

Towards Energy Auto Tuning.

Sebastian Götz, Claas Wilke, Matthias Schmidt,
Sebastian Cech and Uwe Aßmann

Green IT, Singapore, 25.10.2010



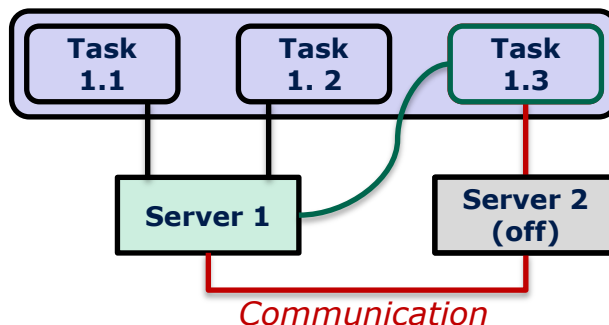
www.cool-software.org



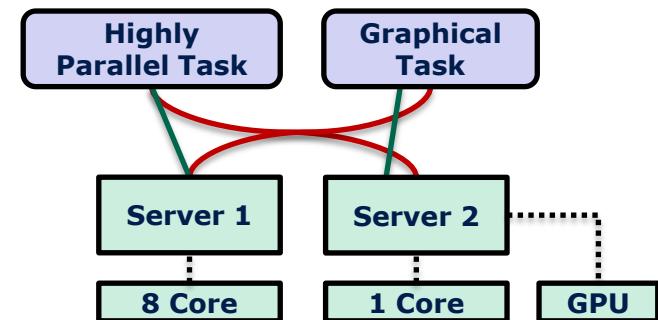
Bundesministerium
für Bildung
und Forschung

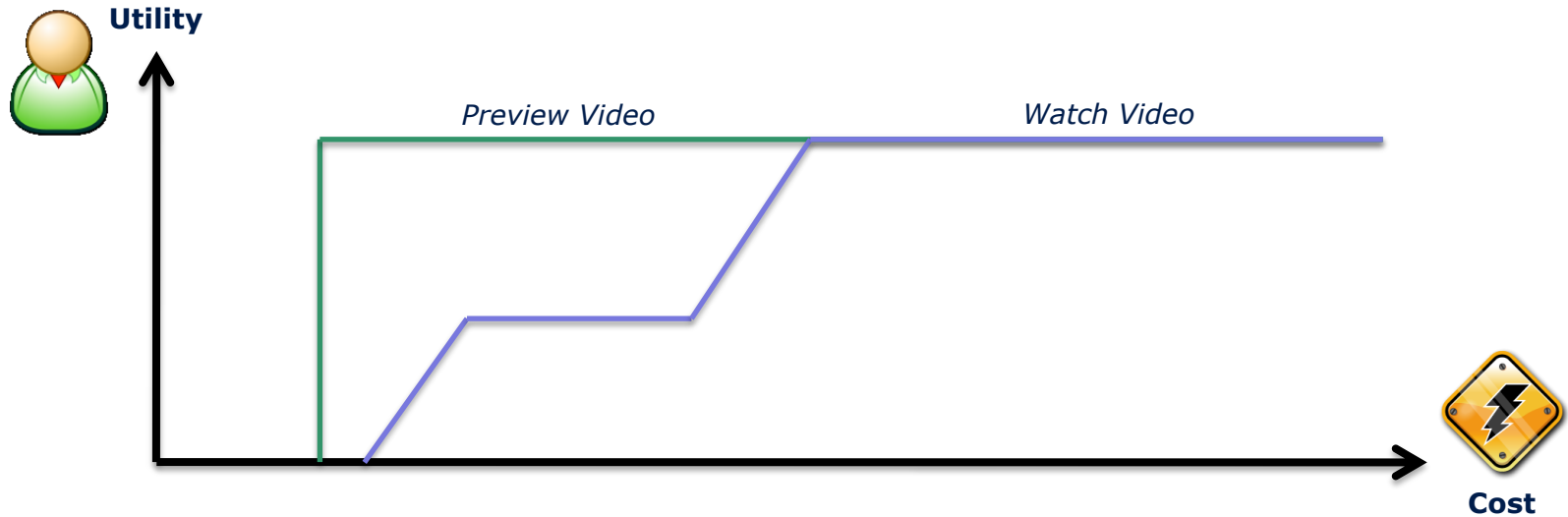
- Energy use of servers is steadily increasing [EPA07]
- Energy use of single resources thoroughly optimized
 - At first hardware optimizations
 - But also software optimizations
- Global optimization of multiple, shared resources will enable significant improvements

