Self-Awareness at the Level of Heterogeneous Compute Nodes

Marco Platzner
platzner@upb.de

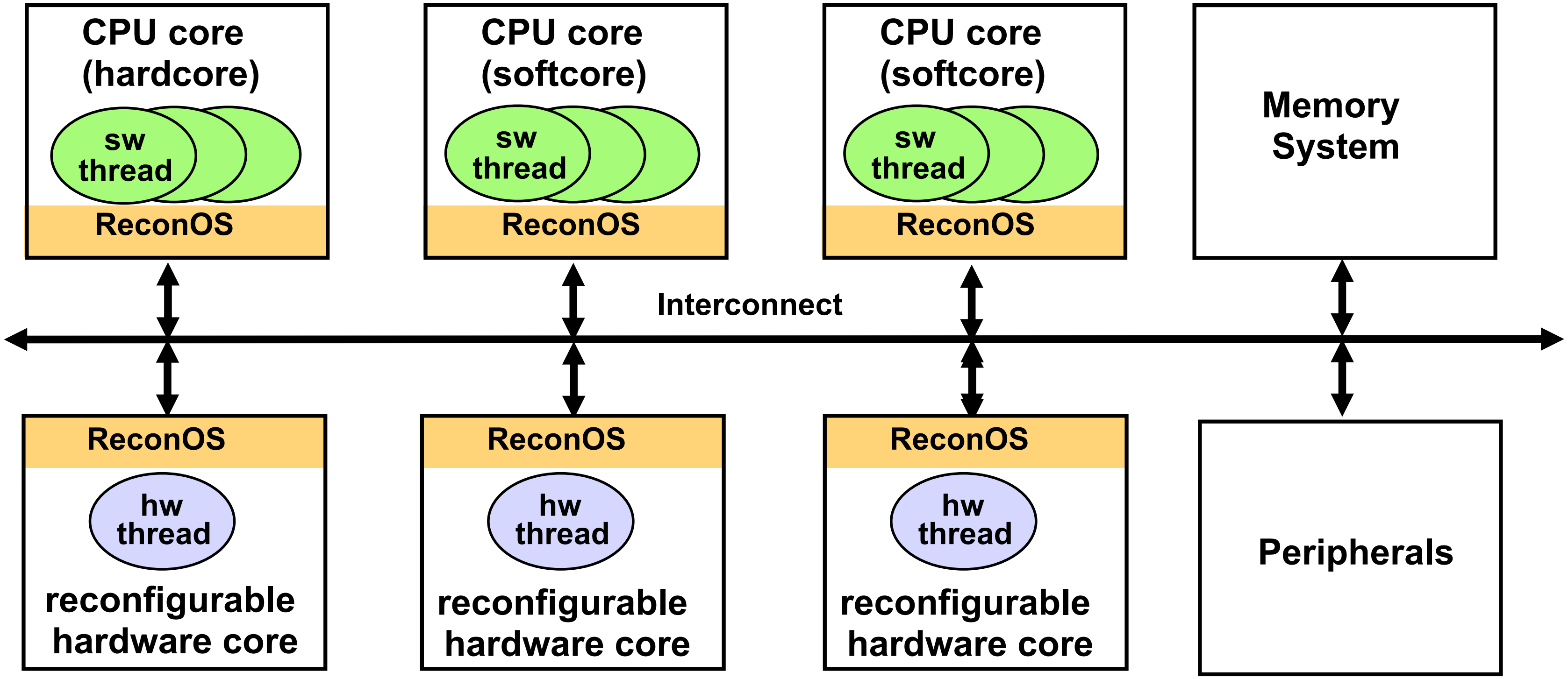
PADERBORN UNIVERSITY
The University for the Information Society

Outline

- Heterogeneous compute nodes
  - architecture and programming model
  - building self-adaptive systems
- The EPiCS project
  - reference architecture and design patters
  - case studies with heterogeneous compute nodes
- Planned work
  - high-performance compute nodes/clusters
  - micro aerial vehicles
- Conclusion

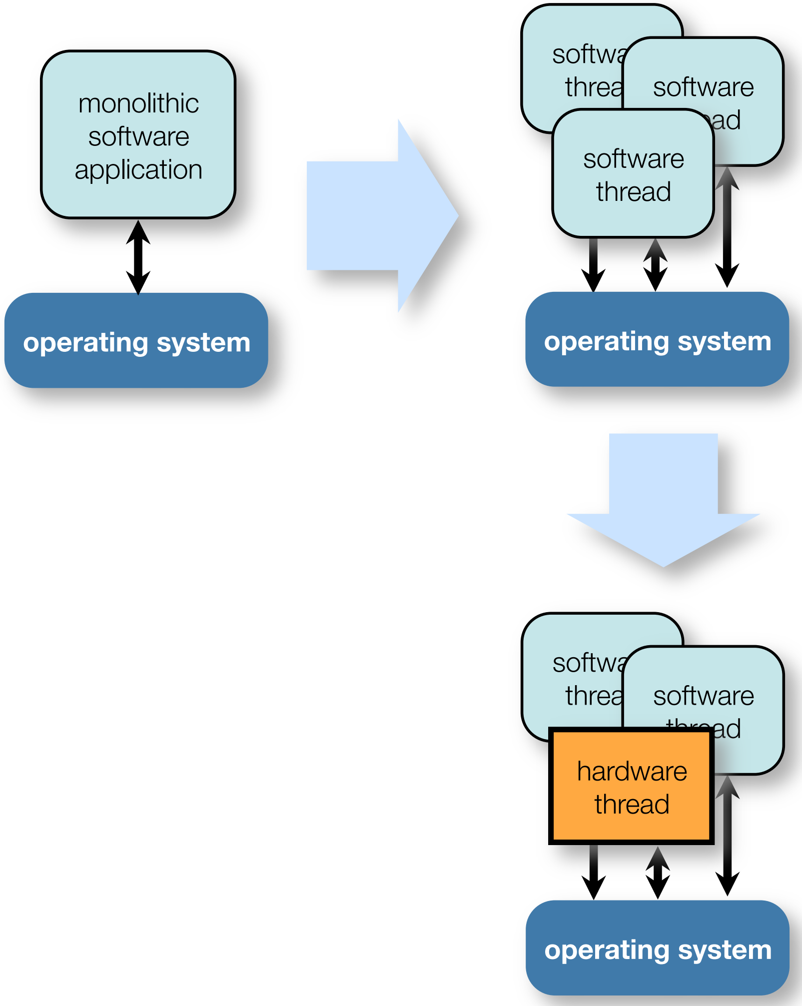
Heterogeneous Multi-Cores

- Combine CPUs with reconfigurable hardware cores



ReconOS – Programming Model

- ReconOS extends the multithreaded programming model to FPGAs



[A. Agne et al. *ReconOS - An Operating System Approach for Reconfigurable Computing*. IEEE Micro, 2014.]
[E. Lübbers & M. Platzner. *ReconOS: Multithreaded Programming for Reconfigurable Computers*. ACM TECS, 2009.]

