

Towards Autonomic and Organic Computing Platforms

Wolfgang Karl

Institut für Rechnerentwurf und Fehlertoleranz

Universität Karlsruhe (TH)

OCI Workshop, Hannover

1.-2. Dezember 2003



Motivation

□ Future systems architectures:

- ◆ Complex in composition and behaviour
- ◆ Composed of an increasingly diverse set of interacting components
- ◆ Non-technical personnel:
 - ◆ Must be easy to use and fault-tolerant
- ◆ Practitioners:
 - ◆ Must enable to easily build scalable applications
- ◆ Must be self-x
 - ◆ Self-organizing, self-aware, self-optimizing, self-healing
- ◆ Require the existence of flexible, introspective data acquisition mechanisms



Motivation

❑ Current systems implement feedback loops:

◆ Drawback 1: Restricted monitoring capabilities

- ◆ CPUs offer some performance counters
- ◆ Enable limited data acquisition
- ◆ Sampling within program regions
- ◆ Do not enable more sophisticated queries:
 - ◆ Exact address information, queue length for network adapters, or size of predictor tables

◆ Drawback 2: Single source monitoring

- ◆ Permits limited view of current system status
- ◆ Prohibits system-wide adaption and optimization



Motivation

- ❑ Flexible and unifying framework for non-intrusive hardware monitoring of virtually any component
 - ◆ System-wide, unified introspection as a building block for autonomic or organic systems
 - ◆ Data preprocessing on the actual monitoring probes (by leveraging reconfigurability)
 - ◆ Standardized access to performance information using high-level queries
 - ◆ Integration of hardware and system software probes in a common framework
 - ◆ Correlation of monitoring information from several sources to assemble a global system state for required metrics
 - ◆ Autonomic Optimization of system components
 - ◆ Ability for optimizing performance, power savings, heat dissipation, security, and reliability within a single framework