Consolidation



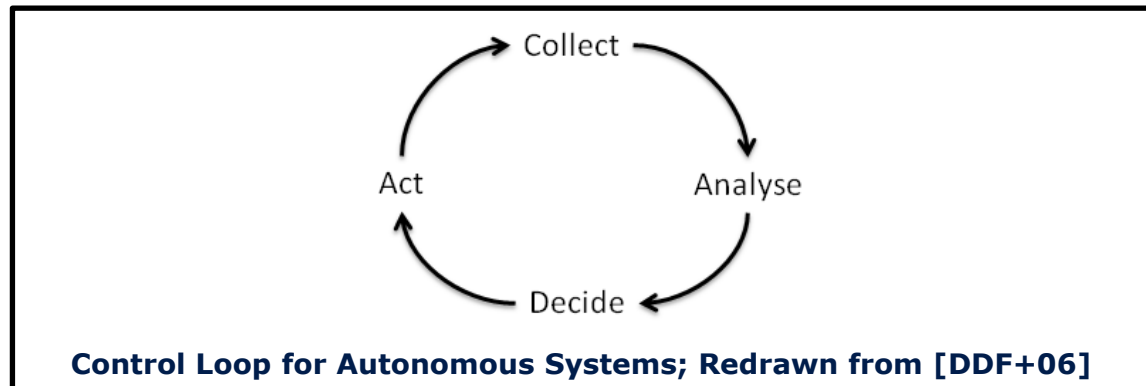
Selection





- User intention can change
 - Implied energy consumption, too (e.g. changed utilization)
- System needs to adjust itself steadily
→ **Energy Auto Tuning**

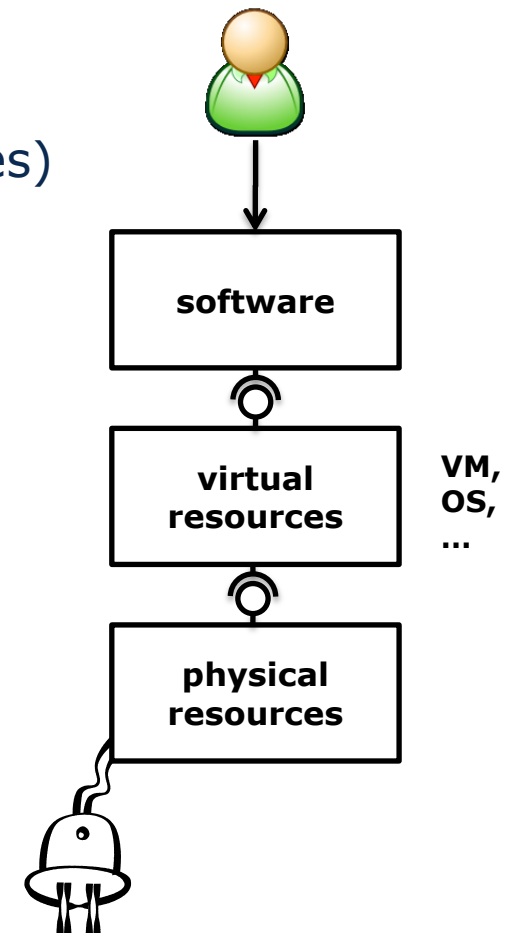
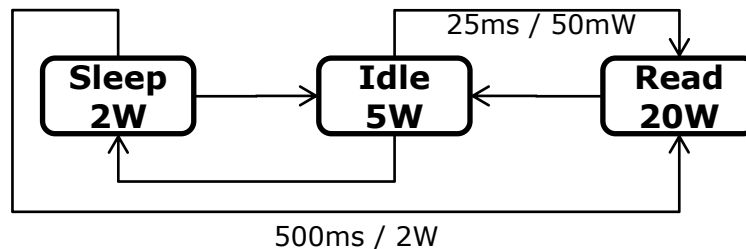
- Other terms: self-optimization, control loop, self-adaptive system
- Term *auto tuning* from *High Performance Computing* (HPC)
- System adjusts itself to the current situation



- Focus in HPC on *algorithms*
- ⇒ Transfer to **architectures** (*components and connectors*)
- ⇒ *Self-optimizing energy-adaptive system*

