A CONTROL-BASED FRAMEWORK FOR SELF-MANAGING DISTRIBUTED COMPUTING SYSTEMS

S. Abdelwahed and S. Neema

EECS Department

Vanderbilt University

N. Kandasamy

ECE Department

Drexel University

WOSS, Newport Beach, CA, October 31, 2004

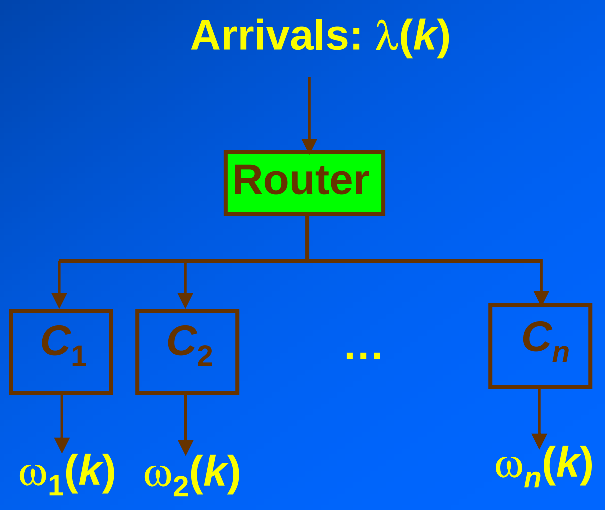
OUTLINE

- Introduction
- Online control concepts
- Case studies
- Ongoing work

INTRODUCTION

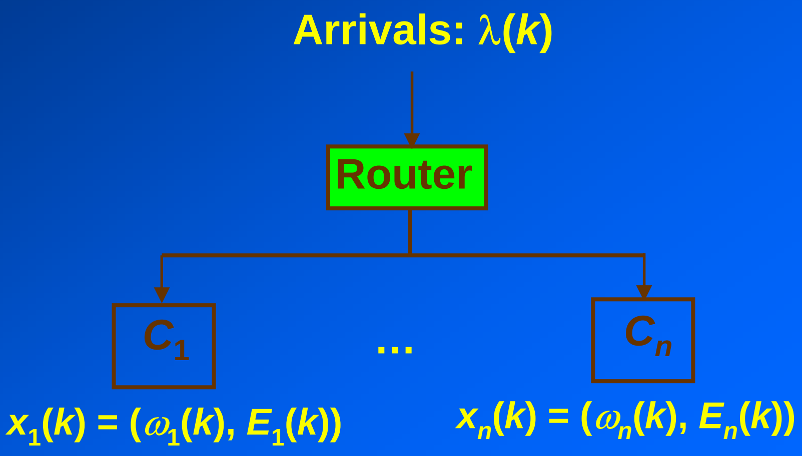
- Research goals
  - Design *self-optimizing* computer systems that continually aim to improve performance and efficiency
  - Maintain desired quality-of-service (QoS) under dynamic operating conditions
- Proposed approach
  - Develop an online control framework to design such systems
- Resource management applications
  - Energy-efficient computing (ICAC 04, RTAS 04)
  - Load balancing

