Concurrent Systems

Nebenläufige Systeme

XIV. Pickings

Wolfgang Schröder-Preikschat

February 7, 2017

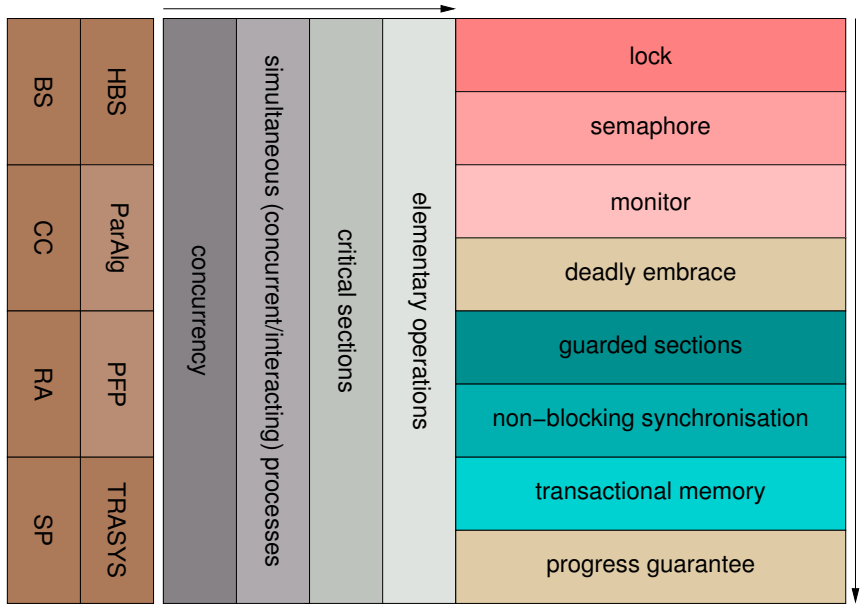
Agenda

Recapitulation
- Concurrent Systems

Perspectives
- Parallel Systems
- Computing Equipment
- Further Education

Outline

Content of Teaching and Cross-References



Outline

Main Research at the Chair

- **composability** and **configurability**
  - application-oriented (varying, type-safe) system software
- **specialisation**
  - dedicated operating systems: integrated, adaptive, parallel

Main Research at the Chair

- **composability** and **configurability**
  - application-oriented (varying, type-safe) system software
- **specialisation**
  - dedicated operating systems: integrated, adaptive, parallel
- **reliability**
  - gentle fault and intrusion tolerance
- **thriftiness**
  - ressource-aware operation of computing systems
- **timeliness**
  - migration paths between time- and event-triggered real-time systems

Main Research at the Chair

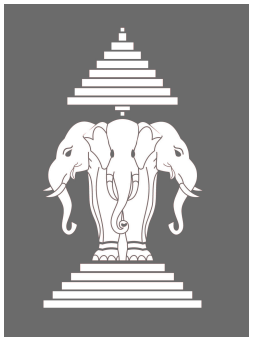
- **composability** and **configurability**
  - application-oriented (varying, type-safe) system software
- **specialisation**
  - dedicated operating systems: integrated, adaptive, parallel
- **reliability**
  - gentle fault and intrusion tolerance
- **thriftiness**
  - ressource-aware operation of computing systems
- **timeliness**
  - migration paths between time- and event-triggered real-time systems
- **concurrency**
  - coordination of cooperation and competition between processes

Main Research at the Chair

- **composability** and **configurability**
  - application-oriented (varying, type-safe) system software
- **specialisation**
  - dedicated operating systems: integrated, adaptive, parallel
- **reliability**
  - gentle fault and intrusion tolerance
- **thriftiness**
  - ressource-aware operation of computing systems
- **timeliness**
  - migration paths between time- and event-triggered real-time systems
- **concurrency**
  - coordination of cooperation and competition between processes

↪ "concurrent systems" is more or less **cross-cutting** thereto...