Case Study: Object Tracking in Videos

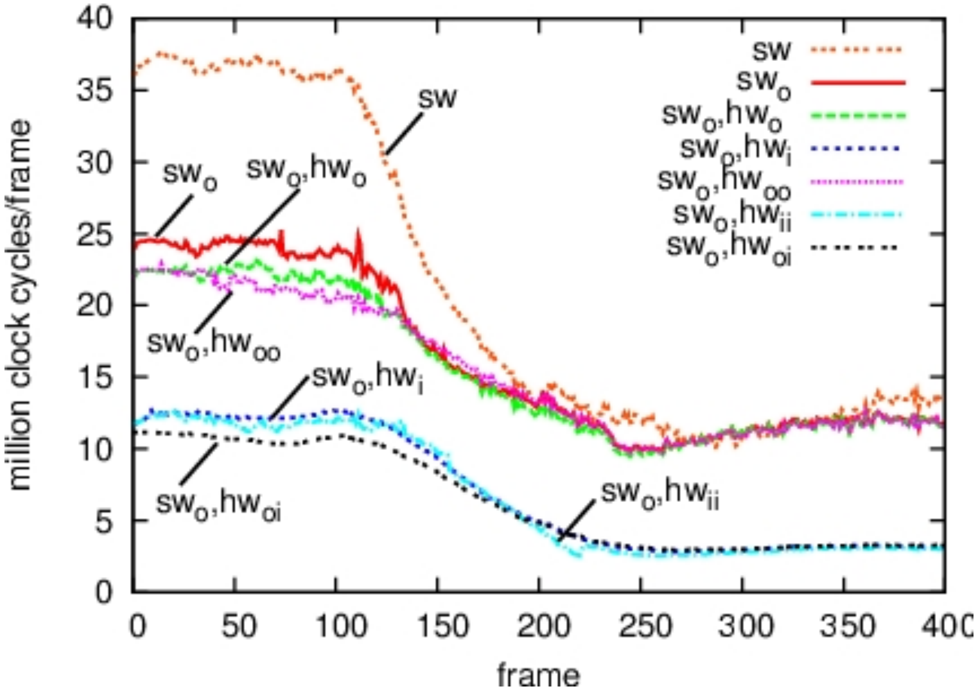
- Particle filter on ReconOS V2.0 / Virtex-4 (2 x PPC 405)
  - sw: all threads run in software
  - hw*: a number of threads run in hardware
  - sw*: a number of threads run on second (worker) CPU

estimated system state (particle)

α={*x*,*y*,*s*}
position: *x*, *y*
scaling factor: *s*

$$\begin{aligned} \alpha_t &= \alpha_{t-1} + v_{\alpha_{t-1}} + u_{\alpha_t} \\ v_{\alpha_t} &= \alpha_t - \alpha_{t-1} \\ u_{\alpha_t} &\sim \mathcal{N}(0, \sigma^2) \end{aligned}$$

frame 5 | frame 90 | frame 150 | frame 260



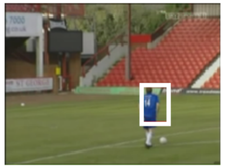
[M. Happe et al. *A Self-adaptive Heterogeneous Multi-core Architecture for Embedded Real-time Video Object Tracking*. International Journal of Real-Time Image Processing, 2011.]

Case Study: Object Tracking in Videos

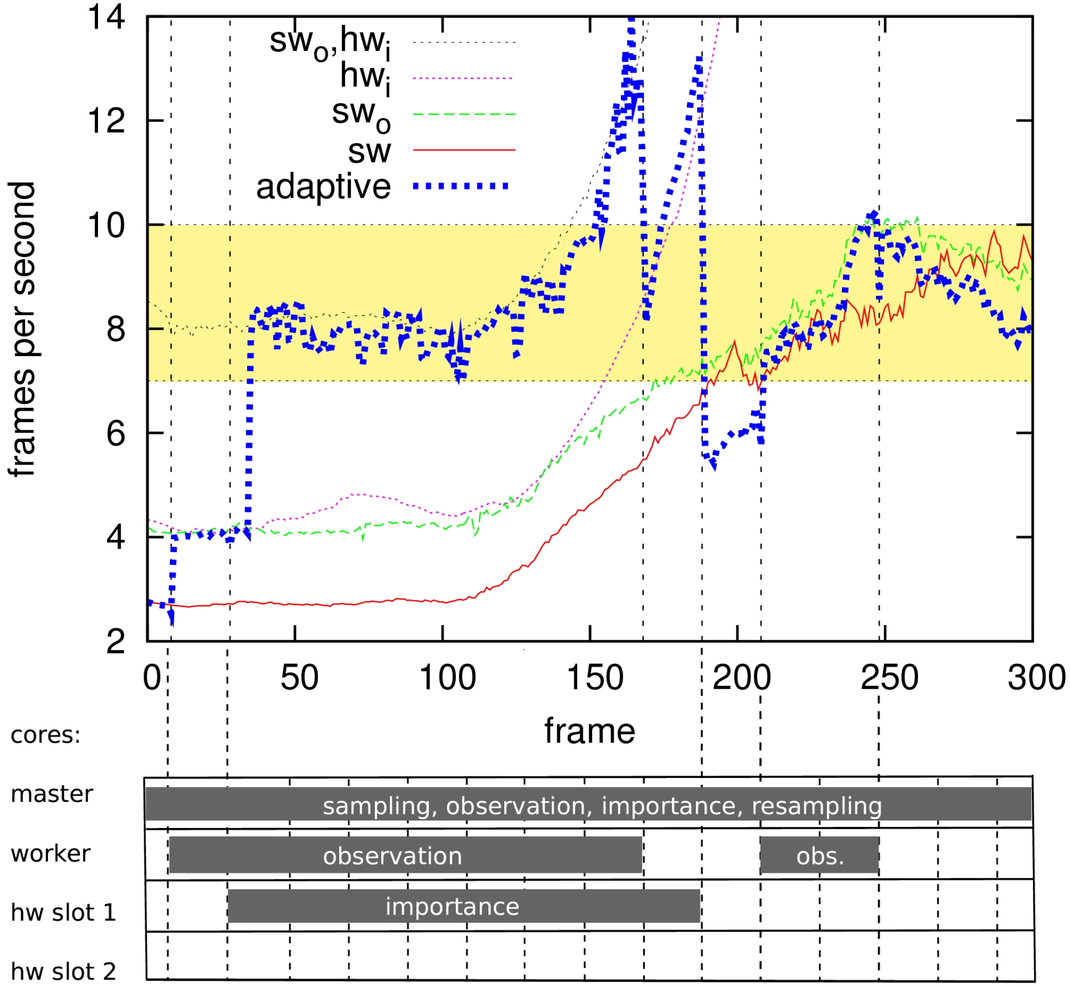
- Self-adaptive system operation
  - constraint: performance in [7,10] fps
  - objective: minimize number of cores