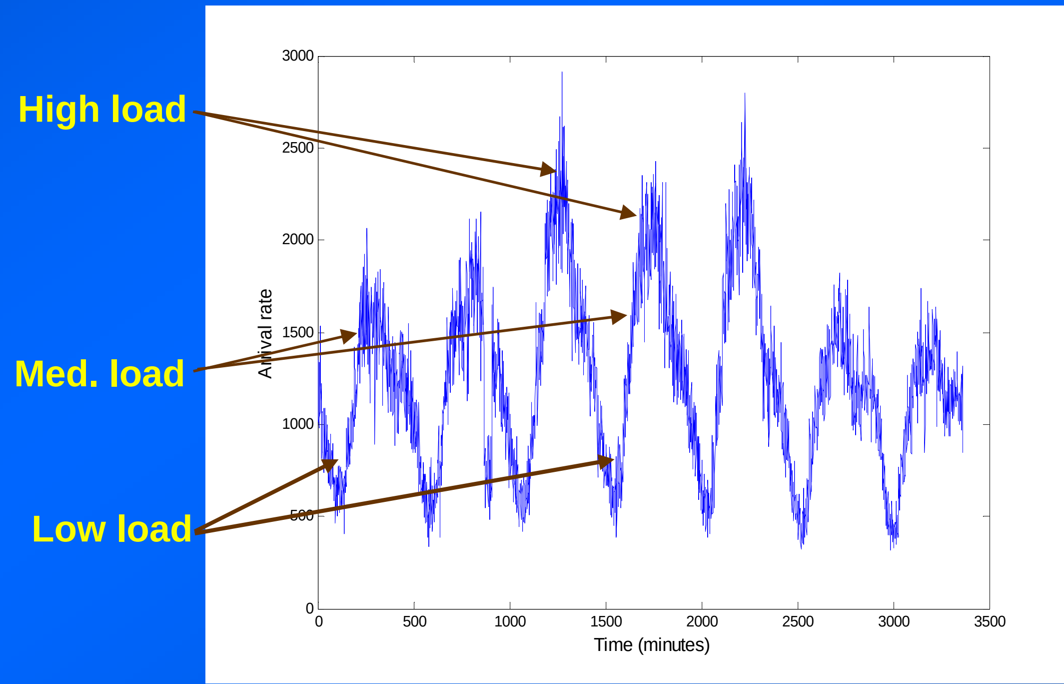
MOTIVATING EXAMPLE



- System model
  - Router distributes requests to individual computers
  - The CPU on each computer is DVS-capable (can operate at multiple frequencies)
- Energy costs
  - A base cost for each operating computer
  - Dynamic CPU power consumption
- QoS goal
  - Achieve an average latency for requests, as specified by the SLA
- Operate cluster in energy-efficient fashion

EXAMPLE (Contd.)





- Goal:

$$\sum_k \left\| x(k) - x^* \right\|_Q + \left\| u(k) \right\|_R$$

$$\text{Subject to:}$$

$$x(k+1) = f(x(k), u(k), \lambda(k))$$

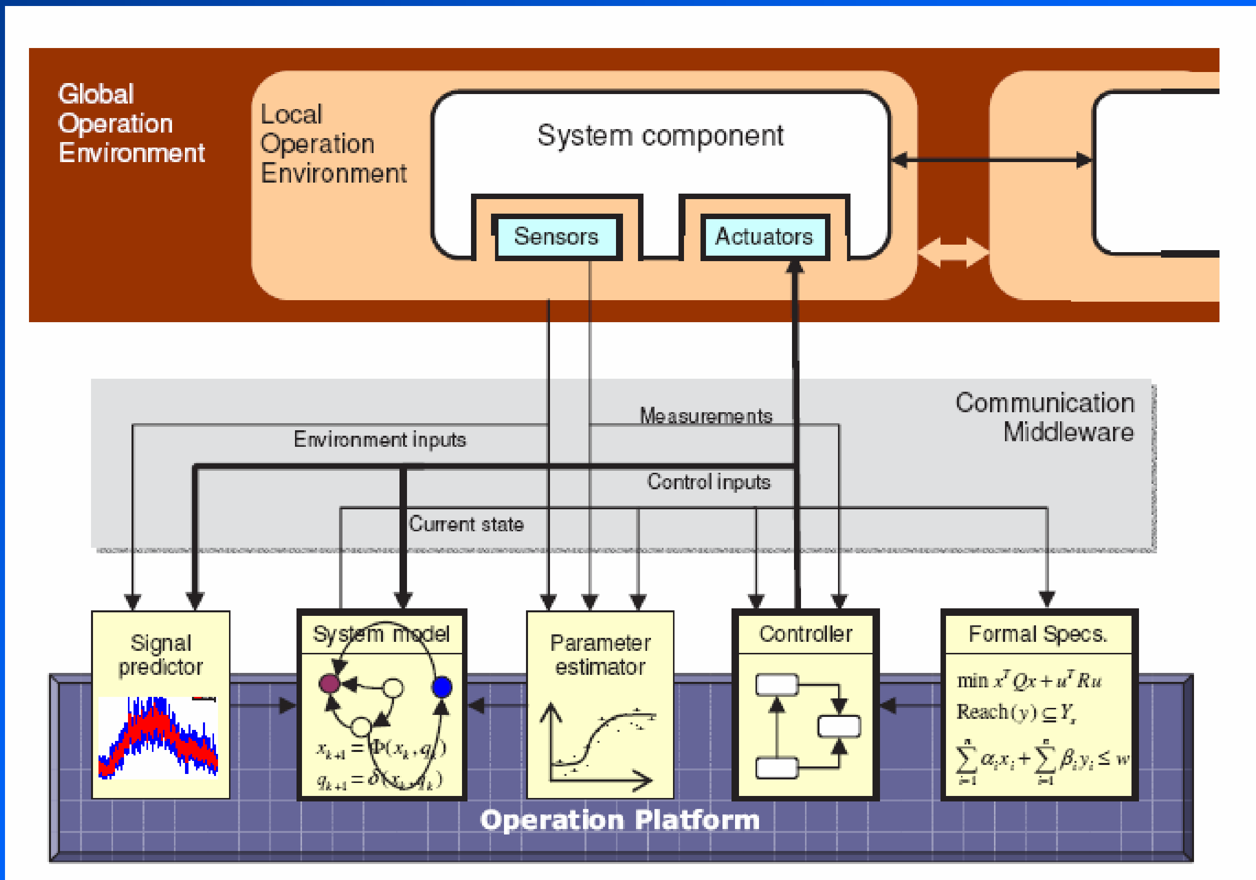
$$h(x(k)) \geq 0$$

- Cost function includes
  - Penalty for not meeting agreed latency
  - Corresponding energy cost

EXAMPLE (Contd.)

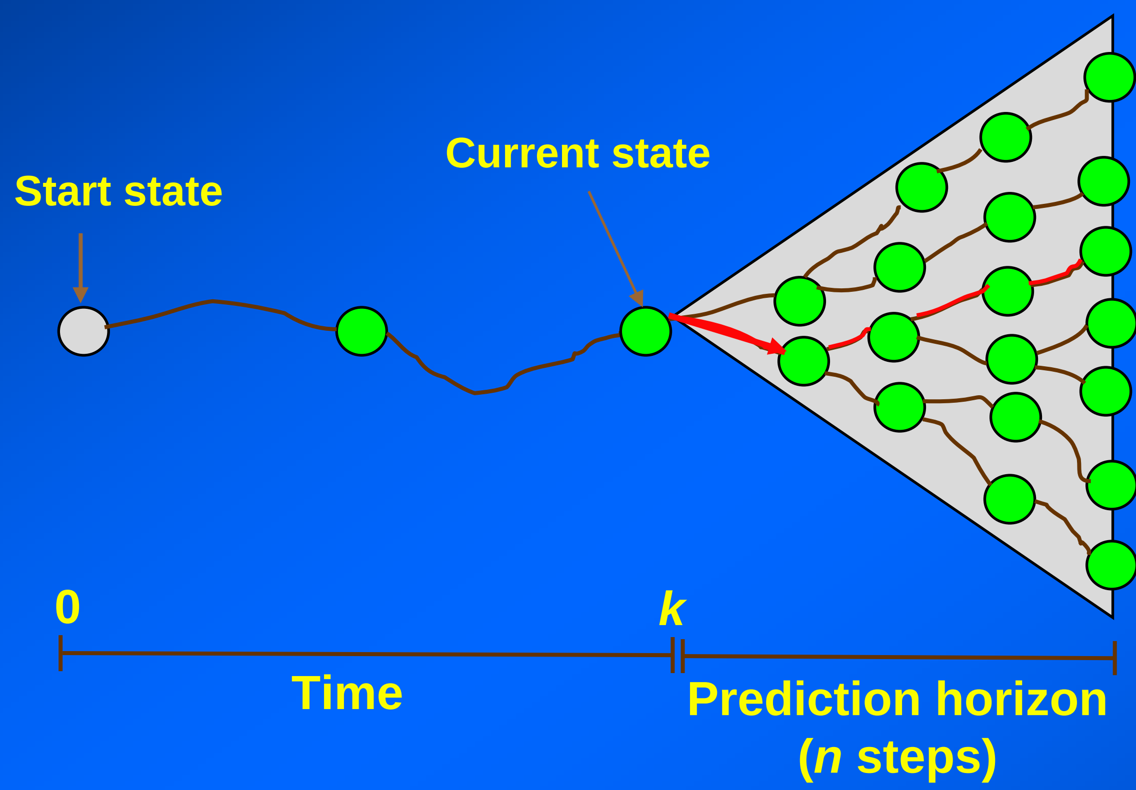
- Energy costs
  - Base cost for each operating computer (power supply, hard disks, etc)
  - Dynamic power consumed by the CPU
- Controllable variables
  - The number of computers operating at time *k*
  - The operating frequency of each CPU at time *k*
- Key characteristics of the control problem
  - Some control actions have long dead times; e.g., switching on a computer
  - System must be optimized over a discrete domain
  - Optimization must be performed under explicit constraints
- Technical idea:
  - Use *model predictive* or *receding horizon* control to optimize performance

