Concurrent Systems

Nebenläufige Systeme

XIV. Pickings

Wolfgang Schröder-Preikschat

January 29, 2015
Agenda

Recapitulation
  Concurrent Systems

Perspectives
  Parallel Systems
  Computing Equipment
  Further Education
Outline

Recapitulation
  Concurrent Systems

Perspectives
  Parallel Systems
  Computing Equipment
  Further Education
Content of Teaching and Cross-References

- Transactional memory
- PFP
- Elementary operations
- Critical sections
- Simultaneous (concurrent/interacting) processes
- Concurrency
- BS
- HBS ParAlg
- CC RA
- TRASYS
- Lock
- Semaphore
- Monitor
- Deadly embrace
- Guarded sections
- Non-blocking synchronisation
- Transactional memory
- Progress guarantee
Outline

Recapitulation
Concurrent Systems

Perspectives
Parallel Systems
Computing Equipment
Further Education
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**
  - resource-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**
  - resource-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**
  - resource-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

- 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

1http://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](http://univis.uni-erlangen.de)
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

\[1\text{http://univis.uni-erlangen.de} \rightarrow \text{Research projects} \rightarrow \text{LAOS}\]
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

Coherency Kernel

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
Coherency Kernel

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

\[\text{http://univis.uni-erlangen.de} \rightarrow \text{Research projects} \rightarrow \text{COKE}\]
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

\(^2\text{http://univis.uni-erlangen.de} \rightarrow \text{Research projects} \rightarrow \text{COKE}\)
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

---

²http://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
  - famille 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)**


\[\text{\textsuperscript{2}}http://univis.uni-erlangen.de \rightarrow \text{Research projects} \rightarrow \text{COKE}\]
Heterogeneous Resource-Aware Multi-Processing

GPU-centric resource management
- timely predictable run-time system
- run-to-completion kernel
- prioritisation 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-oriented memory management
- run-time adaptation and relocation of dynamic data structures
- tailor-made system software
- 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
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
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

---

3http://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

- **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

³http://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
  - prioritisation 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

³http://univis.uni-erlangen.de → Research projects → RAMP
Run-Time Support System for Invasive Computing

Octo is 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

\( \text{DFG: 2.5 doctoral researchers, 1 research/3 student assistants} \)

\[ \text{http://univis.uni-erlangen.de \rightarrow Research projects \rightarrow iRTSS} \]
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

\(^{4}\)\url{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 \rightarrow \text{Research projects} \rightarrow \text{iRTSS}
Power-Aware Critical Sections

- Scalable synchronisation on the basis of agile critical sections
- 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.

DFG: 2 doctoral researchers, 2 student assistants

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

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](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.

---

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.

---

5http://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.

- DFG: 2 doctoral researchers, 2 student assistants

5 http://univis.uni-erlangen.de → Research projects → PAX
## Multi/Many-Core Processor Pool

<table>
<thead>
<tr>
<th>faui4*</th>
<th>clock</th>
<th>cores per domain</th>
<th>domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>physical</td>
<td>logical</td>
</tr>
<tr>
<td>8e</td>
<td>2.9 GHz</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>8f</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9big01</td>
<td>2.5 GHz</td>
<td>6</td>
<td>–</td>
</tr>
<tr>
<td>9big02</td>
<td>2.2 GHz</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>9phi01</td>
<td>1.2 GHz</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1.1 GHz</td>
<td>57</td>
<td>228</td>
</tr>
<tr>
<td>sccc</td>
<td>1.5 GHz</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>800 MHz</td>
<td>2</td>
<td>–</td>
</tr>
<tr>
<td>InvasIC</td>
<td>3.5 GHz</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>25 MHz</td>
<td>4</td>
<td>–</td>
</tr>
</tbody>
</table>
**Multi/Many-Core Processor Pool**

<table>
<thead>
<tr>
<th>faui4*</th>
<th>clock</th>
<th>cores per domain</th>
<th>domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>physical</td>
<td>logical</td>
</tr>
<tr>
<td>8e</td>
<td>2.9 GHz</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>8f</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9big01</td>
<td>2.5 GHz</td>
<td>6</td>
<td>–</td>
</tr>
<tr>
<td>9big02</td>
<td>2.2 GHz</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>9phi01</td>
<td>1.2 GHz</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1.1 GHz</td>
<td>57</td>
<td>228</td>
</tr>
<tr>
<td>scc</td>
<td>1.5 GHz</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>800 MHz</td>
<td>2</td>
<td>–</td>
</tr>
<tr>
<td>InvasIC</td>
<td>3.5 GHz</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>25 MHz</td>
<td>4</td>
<td>–</td>
</tr>
</tbody>
</table>

**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