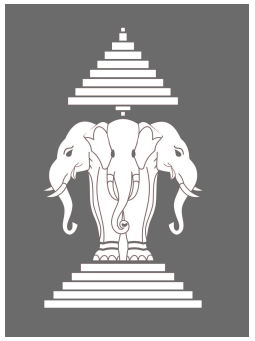
Latency Awareness in Operating Systems



[1] `http://univis.uni-erlangen.de` → Research projects → LAOS

Latency Awareness in Operating Systems

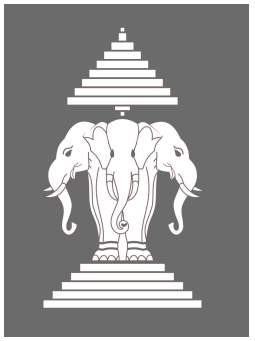
- **latency prevention**
  - lock- and wait-free synchronisation
  - integrated generator-based approach
- **latency avoidance**
  - interference protection
  - race-conflict containment
- **latency hiding**
  - operating-system server cores
  - asynchronous remote system operation



[1] `http://univis.uni-erlangen.de` → Research projects → LAOS

Latency Awareness in Operating Systems

- **latency prevention**
  - lock- and wait-free synchronisation
  - integrated generator-based approach
- **latency avoidance**
  - interference protection
  - race-conflict containment
- **latency hiding**
  - operating-system server cores
  - asynchronous remote system operation

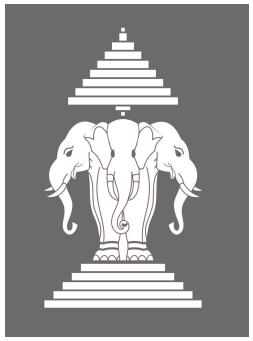


- experiments with different **operating-system architectures**
  - process-/event-based and hardware-centric operating-system kernels
  - LAKE, Sloth

[1] `http://univis.uni-erlangen.de` → Research projects → LAOS

Latency Awareness in Operating Systems

- **latency prevention**
  - lock- and wait-free synchronisation
  - integrated generator-based approach
- **latency avoidance**
  - interference protection
  - race-conflict containment
- **latency hiding**
  - operating-system server cores
  - asynchronous remote system operation
- experiments with different **operating-system architectures**
  - process-/event-based and hardware-centric operating-system kernels
  - LAKE, Sloth
- DFG: 2 doctoral researchers, 2 student assistants



[1] `http://univis.uni-erlangen.de` → Research projects → LAOS

Coherency Kernel

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

Coherency Kernel

- **event-based minimal kernel**
  - cache-aware main-memory footprint
  - hyper-threading of latent actions

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

Coherency Kernel

- **event-based minimal kernel**
  - cache-aware main-memory footprint
  - hyper-threading of latent actions
- featherweight **agreement protocols**
  - overall kernel-level synchronisation
  - familie of consistency kernels

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

Coherency Kernel

- **event-based minimal kernel**
  - cache-aware main-memory footprint
  - hyper-threading of latent actions
- featherweight **agreement protocols**
  - overall kernel-level synchronisation
  - familie of consistency kernels
- **problem-oriented consistency**
  - sequential, entry, release consistency
  - functional hierarchy of consistency domains
  - memory domains for NUMA architectures

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

Coherency Kernel

- **event-based minimal kernel**
  - cache-aware main-memory footprint
  - hyper-threading of latent actions
- featherweight **agreement protocols**
  - overall kernel-level synchronisation
  - familie of consistency kernels
- **problem-oriented consistency**
  - sequential, entry, release consistency
  - functional hierarchy of consistency domains
  - memory domains for NUMA architectures

- implementation as to different **processor architectures**
  - partial or total, resp. {in,}coherent shared memory

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 8

Coherency Kernel

- **event-based minimal kernel**
  - cache-aware main-memory footprint
  - hyper-threading of latent actions
- featherweight **agreement protocols**
  - overall kernel-level synchronisation
  - familie of consistency kernels
- **problem-oriented consistency**
  - sequential, entry, release consistency
  - functional hierarchy of consistency domains
  - memory domains for NUMA architectures

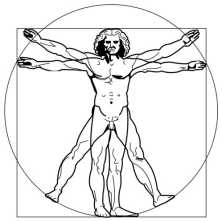
- implementation as to different **processor architectures**
  - partial or total, resp. {in,}coherent shared memory

- DFG: 2 doctoral researchers (1 FAU, 1 BTU)