ONLINE CONTROL FRAMEWORK



- Actions optimizing system behavior are derived over a limited prediction horizon
- Filter estimates future environment parameters using past values
- An *explicit* internal model captures system behavior
- Optimizer provides the appropriate control actions

CONTROL ALGORITHM

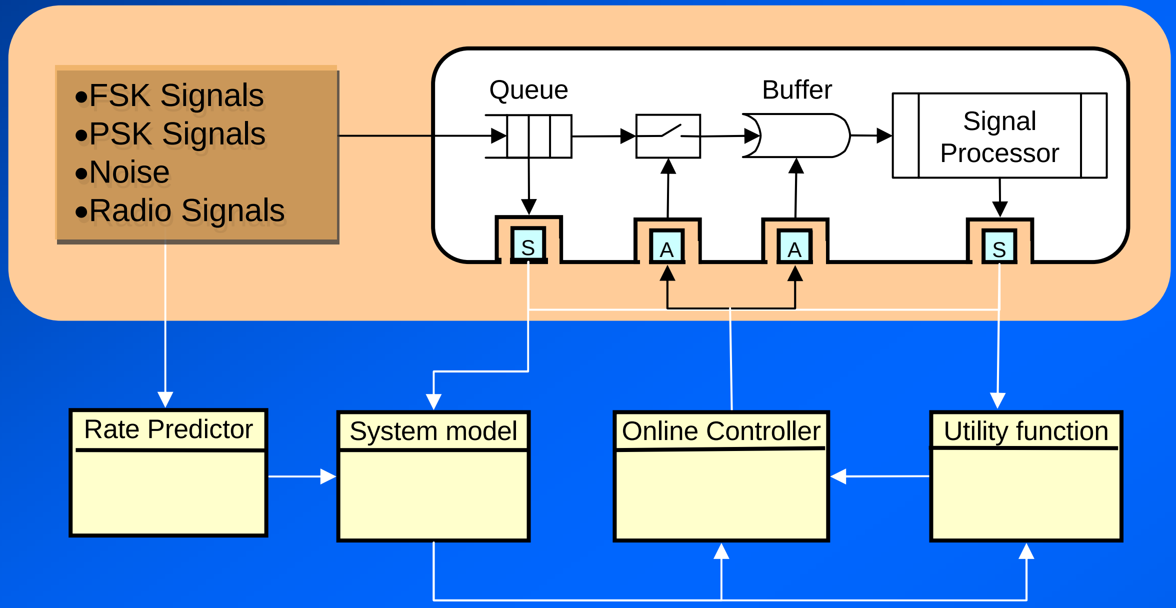


- Use behavioral model to estimate future system states over the prediction horizon
- Obtain the sequence of control inputs minimizing the cost function
- Apply the first control input in the sequence at time *k*; discard the rest

CONTROL ALGORITHM (Contd.)

- Advantages
  - Look ahead optimization compensates for dead times
  - Multi-variable optimization can be performed over discrete domain under explicit constraints
  - Model parameters can be learned and/or modified online
- Challenges
  - *Modeling*: Control quality is only as good as the model itself
  - *Verification*: Must ensure that the control algorithm reaches and maintains the desired operating region
  - *Complexity*: A hierarchical control scheme is needed to manage large-scale distributed systems

SIGNAL PROCESSING



- Objective: identify relevant data from incoming signals
- Signals are received at a random rate
- Detection accuracy and computation time depends of the signal size
- Controller required to minimize the latency and maximize accuracy

System Dynamics:

$$\hat{q}_i(t+1) = q_i(t) + \left(\hat{\lambda}_i(t) c_i(r_i(t)) - v_i(t)\right)$$