frame 5


frame 90


frame 150


frame 260

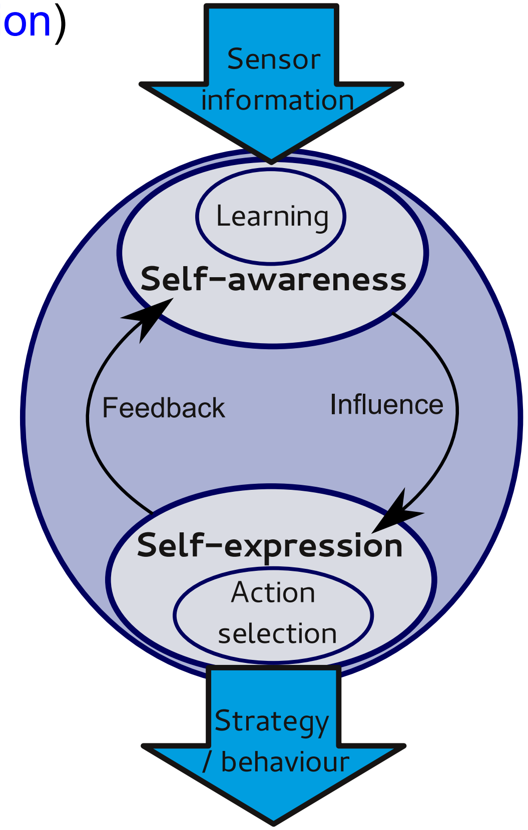


Outline

- Heterogeneous compute nodes
  - architecture and programming model
  - building self-adaptive systems
- The EPiCS project
  - reference architecture and design patters
  - case studies with heterogeneous compute nodes
- Planned work
  - high-performance compute nodes/clusters
  - micro aerial vehicles
- Conclusion

EU FET Proactive "EPiCS" (2010-2014)

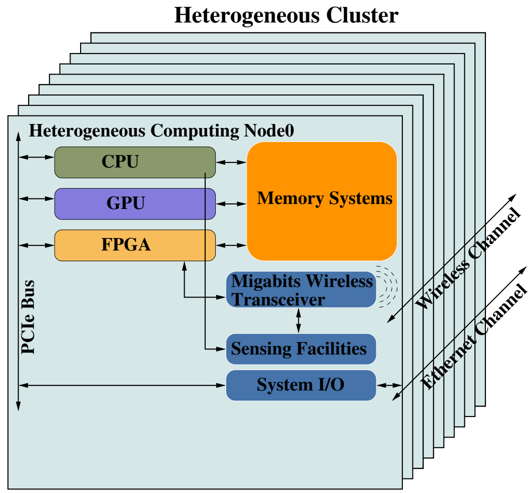
- Engineering Proprioception in Computing Systems
  - use proprioceptive sensors to monitor "one-self"
  - reason about their behavior (self-awareness)
  - adapt behavior to changing conditions (self-expression)

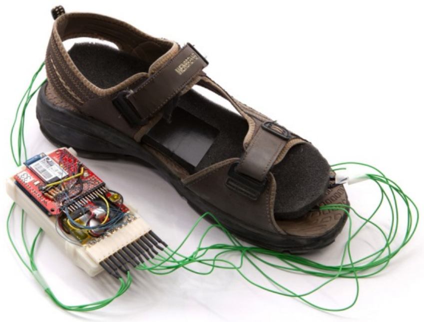


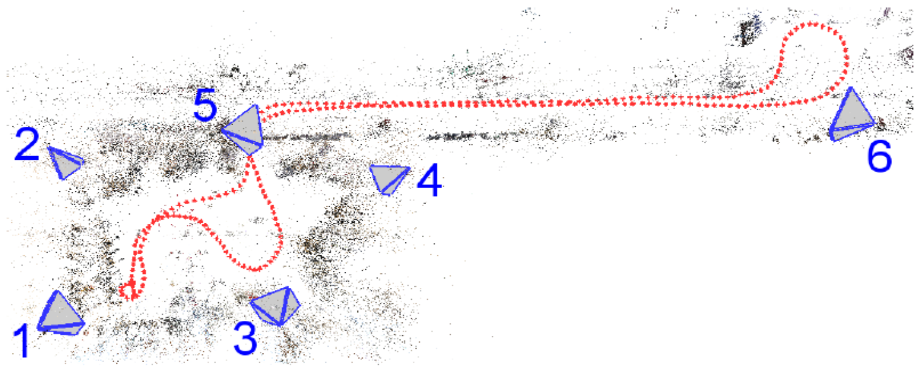
- Partners
  1. Paderborn University
  2. Imperial College London
  3. University of Oslo
  4. Klagenfurt University
  5. University of Birmingham
  6. EADS Innovation Works Munich
  7. Swiss Federal Institute of Technology Zurich
  8. Austrian Institute of Technology GmbH Vienna

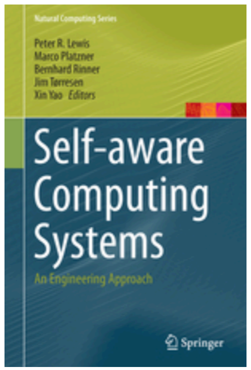
EU FET Proactive “EPiCS” (2010-2014)

- Application domains
  - heterogeneous compute clusters for financial modeling
  - distributed smart cameras
  - active music systems



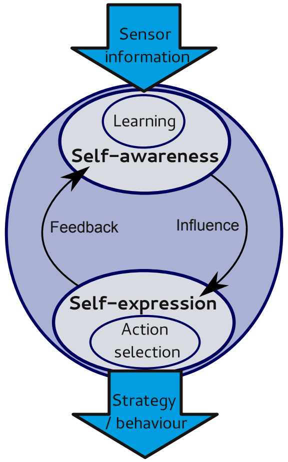




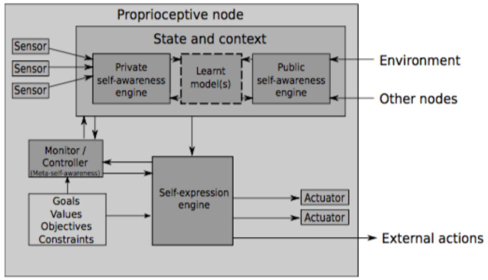


[P. R. Lewis, M. Platzner, B. Rinner, J. Tørresen, X. Yao (eds.) *Self-aware Computing Systems: An Engineering Approach*. Springer, 2016.]