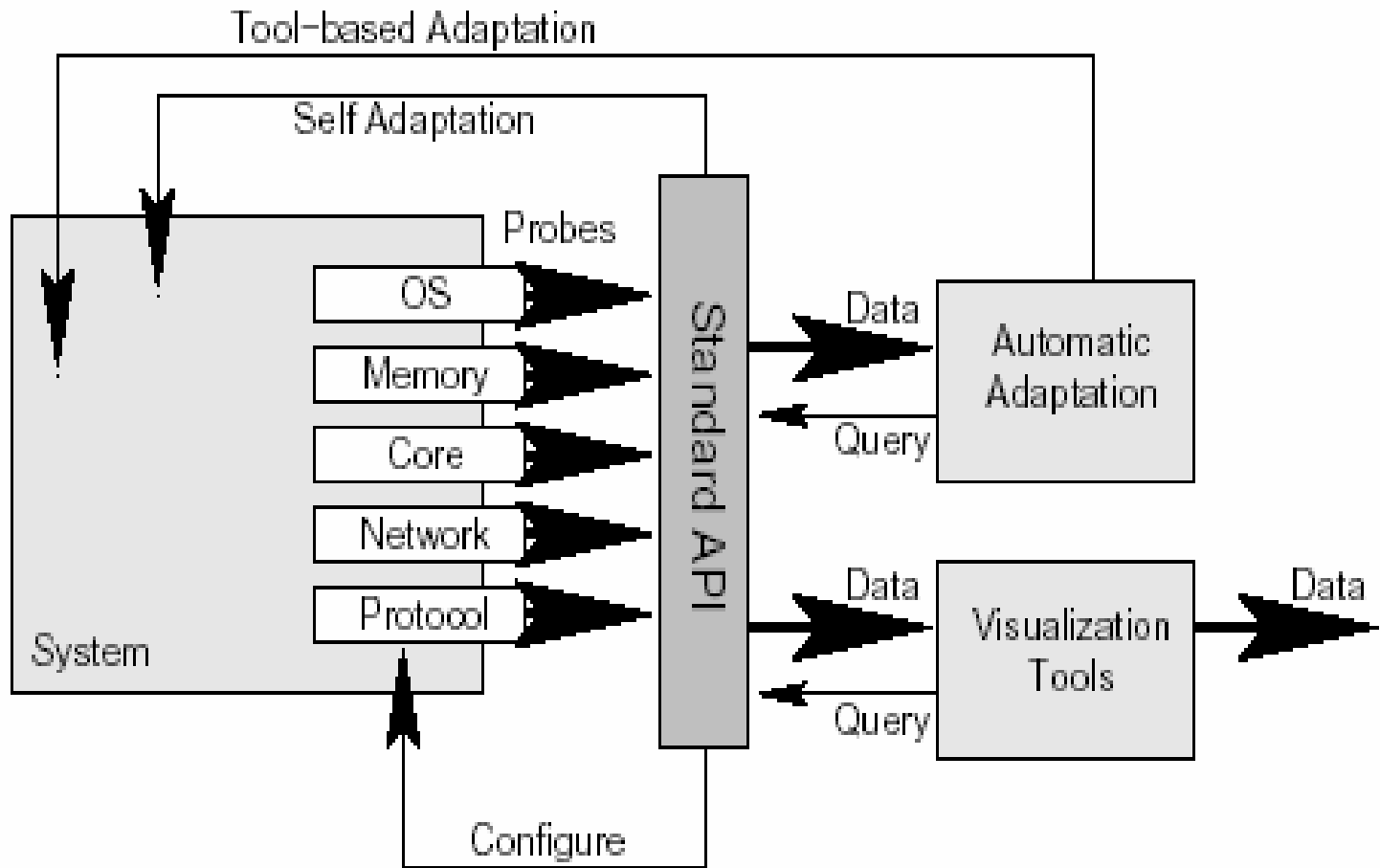


Approach

- ❑ **Generic and standardized monitoring capsule**
 - ◆ Capable of holding an adaptable, reconfigurable monitoring device
 - ◆ Domain-specific analysis modules
 - ◆ Can be loaded into these capsules to efficiently process monitoring data close to its source
 - ◆ Standardized API to control monitoring capsules
 - ◆ Query the processed data uniformly across all capsules in the system



Monitoring Architecture



Generic monitoring capsule

- ❑ Monitors operate independently of the processor
- ❑ Each monitor
 - ◆ operate as an independent device that snoops on a bus or a communication wire or
 - ◆ monitor system resources in the OS or runtime system
- ❑ Capsule can be deployed in form of HW or SW capsules, throughout the system



Generic Monitoring Capsule

□ Monitoring Infrastructure

◆ Generic Monitoring Capsule

- ◆ Responsible for observing data and storing monitoring results

◆ Monitor modules

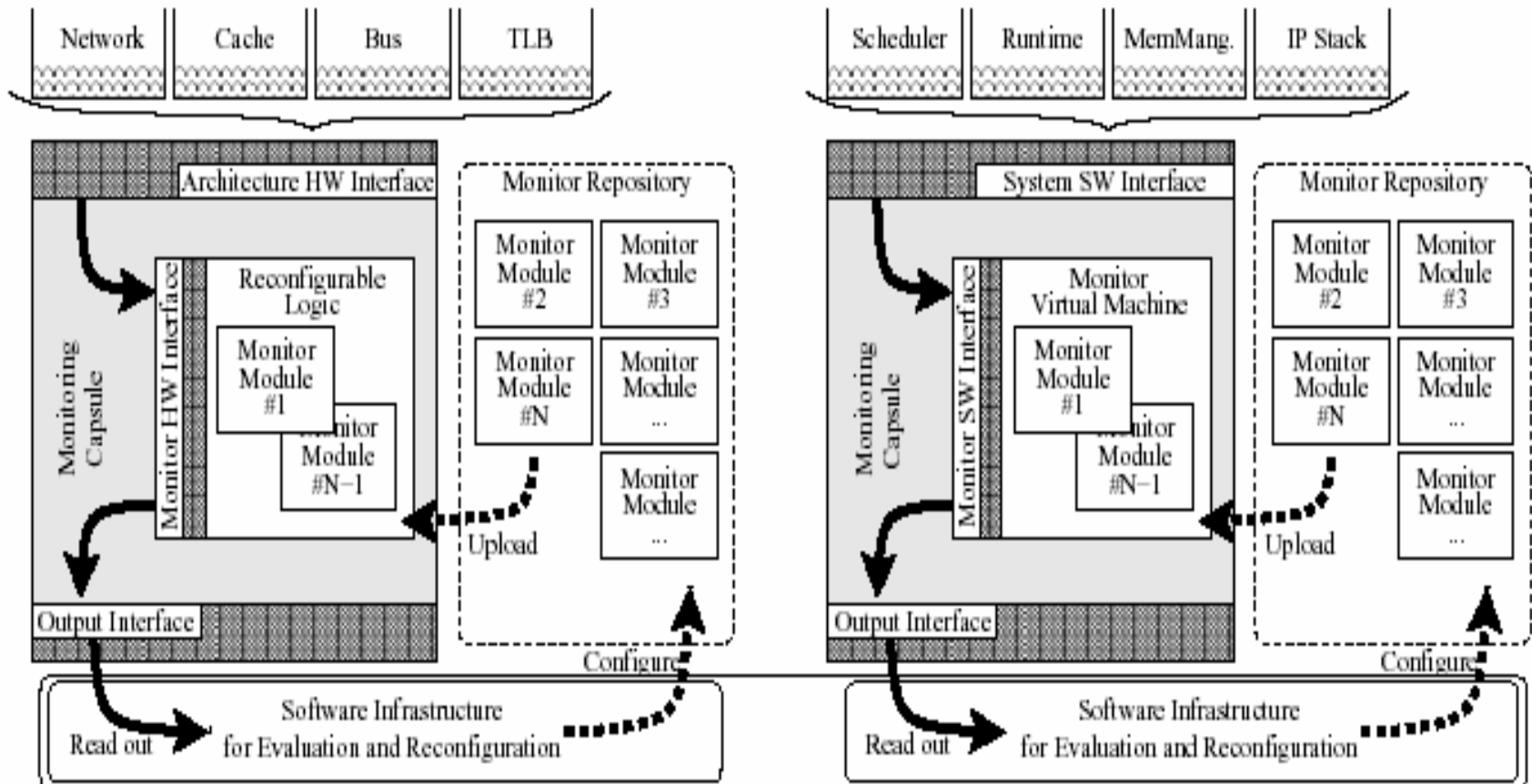
- ◆ Performing data analysis and filtering