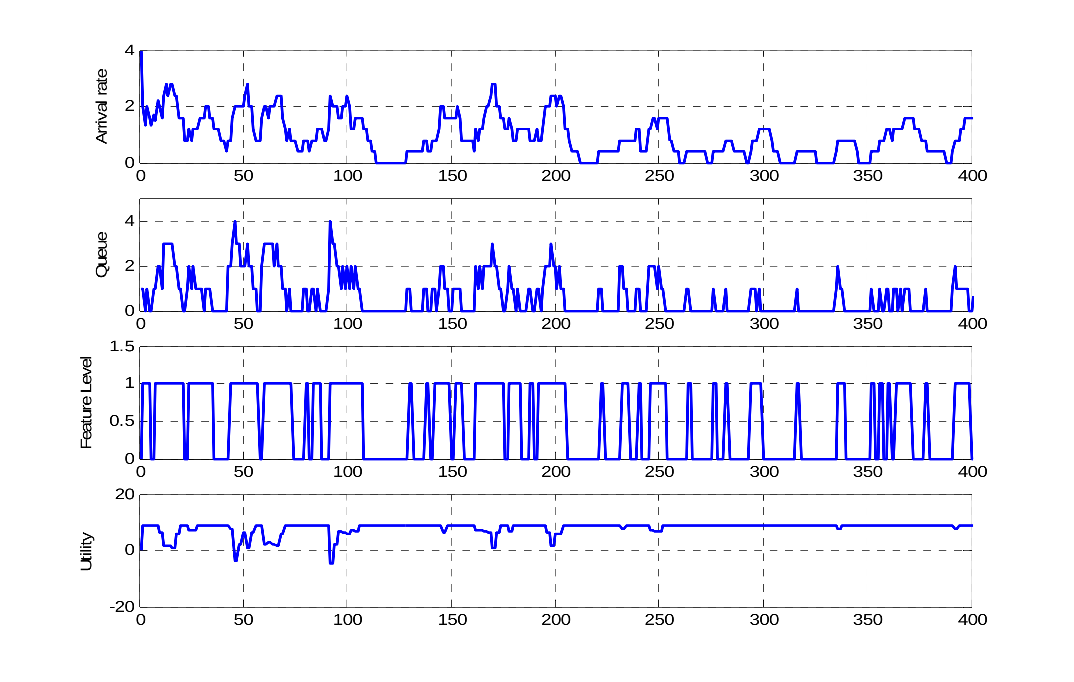
Estimated queue level (Latency)

$$\hat{a}_i(t+1) = f_i(a_i(t), r_i(t), v_i(t))$$

Estimated detection Accuracy

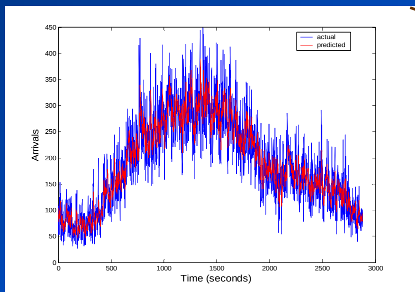
$$\max_t \left[w_1 \hat{a}(t) - w_2 \hat{q}(t) \right]$$

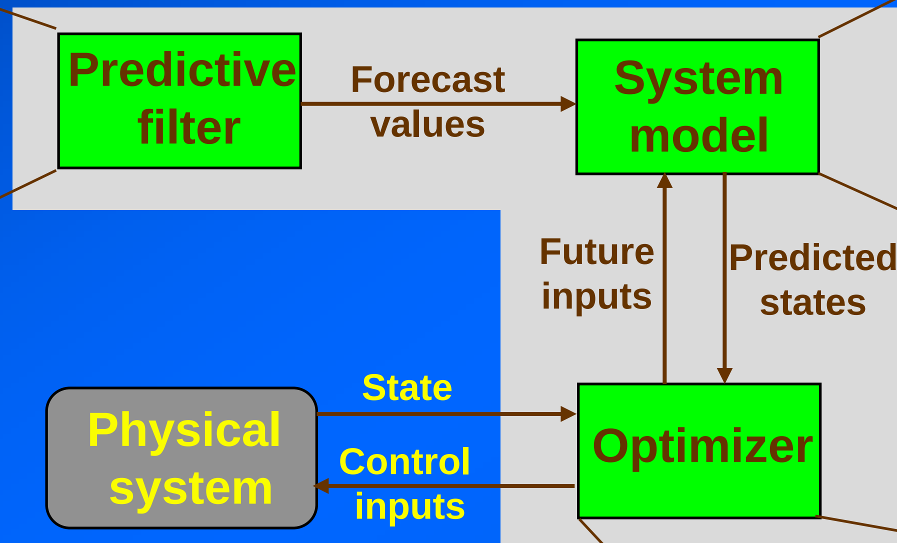
Performance Evaluation



- Prediction horizon = 3 steps
- Time step = 1 hour
- Local controllers optimize performance for a given mode