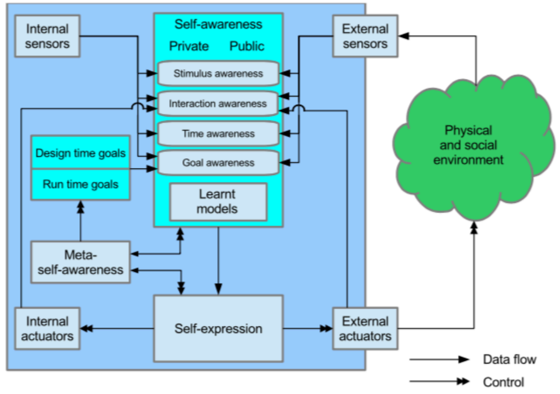
From Concept to Design Patterns



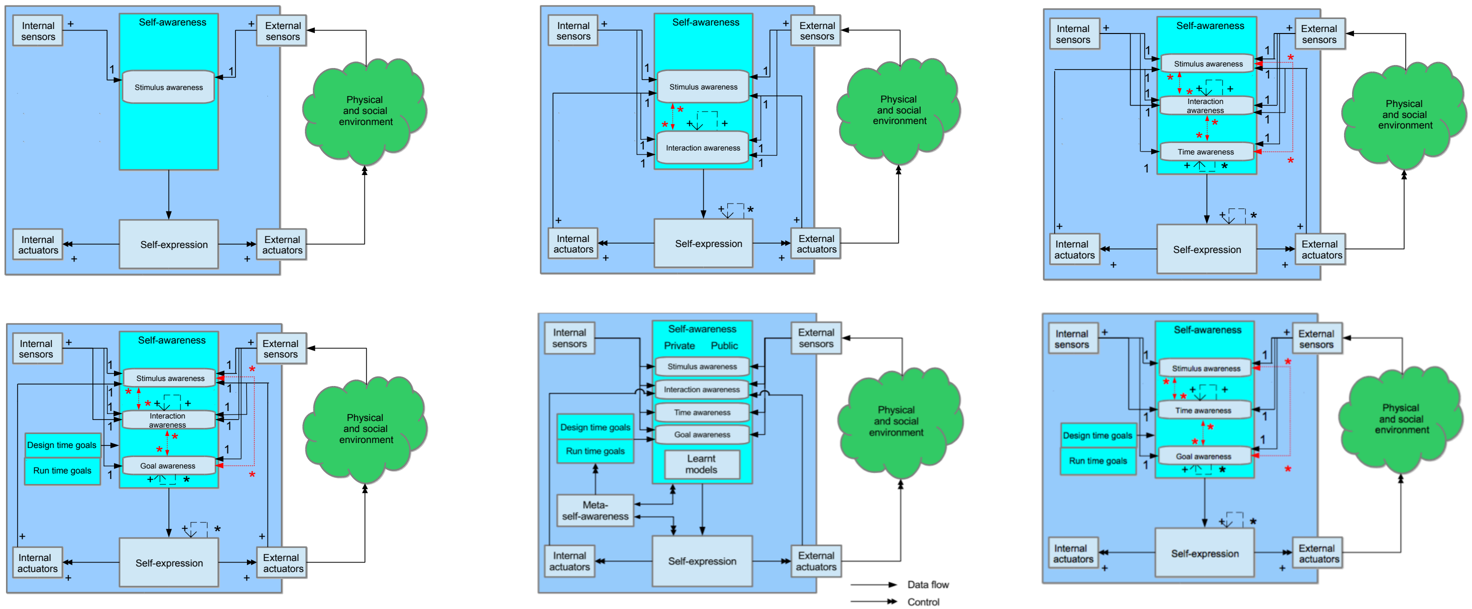
reference architecture



refined reference architecture /
different levels of
computational self-awareness