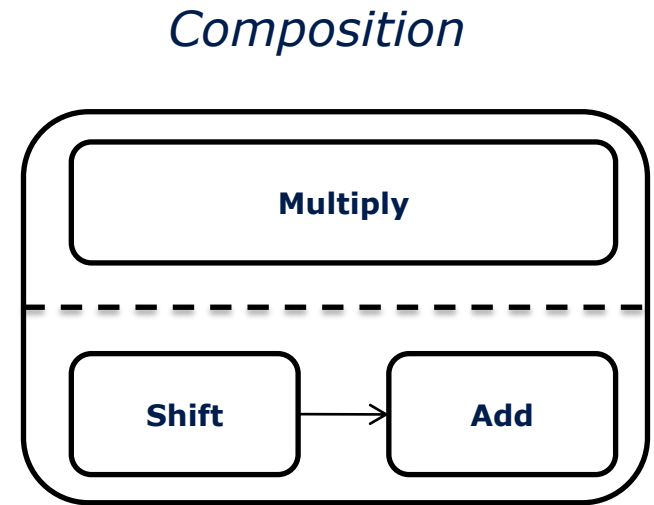
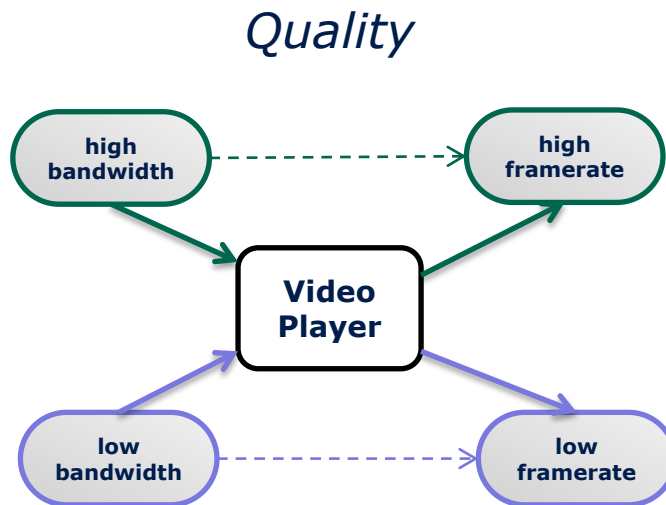
1. Hardware/Software Modeling

- Components (provided/required services)
- Variant modeling (for reconfiguration)
 - Quality-of-Service (QoS) properties
 - Alternative implementations
- Behavior modeling (energy states)



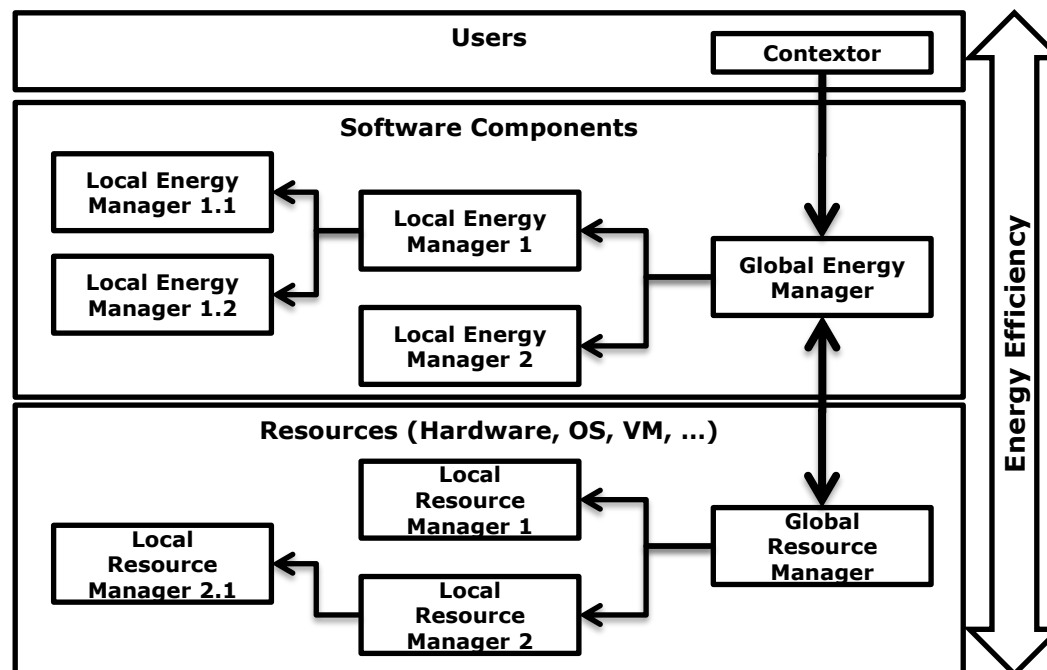
2. Dependency Modeling

- Between Components (SW-SW and SW-HW)
- Variability



3. Runtime Environment

- Monitoring (Collect + Analyze Phase)
- Reconfiguration (Decide + Act Phase)



Focus of this paper on:

- **Hardware/Software Modeling**
 - CoolComponentModel (CCM)
- **Dependency Modeling**
 - Energy Contract Language (ECL)

Outlined:

- **Runtime Environment**
 - Three Layer EAT Runtime Environment (THEATRE)

