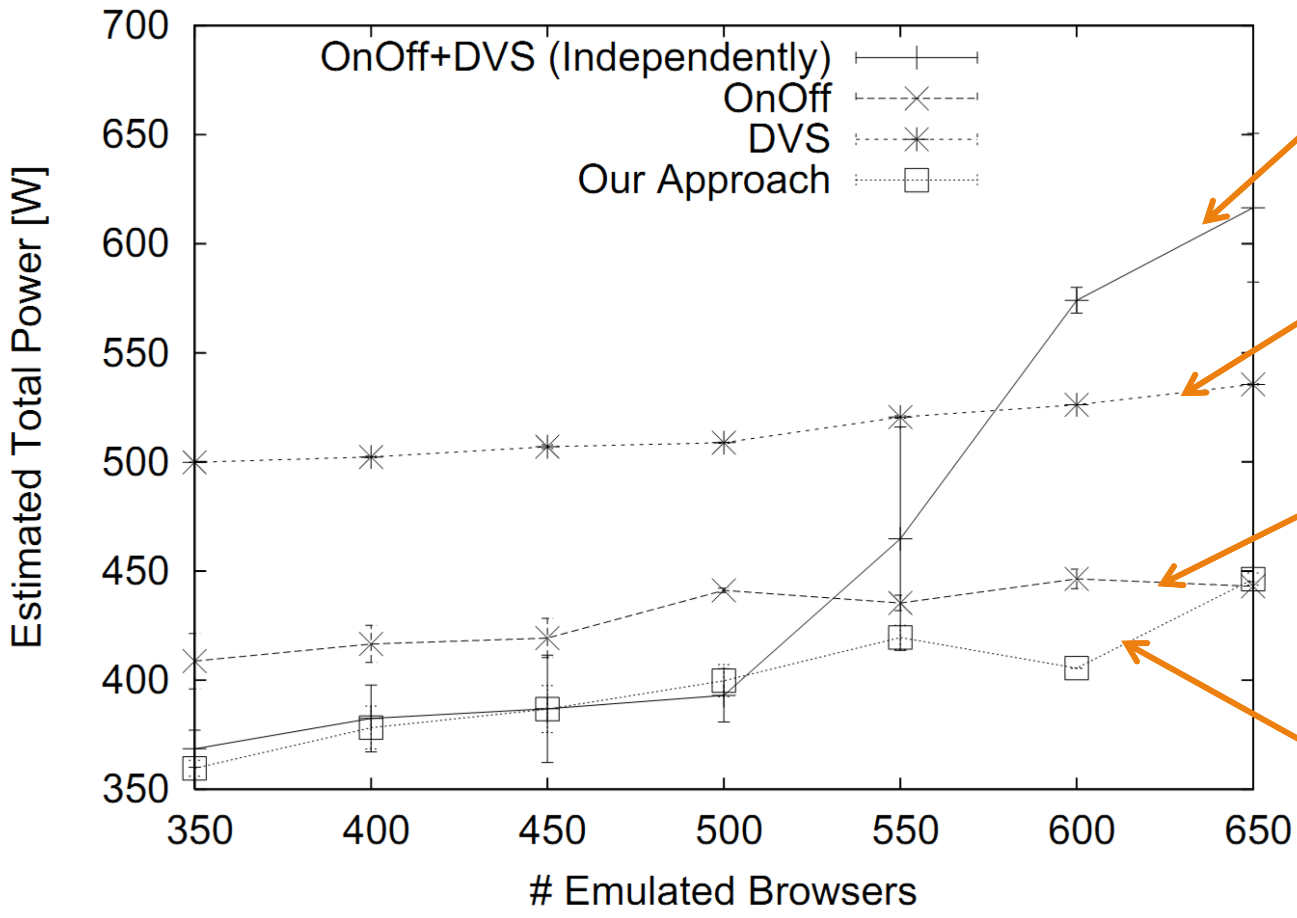
Local control decisions, based on a globally obtained system snapshot

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

EXPERIMENTAL RESULTS

Evaluation on a 3-tier server farm

Experimental results



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

DVS alone

On/Off alone

Together with co-adaptation: successfully resolves conflict!

Summary

This paper

- presents a simple mechanism for identifying potentially adverse interactions between policies at component composition time,
- describes a methodology for designing co-adaptation, where the adverse interaction is eliminated, and
- evaluates the method with a Web server farm case-study.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Questions?

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

References

[1] Jin Heo, Dan Henriksson, Xue Liu and Tarek Abdelzaher, “Integrating Adaptive Components: An Emerging Challenge in Performance-Adaptive Systems and a Server Farm Case-Study,” *The 28th IEEE Real-Time Systems Symposium (RTSS 07)* December 3-6, 2007, Tucson, Arizona, USA

[2] This presentation contains slides from Tarek Abdelzaher

http://www.docstoc.com/docs/38789686/Integrating-Adaptive-Components-An-Emerging-Challenge-in

[3] Tarek Abdelzaher, “Composition and Scaling Challenges in Sensor Networks: an Interaction-Centric View,” *Chapter 1 in S. Nikoletseas, J.D.P. Rolim (eds.) – Theoretical Aspects of Distributed Computing in Sensor Networks, Springer, 2011*