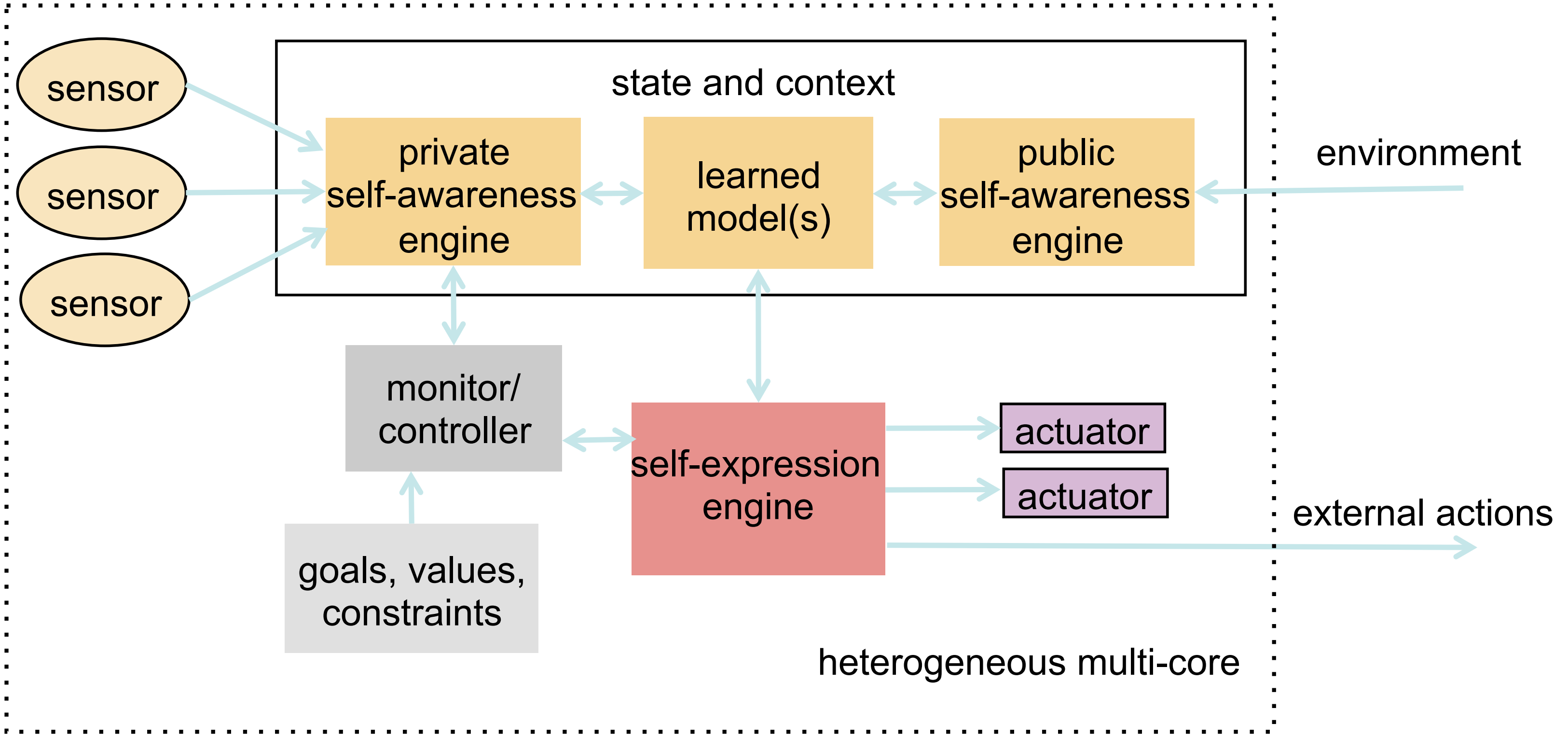


design patterns



Computational Self-awareness

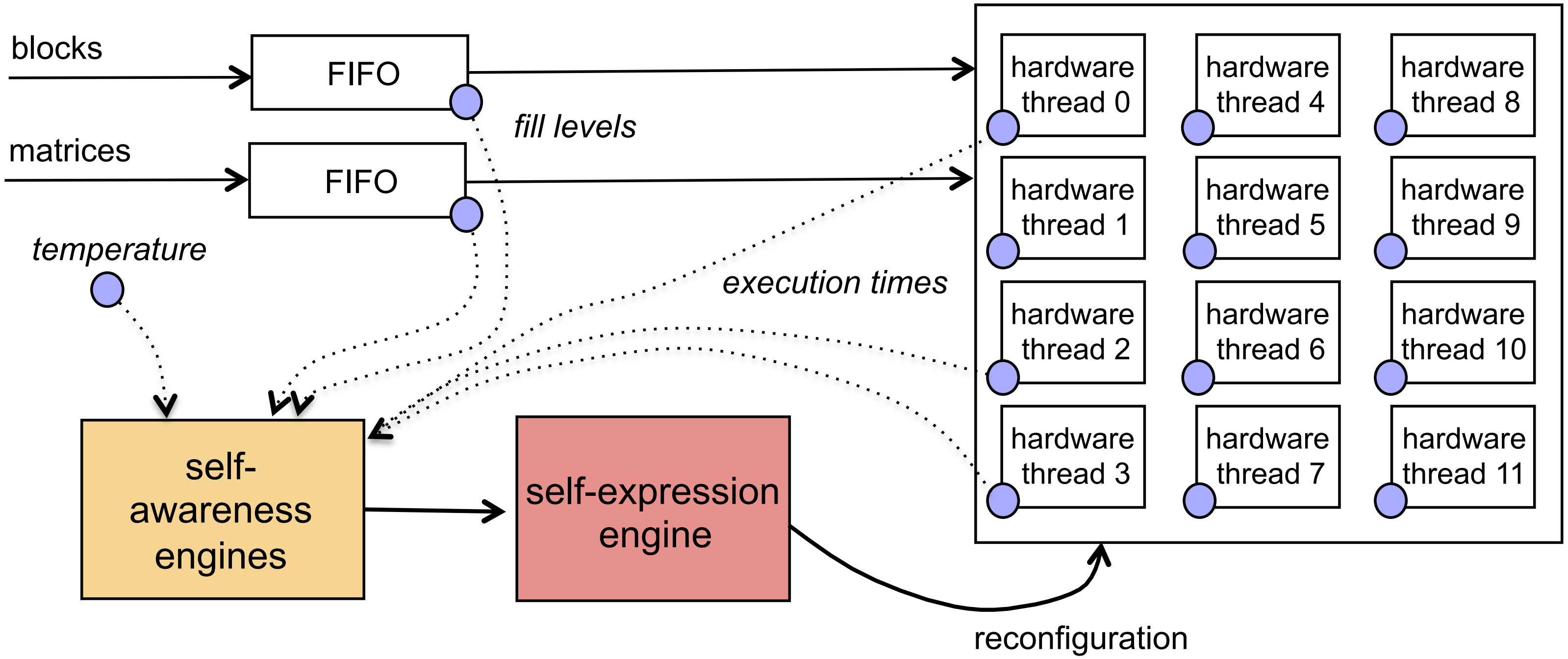
- Reference architecture adapted to heterogeneous compute nodes



[A. Agne et al. *Self-awareness as a Model for Designing and Operating Heterogeneous Multicores.* ACM TRETS, 2014.]

Case Study: 1+12 Multi-Core

- Workload
  - sort blocks of data, varying workload rate W_s
  - multiply matrices, infinite workload W_m
- Goal
  - objective: maximize the number of matrix multiplications
  - constraint: do not discard any block to be sorted



Case Study: 1+12 Multi-Core

- Why is this a challenge?
  - workload W_s varies over time, not known in advance
  - computation times vary with input data
  - computation and reconfiguration processes compete for memory bandwidth
  - reconfiguration processes compete for the reconfiguration interface (ICAP)

- Self-expression (SE): reconfigure more or less sorting threads

k	…	time step
$L_s(k)$	…	FIFO fill level at time step k
L_{max}	…	FIFO capacity
$N_s(k)$	…	number of active sorting threads at time step k
N_{max}	…	maximum number of hardware threads / reconfigurable slots