◆ Interaction by a set of standard interfaces

□ Capsule can be deployed in form of HW or SW capsules, throughout the system

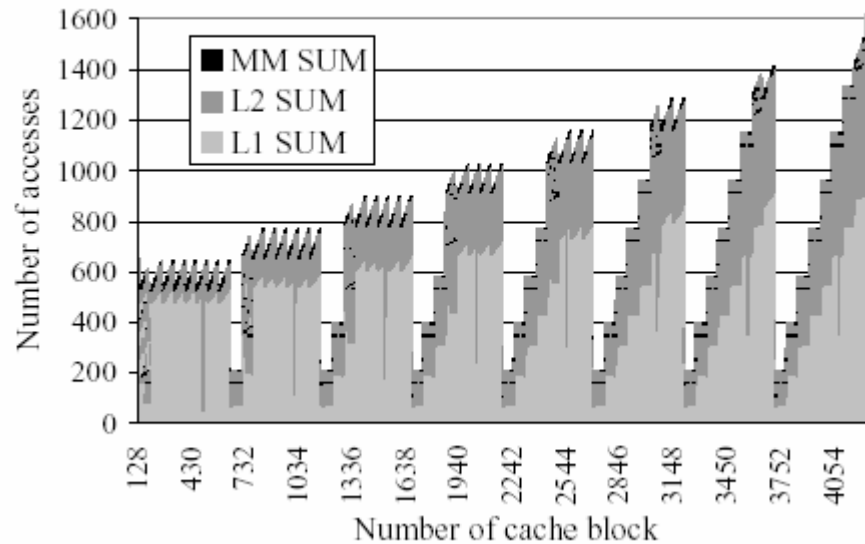
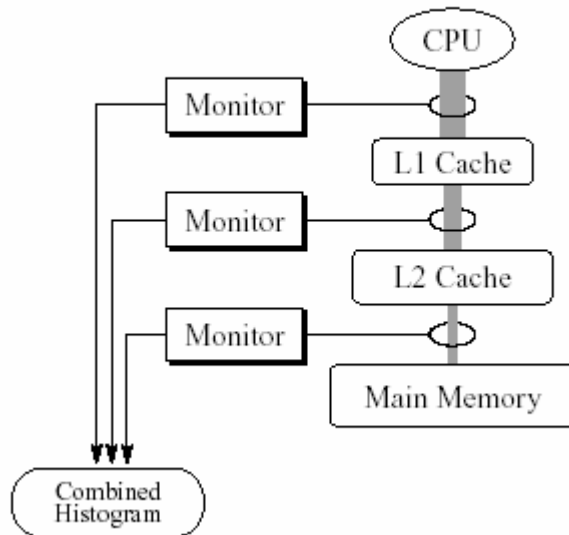


