

Predictable Many-core Processor

Eduardo Quiñones

(BSC)

Sascha Uhrig

(Technical University of Dortmund)

parMERASA Dissemination Event,
Barcelona, 23rd September 2014



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

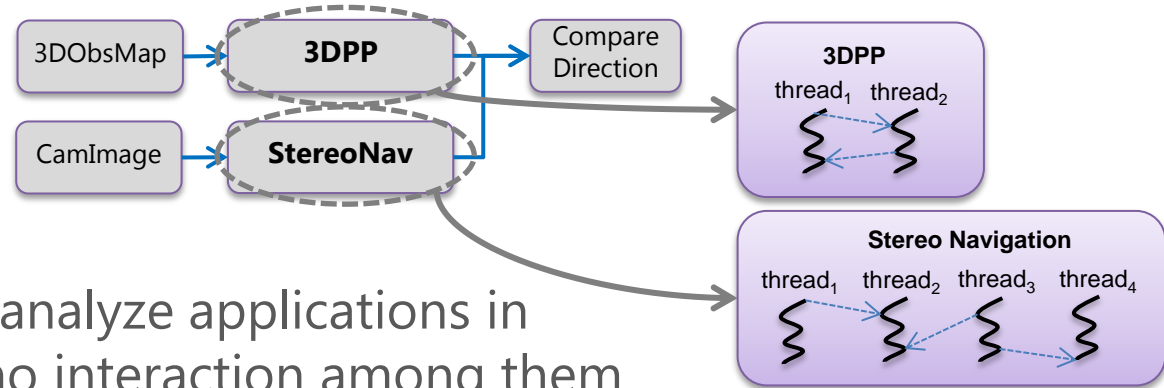


technische universität
dortmund

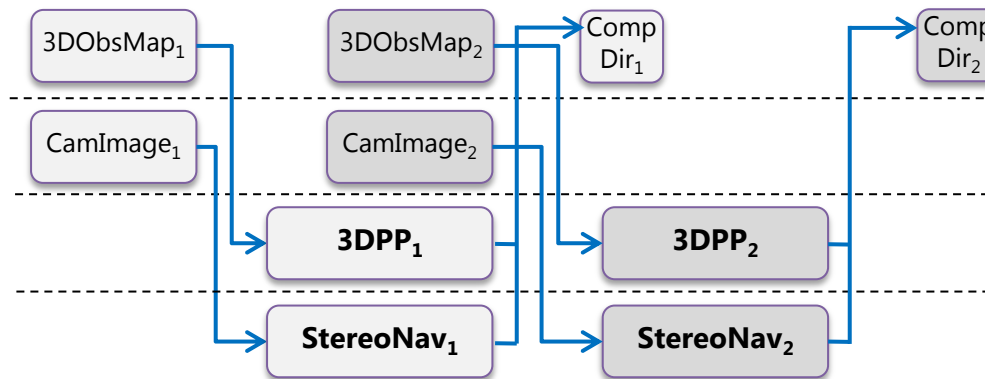


Universität
Augsburg
University

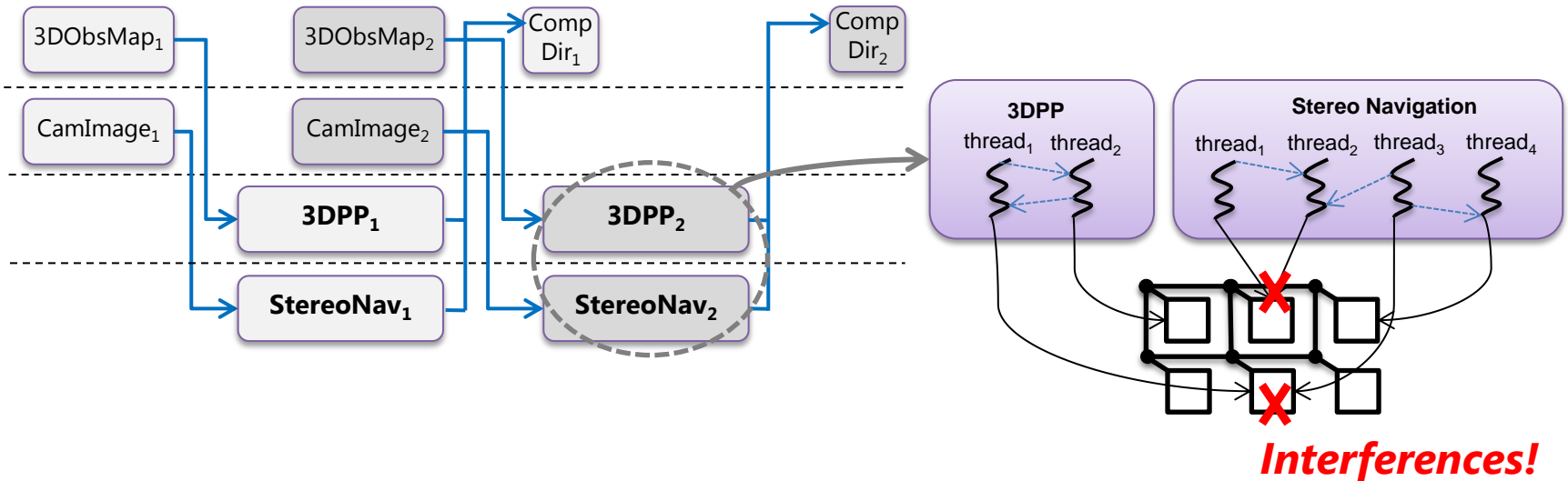
- Safety critical real-time systems rely on **composability**
 - The functional and timing behaviour of applications in isolation remains the same after system integration
 - Named *incremental qualification* in avionics or *freedom from interference* in automotive
- ARINC 653 and ISO 26262 define a SW/HW mechanism to prevent interferences among applications
 - Enabled by guaranteeing **robust partitioning** among applications
 - **Software partitions** as the incarnation of the robust partitioning