$$\forall k : L_s(k) \leq L_{max}$$

Case Study: 1+12 Multi-Core

- SE strategy "proportional" $N_s(k) := k_p.L_s(k)$ $k_p := N_{max}/L_{max}$

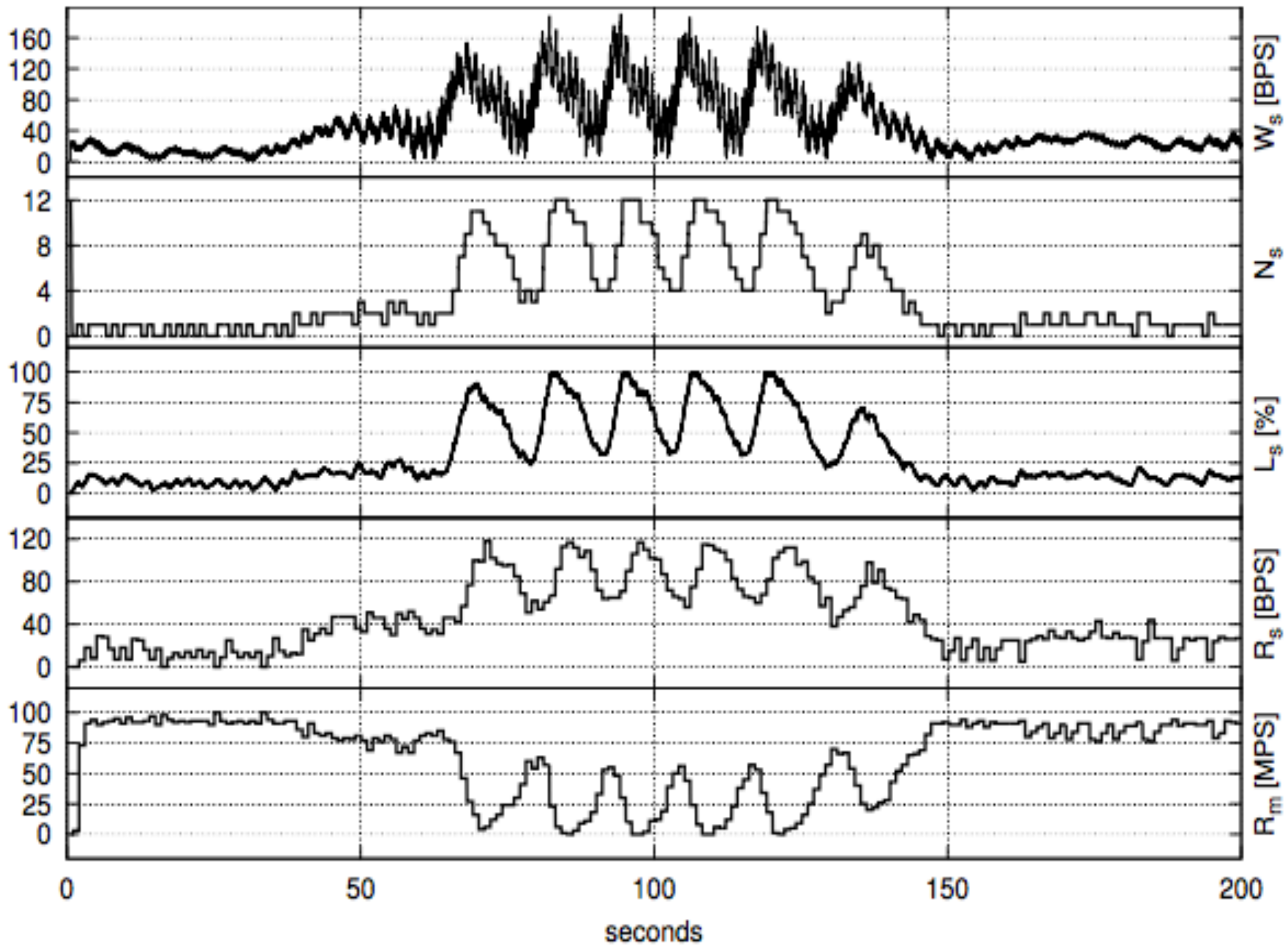
sorting workload

number of sorting threads

fill level [%]

sorting rate

matrix multiplication rate



Case Study: 1+12 Multi-Core

- SE strategy "all_or_nothing"

$$N_s(k) := \begin{cases} 0 & \text{if } L_s(k) = 0 \\ N_{\max} & \text{if } L_s(k) > L_{\text{trigger}} \quad \text{or} \quad L'_s(k) > \alpha \\ N_s(k-1) & \text{else} \end{cases}$$

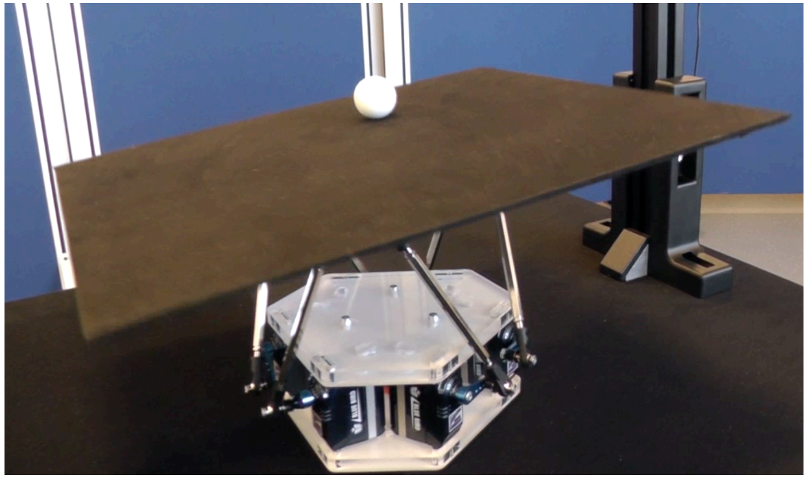
$$\alpha := \frac{L_{\max}}{\sum_{i=1}^{N_{\max}} D_{i-1}} \qquad L'_s(k) = (L_s(k) - L_s(k-1))/\Delta t$$

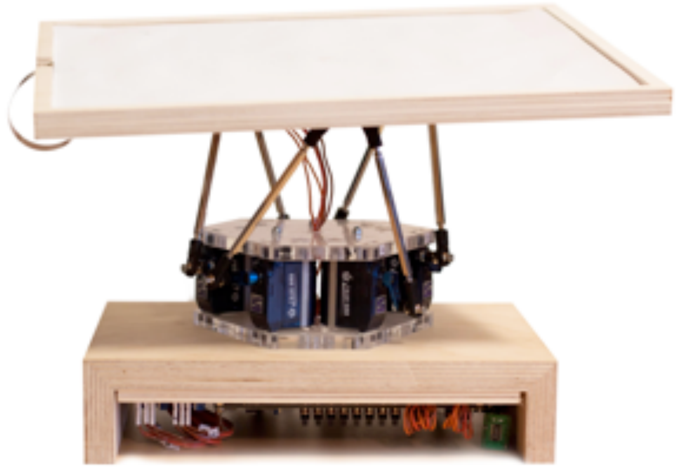
D_i … recent configuration delay for slot I

- SE strategy "meta"
  - switch between strategies "proportional" and "all_or_nothing"