Cool Component Model

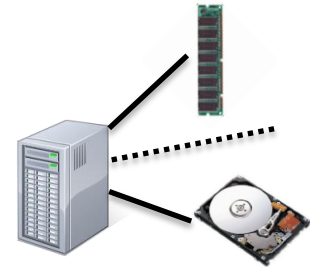
- **Structure**



User

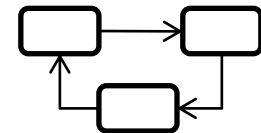


Software

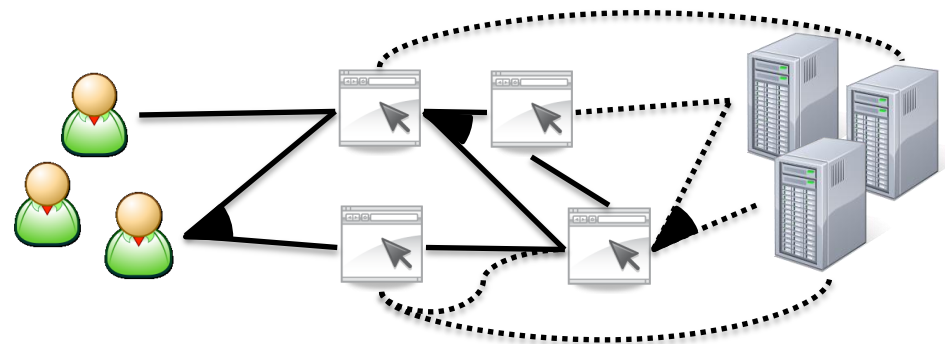


Hardware

- **Behavior**



- **Variants**



Energy Contract Language

```
characteristic imageWidth(width : Integer) {  
    domain Integer unit pixel  
    value width  
}
```

```
quality highResolution {  
    imageWidth(352)  
    imageHeight(288)  
} predefined characteristics
```

```
profile NetworkProfile for resource Network {  
    provides exactly dataRate(self::sendReceive::dataRate)  
}
```

Energy Contract Language

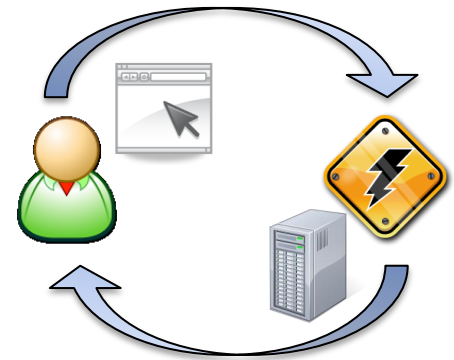
```
profile ServerProfile for component VideoServer {
  state highQuality {
    requires resource Network
    requires min dataRate(300.0)

    provides min frameRate(30)
    provides highResolution
  }

  state lowQuality {
    requires resource Network
    requires min dataRate(100.0)

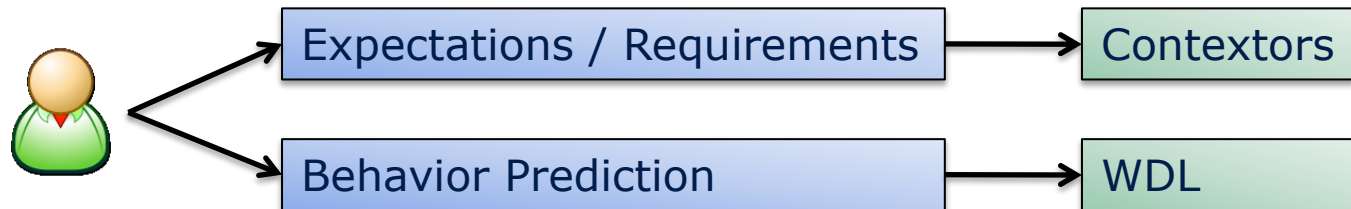
    provides min frameRate(10)
    provides lowResolution
  }
}
```

- Global optimization of multiple, shared resources will enable significant improvements
- User utility and energy use vary over time
 - ⇒ Energy Auto Tuning
 - ⇒ Self-optimizing energy-adaptive Systems
- **Contribution:**
 - Cool Component Model (HW/SW Modeling)
 - Energy Contract Language (Dependencies)

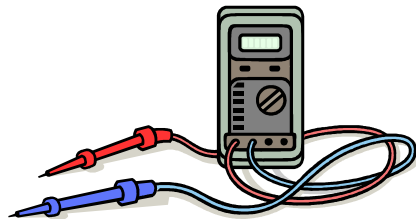


- Future Work

- Runtime environment (THEATRE)
- User Layer



- Evaluation by measurements





Thank you for listening!

Questions?



sebastian.goetz@acm.org

- [EPA07]** *U.S. Environmental Protection Agency, ENERGY STAR Program, “Report to congress on server and data center energy efficiency public law 109-431”, 2007.*
- [DDF+06]** *S. Dobson, S. Denazis, A. Fernández, D. Gaiti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, “A survey of autonomic communications”, ACM Transactions on Autonomous Adaptive Systems, vol. 1, no. 2, pp. 223–259, 2006.*

Cool Component Model