 - Applied only in **single-core** architectures



1. Parallelize and analyze applications in isolation with no interaction among them

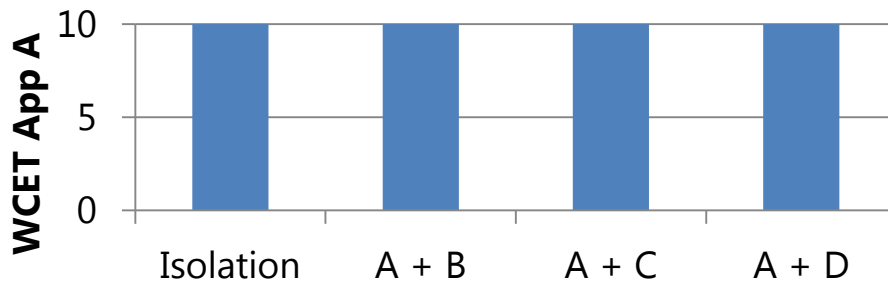


2. Integrate applications into the system maintaining their timing properties (composability)



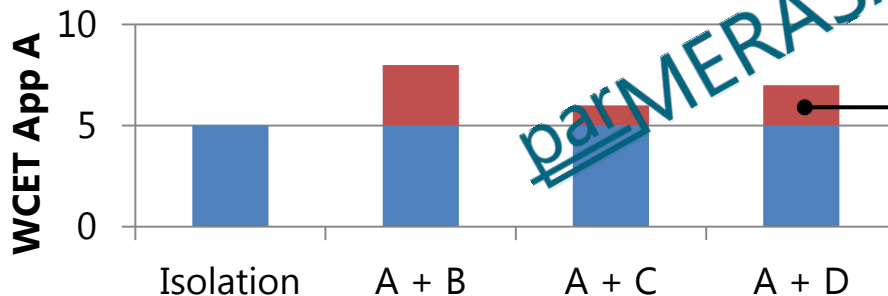
- Threads will compete for processor resources (e.g. NoC, Memory) **breaking the time composability** property
 - Communication **internal** to parallel applications (**intra**) -->
 - Application design
 - Communication **among** applications (**inter**) →
 - System integration

1. Timing analysis of applications considers the impact of intra and inter



$$WCET_{integration} = WCET_{analysis_isolation}$$

1. Timing analysis of applications considers only intra; inter is considered at system integration



Impact of system integration:

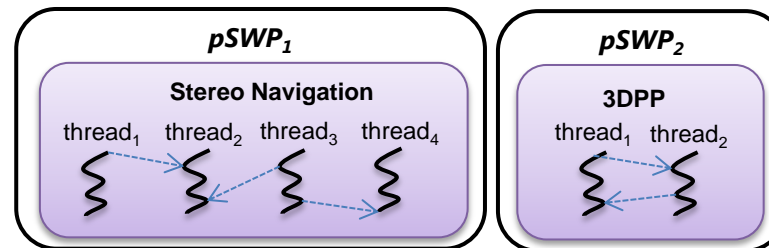
$$WCET_{integration} = WCET_{analysis_isolation} + \Delta_{inter}$$

- Time composability is provided partially at application level, a.k.a. **time compositionality**
 - Timing analysis of applications is done in isolation (intra)
 - The impact of system integration (inter) is considered as an additive factor
 - The inter-application communication is known at system integration

$$\mathbf{WCET}_{\text{integration}} = \mathbf{WCET}_{\text{analysis_isolation}} + \Delta_{\text{inter}}$$

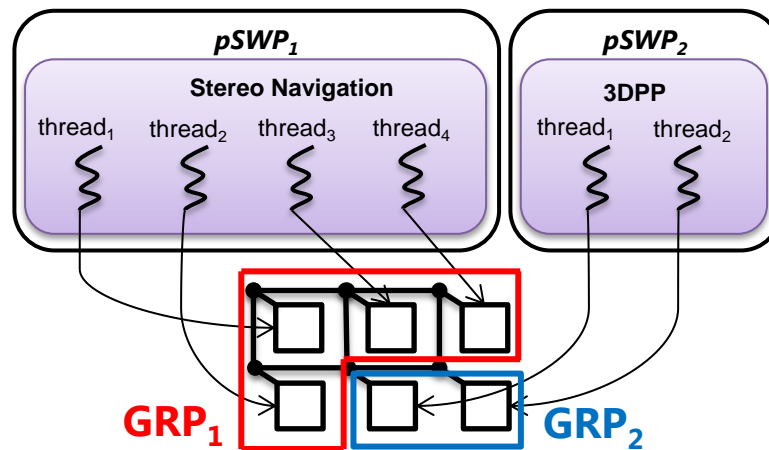
- parMERASA architecture implements mechanisms that guarantees **time isolation**
 - Parallel Software Partitions at software level
 - Guaranteed Resource Partitions at hardware level

- Applications are **encapsulated within pSWP**
 - Threads from a pSWP **cannot affect** threads from a different pSWP being run simultaneously
 - Threads from the same pSWP **do not** fulfill the robust partition
 - Interferences must be considered by the timing analysis

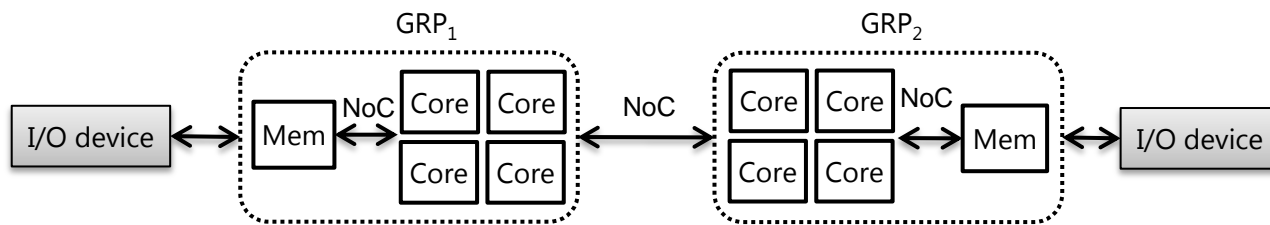


- New application **communication rules**
 - Intra-partition communication cannot exceed pSWP boundaries
 - Inter-partition communication must be considered as an additive factor (Δ_{inter})

- Hardware execution environment composed of a **pool of processor resources** (including cores, NoC, memories, etc.) in which pSWPs run
 - Intra-partition communication requests **cannot exceed GRPs**
 - Inter-partition communication requests **must be considered at integration time as an additive factor (Δ_{inter})**
- GRPs are the hardware counterpart of pSWP