CASE STUDY: POWER MANAGEMENT





$$\hat{q}(t+1) = q(t) + (\hat{\lambda}(t) - \frac{\alpha(t+1)}{\hat{c}(t)}) \cdot T_s$$

$$\hat{\omega}(t+1) = (1+\hat{q}(t+1)) \cdot \hat{c}(t) / \alpha(t+1)$$

$$\hat{E}(t+1) = \alpha^2(t+1)$$

$$\hat{J}(t+1) = w_1 \cdot \hat{G}(t+1) + w_2 \cdot \hat{E}(t+1)$$

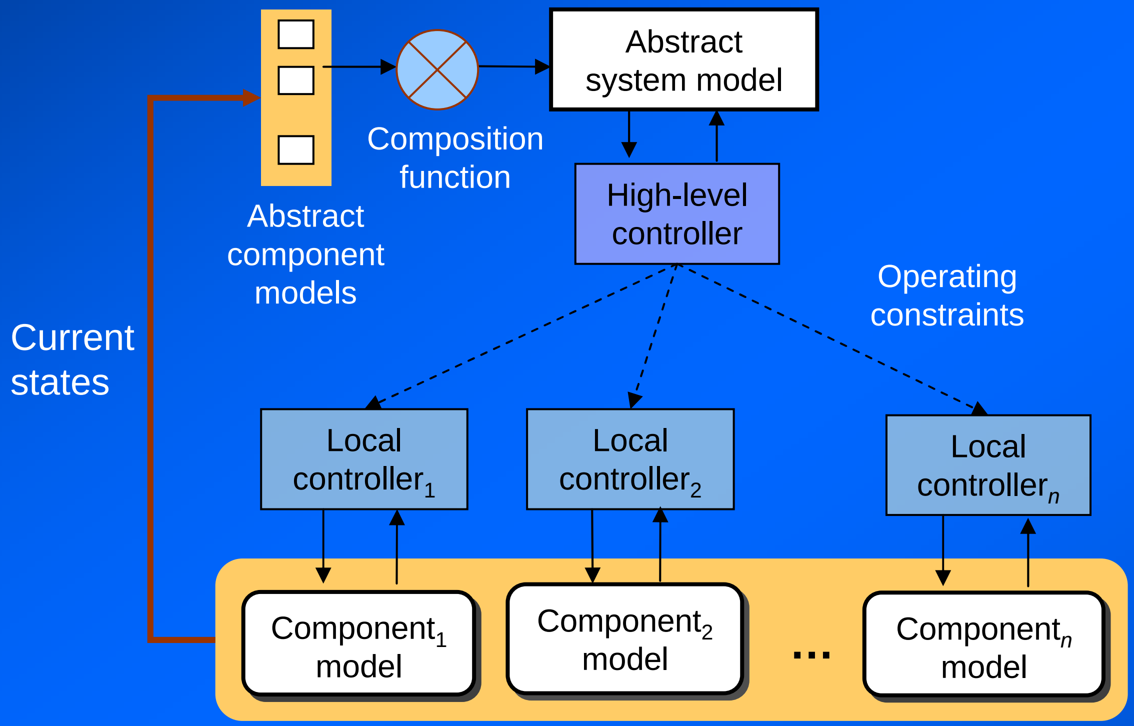
$$\hat{G}(t+1) = 0 \ \ if \ \ \hat{\omega}(t+1) - \omega_{ref} \le 0$$

$$\hat{G}(t+1) = \frac{\hat{\omega}(t+1) - \omega_{ref}}{\omega_{ref}} \ \ if \ \ \hat{\omega}(t+1) - \omega_{ref} > 0$$

- Assume a processor capable of operating at multiple frequencies; e.g., AMD-K-2
- Achieve a specified (average) response time ω_{ref} for requests
- Minimize processor operating frequency, and therefore energy consumption

FUTURE WORK

Distributed (hierarchical) online control



- Global controller manages inter-component interactions and enforces global QoS requirements
- Abstract representation of the components is used for high-level control decisions
- Global control actions provide additional constraints for local controllers
- *Application*: Energy management in large-scale server clusters