[2] `http://univis.uni-erlangen.de` → Research projects → COKE

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 8

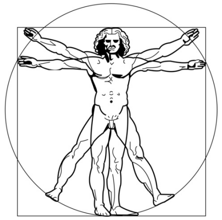
Heterogeneous Resource-Aware Multi-Processing



[3] `http://univis.uni-erlangen.de` → Research projects → RAMP

Heterogeneous Resource-Aware Multi-Processing

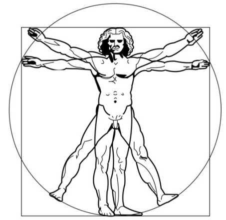
- GPU-centric **resource management**
  - timely predictable run-time system
    - run-to-completion kernel
  - priorisation and isolation of GPU tasks
    - scheduling according to execution costs
  - trade-off handling as to throughput and response time



[3]`http://univis.uni-erlangen.de` → Research projects → RAMP

Heterogeneous Resource-Aware Multi-Processing

- GPU-centric **resource management**
  - timely predictable run-time system
    - run-to-completion kernel
  - priorisation and isolation of GPU tasks
    - scheduling according to execution costs
  - trade-off handling as to throughput and response time



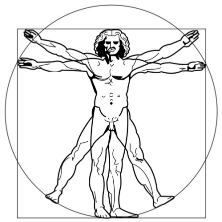
- RAM-centric **run-time executive** for heterogeneous processors
  - application-specific and problem-orientied memory management
  - run-time adaptation and relocation of dynamic data structures

[3]`http://univis.uni-erlangen.de` → Research projects → RAMP

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 9

Heterogeneous Resource-Aware Multi-Processing

- GPU-centric **resource management**
  - timely predictable run-time system
    - run-to-completion kernel
  - priorisation and isolation of GPU tasks
    - scheduling according to execution costs
  - trade-off handling as to throughput and response time



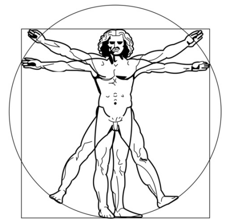
- RAM-centric **run-time executive** for heterogeneous processors
  - application-specific and problem-orientied memory management
  - run-time adaptation and relocation of dynamic data structures
- **tailor-made system software** for heterogeneous image systems
  - support of an incremental improvement of visual quality
  - patterns for adaptive detail adaptation of geometry or textures

[3]`http://univis.uni-erlangen.de` → Research projects → RAMP

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 9

Heterogeneous Resource-Aware Multi-Processing

- GPU-centric **resource management**
  - timely predictable run-time system
    - run-to-completion kernel
  - priorisation and isolation of GPU tasks
    - scheduling according to execution costs
  - trade-off handling as to throughput and response time



- RAM-centric **run-time executive** for heterogeneous processors
  - application-specific and problem-orientied memory management
  - run-time adaptation and relocation of dynamic data structures
- **tailor-made system software** for heterogeneous image systems
  - support of an incremental improvement of visual quality
  - patterns for adaptive detail adaptation of geometry or textures

- DFG: 1 doctoral researcher, 1 student assistant

[3]`http://univis.uni-erlangen.de` → Research projects → RAMP

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 9

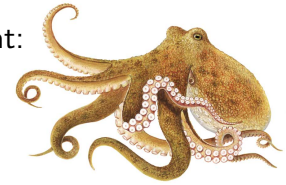
Run-Time Support System for Invasive Computing

[4] http://univis.uni-erlangen.de → Research projects → iRTSS

Run-Time Support System for Invasive Computing

Octo

- borrowed from the designation of a creature that:
  - i is highly parallel in its actions and
  - ii excellently can adapt oneself to its environment
- the kraken (species *Octopoda*)
  - can operate in parallel by virtue of its eight tentacle
  - is able to do customisation through camouflage and deimatic displays and
  - comes with a highly developed nervous system
    - in order to attune to dynamic ambient conditions and effects



POS

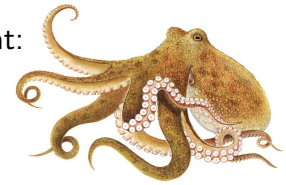
- abbrv. for *parallel operating system*
  - an operating system that not only supports parallel processes
  - but that also functions inherently parallel thereby