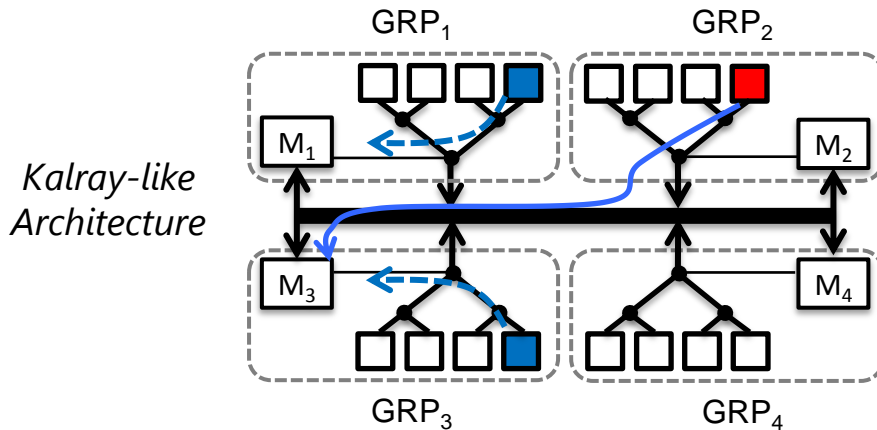


- Clusterized architecture **supporting GRP definition**
 - Cores organized in (virtual) clusters
 - NoC connecting cores and clusters
 - Private memory and I/O per cluster (accessible by all cores)

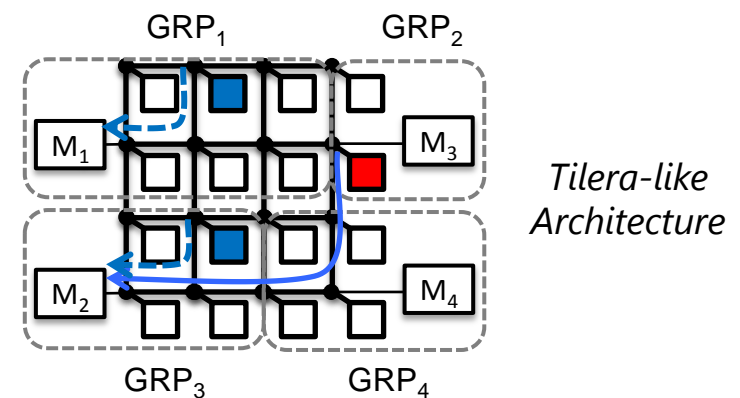


- All components are **time predictable**
 - Worst case traversal time (WCTT)* for NoCs
 - Worst case response time (WCRT)* for the memory hierarchy
 - Predictable cache coherence protocol