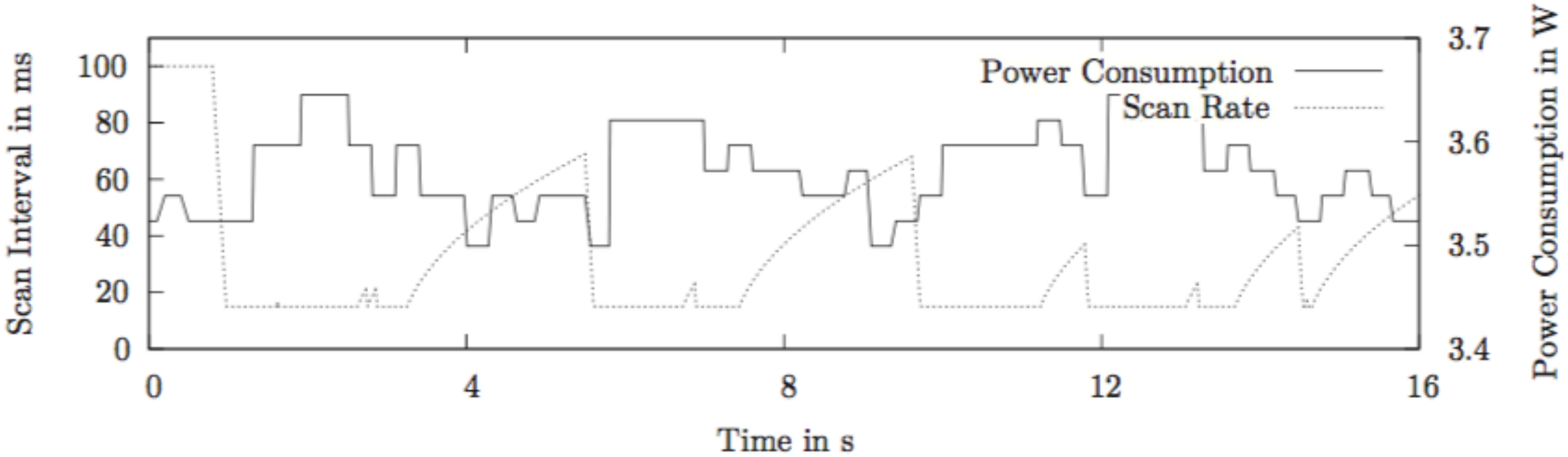
- SE strategy "thermally-aware meta"
  - stop matrix multiplication threads if the temperature exceeds Θ_{low}
  - stop also sorting threads if the temperature exceeds Θ_{high}
  - otherwise apply strategy "meta"

Case Study: Ball-on-Plate





Stewart platform balances ball

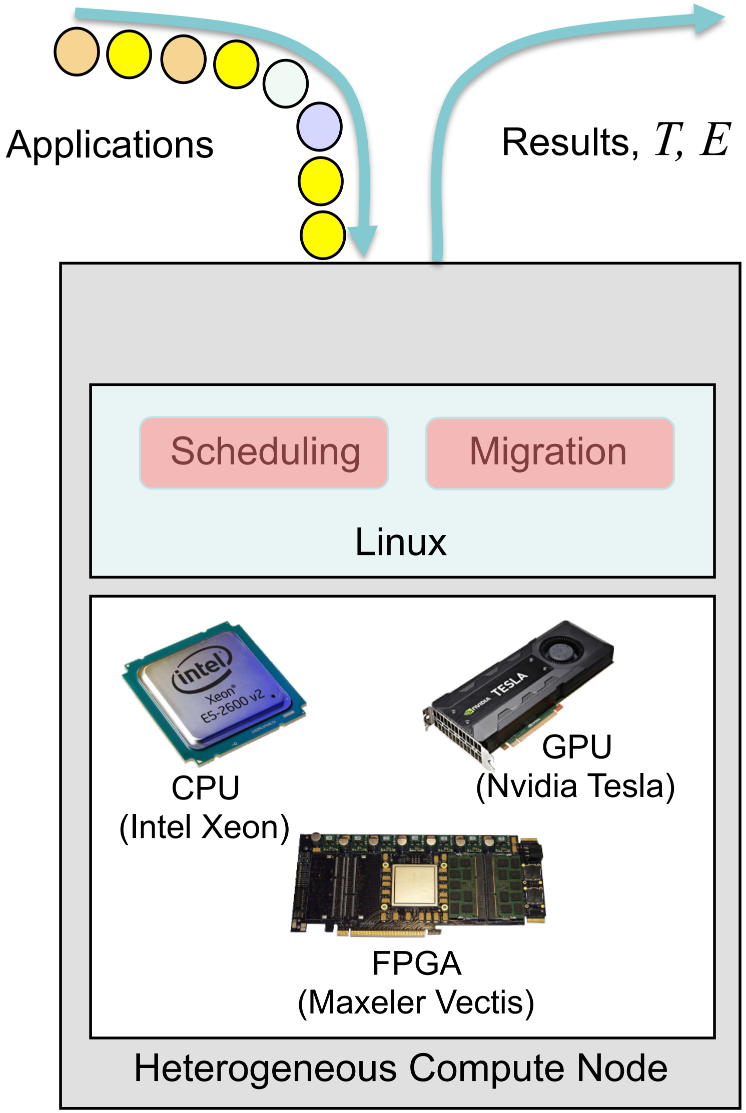


Adapt scan interval to speed of the ball to minimize power

Outline

- Heterogeneous compute nodes
  - architecture and programming model
  - building self-adaptive systems
- The EPiCS project
  - reference architecture and design patters
  - case studies with heterogeneous compute nodes
- Planned work
  - high-performance compute nodes/clusters
  - micro aerial vehicles
- Conclusion

Heterogeneous HPC Nodes

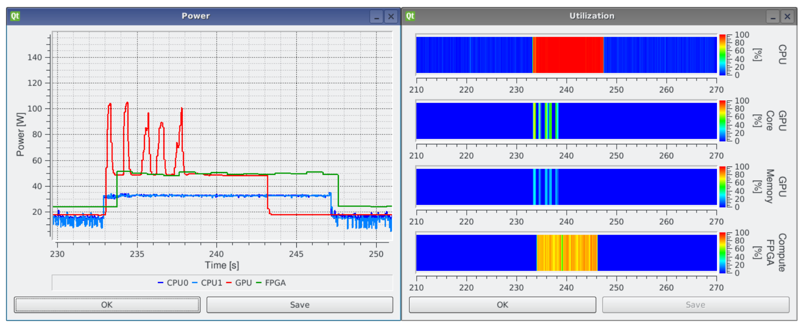


- Heterogeneous scheduling and migration

SFB 901
ON - THE - FLY COMPUTING

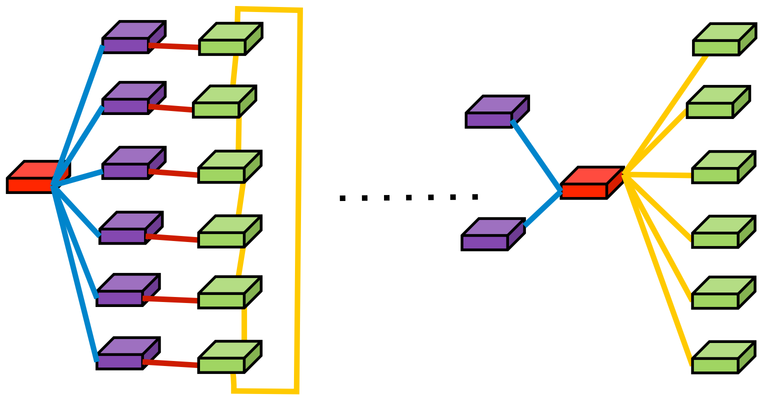
- Framework for accurately measuring runtime, energy, temperature, ...