[4] http://univis.uni-erlangen.de → Research projects → iRTSS

Run-Time Support System for Invasive Computing

Octo

- borrowed from the designation of a creature that:
  - i is highly parallel in its actions and
  - ii excellently can adapt oneself to its environment
- the kraken (species *Octopoda*)
  - can operate in parallel by virtue of its eight tentacle
  - is able to do customisation through camouflage and deimatic displays and
  - comes with a highly developed nervous system
    - in order to attune to dynamic ambient conditions and effects



POS

- abbrv. for *parallel operating system*
  - an operating system that not only supports parallel processes
  - but that also functions inherently parallel thereby
- DFG: 2.5 doctoral researchers, 1 research/3 student assistants

[4] http://univis.uni-erlangen.de → Research projects → iRTSS

Power-Aware Critical Sections

PAX

[5] http://univis.uni-erlangen.de → Research projects → PAX

Power-Aware Critical Sections

- scalable synchronisation on the basis of **agile critical sections**
  - infrastructure
    - load-dependent and self-organised change of protection against race conditions
  - linguistic support
    - preparation, characterisation, and capturing of declared critical sections

[5] http://univis.uni-erlangen.de → Research projects → PAX

Power-Aware Critical Sections

- scalable synchronisation on the basis of **agile critical sections**
  - infrastructure
    - load-dependent and self-organised change of protection against race conditions
  - linguistic support
    - preparation, characterisation, and capturing of declared critical sections
- automated extraction of critical sections
  - notation language for critical sections
  - program analysis and LLVM integration/adaptation

[5] http://univis.uni-erlangen.de → Research projects → PAX

Power-Aware Critical Sections

- scalable synchronisation on the basis of **agile critical sections**
  - infrastructure
    - load-dependent and self-organised change of protection against race conditions
  - linguistic support
    - preparation, characterisation, and capturing of declared critical sections
- automated extraction of critical sections
  - notation language for critical sections
  - program analysis and LLVM integration/adaptation
- power-aware system programming
  - mutual exclusion, guarded sections, transactions
  - dynamic dispatch of synchronisation protocols or critical sections, resp.

[5] http://univis.uni-erlangen.de → Research projects → PAX

Power-Aware Critical Sections

- scalable synchronisation on the basis of **agile critical sections**
  - infrastructure
    - load-dependent and self-organised change of protection against race conditions
  - linguistic support
    - preparation, characterisation, and capturing of declared critical sections
- automated extraction of critical sections
  - notation language for critical sections
  - program analysis and LLVM integration/adaptation
- power-aware system programming
  - mutual exclusion, guarded sections, transactions
  - dynamic dispatch of synchronisation protocols or critical sections, resp.
- tamper-proof power-consumption measuring
  - instruction survey and statistics based on real and virtual machines
  - energy-consumption prediction or estimation, resp.

[5] http://univis.uni-erlangen.de → Research projects → PAX

Power-Aware Critical Sections

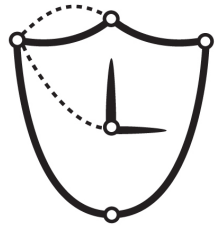
- scalable synchronisation on the basis of **agile critical sections**
  - infrastructure
    - load-dependent and self-organised change of protection against race conditions
  - linguistic support
    - preparation, characterisation, and capturing of declared critical sections
- automated extraction of critical sections
  - notation language for critical sections
  - program analysis and LLVM integration/adaptation

PAX

- power-aware system programming
  - mutual exclusion, guarded sections, transactions
  - dynamic dispatch of synchronisation protocols or critical sections, resp.
- tamper-proof power-consumption measuring
  - instruction survey and statistics based on real and virtual machines
  - energy-consumption prediction or estimation, resp.
- DFG: 2 doctoral researchers, 2 student assistants

[5] `http://univis.uni-erlangen.de` → Research projects → PAX

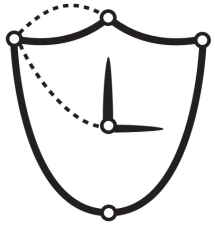
Latency- and Resilience-Aware Networking