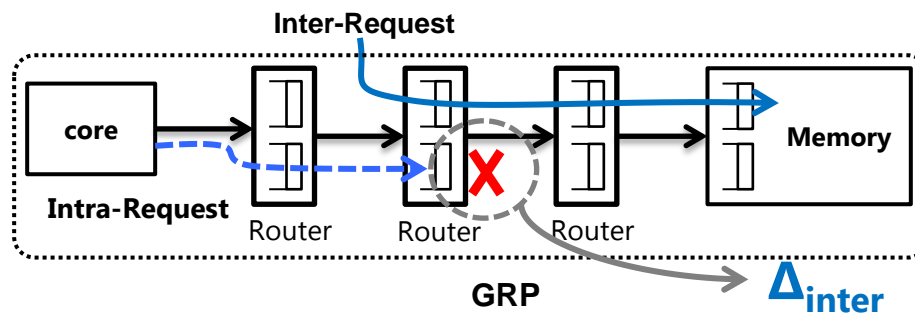
Hardware Cluster Definition



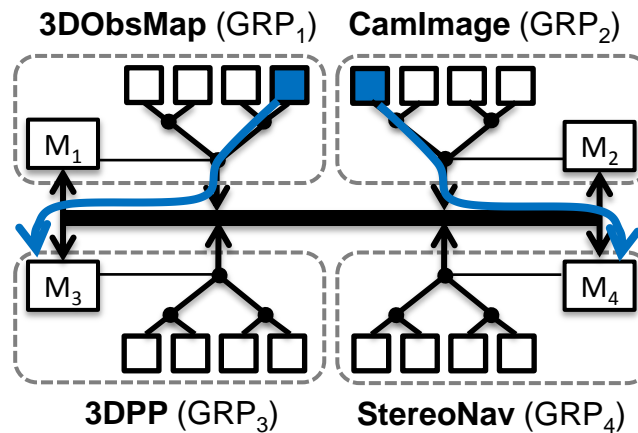
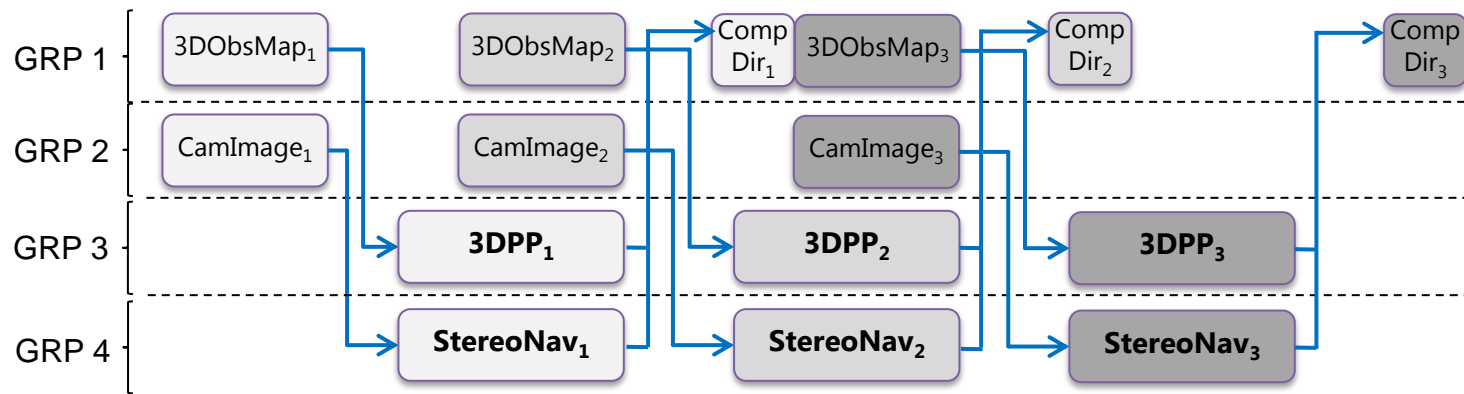
Virtual Cluster Definition (XY routing)



- Intra- considered at **application analysis** ($WCET_{isol}$)
- Inter- considered at **system integration** ($WCET_{int} = WCET_{isol} + \Delta_{inter}$)
 - Computed by separating the communication flows in NoC and memory

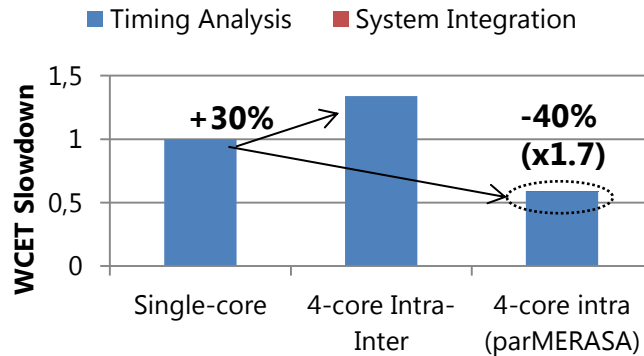


- System integrated of 3DPP and Stereo Navigation executed in a software pipelining fashion