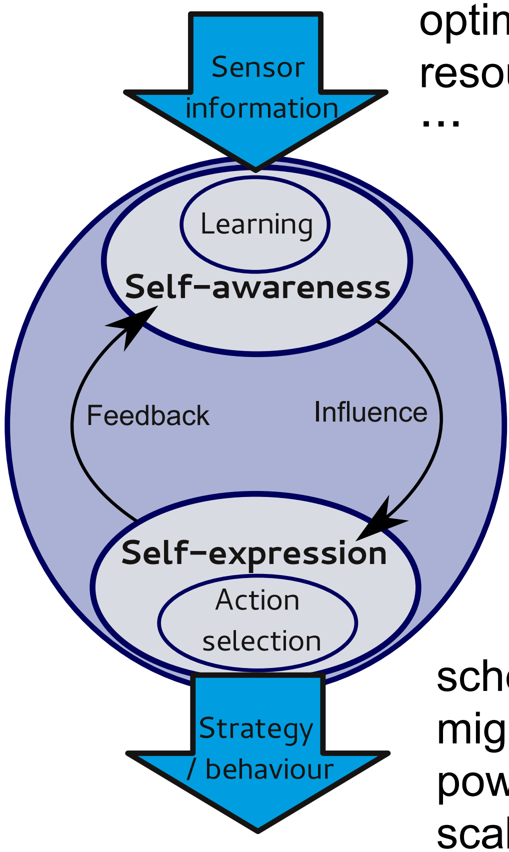
Ampehre



Cluster of Heterogeneous HPC Nodes

- Offline task, workload and resource characterization way too simplistic
  - varying request types
  - varying optimization goals
  - varying resource availabilities
  - reconfigurable interconnects (SDN)





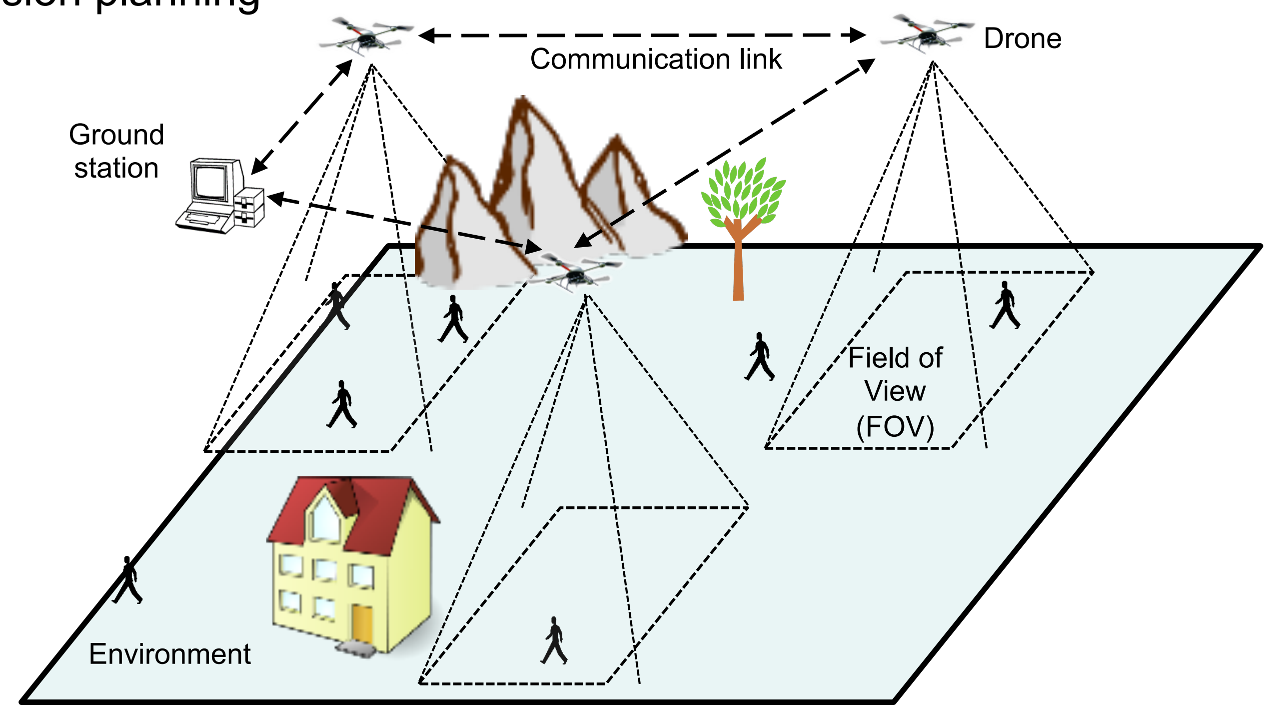
requests,
optimization goals,
resource state,
...

scheduling,
migration,
power management.
scaling out,
network reconfiguration,
...

(planned joint work with Christian Plessl, Paderborn University)

- Highly dynamic environment with strong resource limitations
- Tasks
  - positioning and navigation
  - motion control
  - communication and networking
  - coordination and mission planning
  - computer vision

(planned joint work with Bernhard Rinner, Klagenfurt University)



- My experience with computational self-awareness @ node level
  - highly useful to separate concerns and structure the overall system
  - knowledge about system & environment is concentrated in well-defined modules
  - simplifies setting up and experimenting with different adaptation ideas

- My experience with "defending" computational self-awareness

"quack science!"

source: www.netzwerk-lernen.de

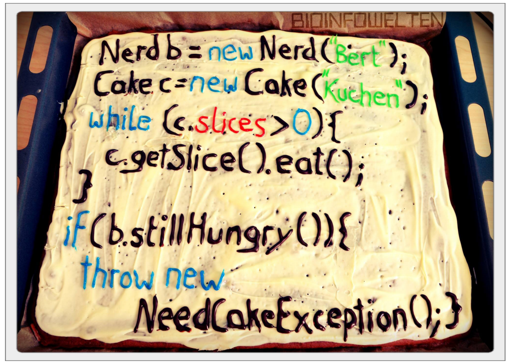
"killer app?"

source: www.colourbox.de

"self-awareness switch!"

source: kids-and-science.de

"algorithm?"



source: bioinfowelten.uni-jena.de