Architecture of the Monitoring Capsule



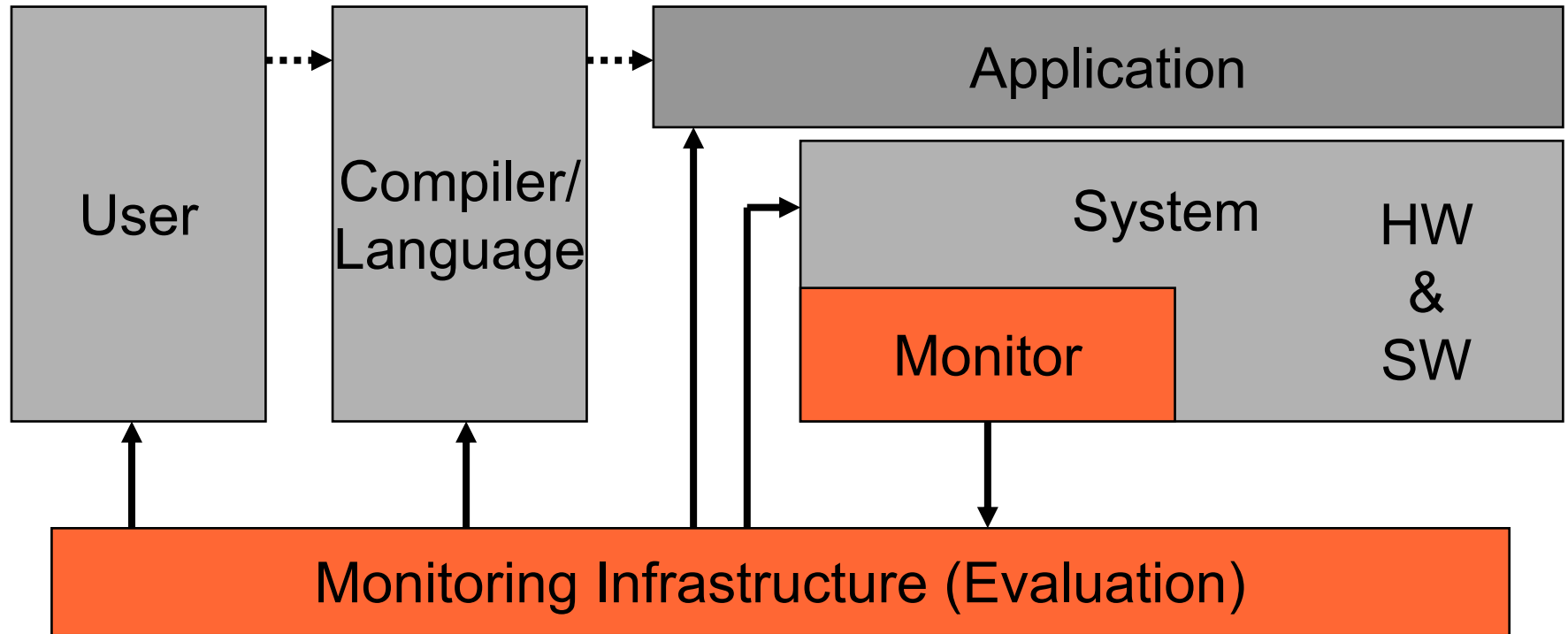
Monitoring Framework

- Reconfigurable system-wide monitoring framework
 - ◆ Multi-level monitoring
 - ◆ Memory system monitoring



Monitoring Framework

□ Autonomic Behaviour at system level



Monitoring Framework

□ Self-adaption

- ◆ A self-adapting module queries a performance monitor to optimize itself according self awareness of its performance or efficiency needs

□ Tool-based adaption

- ◆ For complex analysis tasks or for tasks requiring adaption itself
 - ◆ Tools or middleware services



Applications

- ❑ Processor core monitoring
- ❑ Reconfigurability
- ❑ Memory system monitoring
- ❑ Power adaption
- ❑ Adaptive security
- ❑ Network monitoring



Cooperation

□ Cornell University

- ◆ Fusion Group (Prof. S. McKee)

□ Technische Universität München

- ◆ Lehrstuhl für Rechnertechnik und Rechnerorganisation (Prof. Dr. A. Bode)