[6] `http://univis.uni-erlangen.de` → Research projects → LARN

Latency- and Resilience-Aware Networking

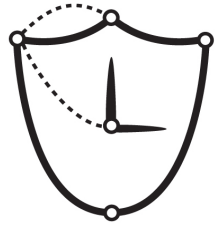
- **real-time capable network communication**
  - transport channel for cyber-physical systems
  - predictable transmission latency
  - in a certain extent guaranteed quality criteria



[6] `http://univis.uni-erlangen.de` → Research projects → LARN

Latency- and Resilience-Aware Networking

- **real-time capable network communication**
  - transport channel for cyber-physical systems
  - predictable transmission latency
  - in a certain extent guaranteed quality criteria



- **deterministic run-time support**

  Auffassung von der kausalen [Vor]bestimmtheit allen Geschehens bzw. Handelns (Duden)

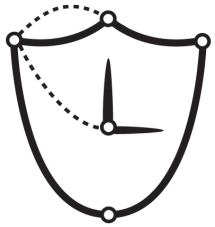
  - latency-aware communication endpoints, optimised protocol stack
  - sepcialised resource management, predictable run-time behavior

[6] `http://univis.uni-erlangen.de` → Research projects → LARN

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 12

Latency- and Resilience-Aware Networking

- **real-time capable network communication**
  - transport channel for cyber-physical systems
  - predictable transmission latency
  - in a certain extent guaranteed quality criteria



- **deterministic run-time support**

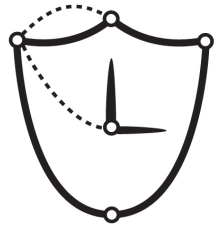
  Auffassung von der kausalen [Vor]bestimmtheit allen Geschehens bzw. Handelns (Duden)

  - latency-aware communication endpoints, optimised protocol stack
  - sepcialised resource management, predictable run-time behavior
    - in time (phase 1) and energy (phase 2) respect

[6] `http://univis.uni-erlangen.de` → Research projects → LARN

Latency- and Resilience-Aware Networking

- **real-time capable network communication**
  - transport channel for cyber-physical systems
  - predictable transmission latency
  - in a certain extent guaranteed quality criteria



- **deterministic run-time support**

  Auffassung von der kausalen [Vor]bestimmtheit allen Geschehens bzw. Handelns (Duden)

  - latency-aware communication endpoints, optimised protocol stack
  - sepcialised resource management, predictable run-time behavior
    - in time (phase 1) and energy (phase 2) respect

- DFG: doctoral researchers, 2 student assistants (1 FAU, 1 Uni SB)

[6] `http://univis.uni-erlangen.de` → Research projects → LARN

©wosch CS (WS 2017, LEC 14) Perspectives – Parallel Systems 12

Multi/Many-Core Processor Pool

faui4*	clock	cores per domain		domain		
		physical	logical	NUMA	tile	
8e 8f	2.9 GHz	8	16	2	–	Xeon
9big01	2.5 GHz	6	–	8	–	Opteron
9big02	2.2 GHz	10	20	4	–	Xeon
9phi01	1.2 GHz	6	12	2	–	Xeon
	1.1 GHz	57	228	2	–	Xeon Phi
scc	1.5 GHz	4	2	1	–	Xeon
	800 MHz	2	–	–	24	Pentium
InvasIC	3.5 GHz	8	16	2	–	Xeon
	25 MHz	4	–	6		LEON/SPARC

Multi/Many-Core Processor Pool

faui4*	clock	cores per domain		domain		
		physical	logical	NUMA	tile	
8e 8f	2.9 GHz	8	16	2	–	Xeon
9big01	2.5 GHz	6	–	8	–	Opteron
9big02	2.2 GHz	10	20	4	–	Xeon
9phi01	1.2 GHz	6	12	2	–	Xeon
	1.1 GHz	57	228	2	–	Xeon Phi
scc	1.5 GHz	4	2	1	–	Xeon
	800 MHz	2	–	–	24	Pentium
InvasIC	3.5 GHz	8	16	2	–	Xeon
	25 MHz	4	–	6		LEON/SPARC

- budgeted acquisition: further n-core systems, transactional memory

  OctoPOS ▪ $n \geq 64$, in 2015

  PAX ▪ $n \geq 16$, in 2016, plus several multi-core micro-controllers

©wosch CS (WS 2017, LEC 14) Perspectives – Computing Equipment 13

Bachelor, Master, or Doctoral Thesis