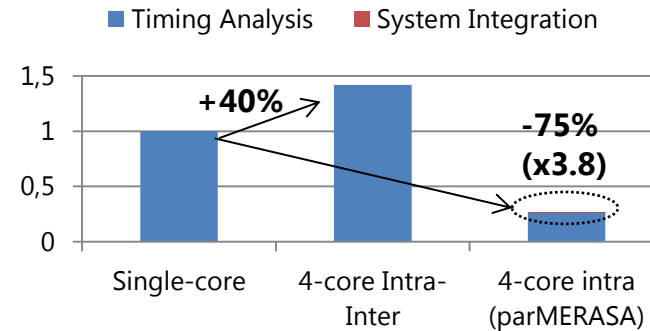


Inter-partition communication

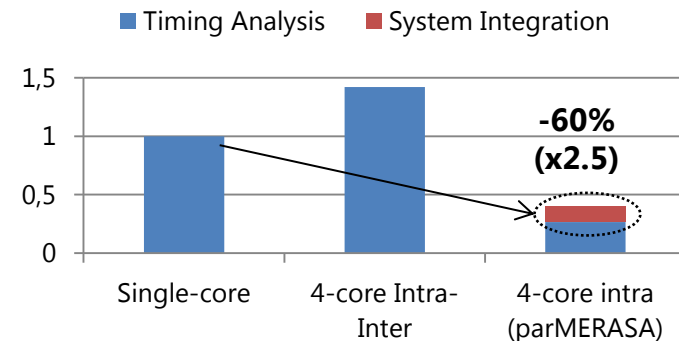
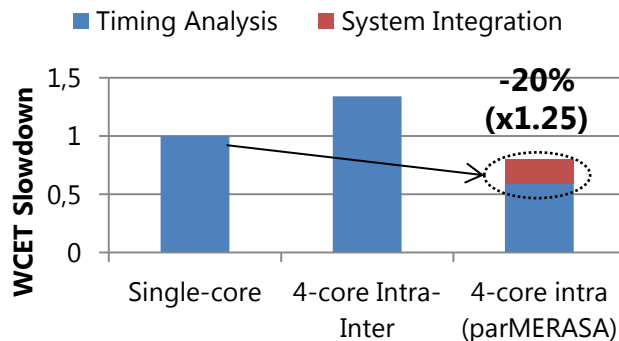
3DPP



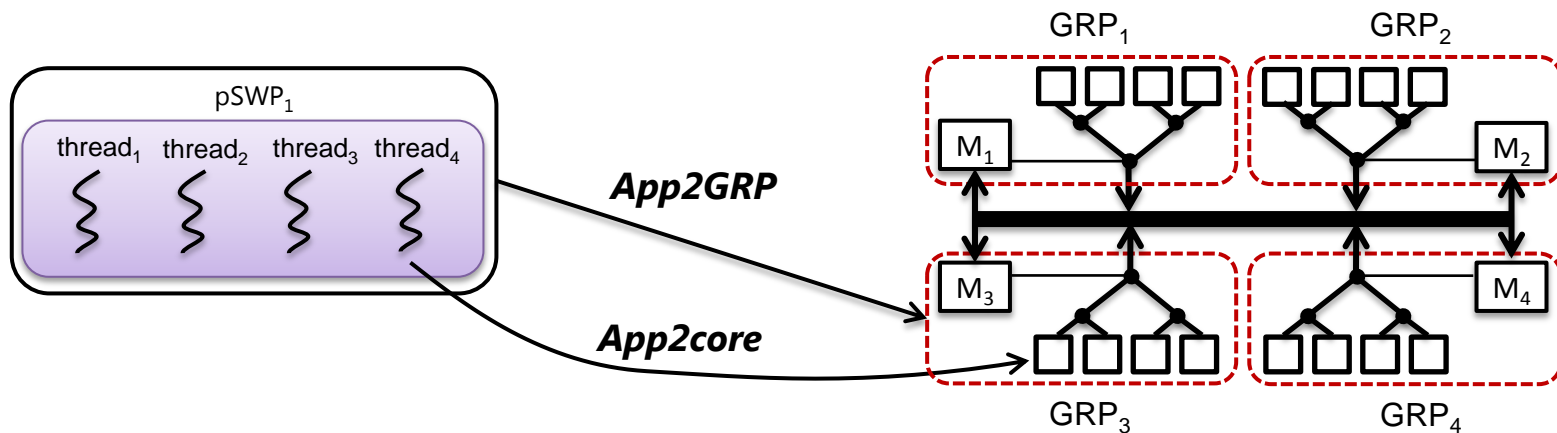
Stereo Navigation



Artificially increase the inter-partition communication by **x100**



- The partition allocation affects the system performance
 - *App2core* mapping \rightarrow $WCET_{isolation}$ is derived
 - *App2GRP* mapping \rightarrow $WCET_{integration}$ is computed minimizing Δ_{inter}



- Memory becomes a more **SERIOUS bottleneck** when increasing the number of cores
 - Local cache per core necessary
 - Accesses to shared memory required because of large input data or shared data
 - What's about **consistency** of shared memory accesses?
 - Need to keep all caches consistent/coherent
 - Means a predictable coherency protocol
 - State-of-the-art: software-based coherence
 - Means to access shared data w/o cache support, or
 - Additional functionality to invalidate/update individual cache lines
- ⇒ **Demand for interference-free hardware coherence technique**

- No cache coherency during *normal* program execution
- Consistency of memory accesses only in special situations
 - Shared data is protected by a mutex or semaphore
 - In code regions organized by barriers
- The ODC² continuously guarantees access to updated shared data
 - Only one thread can access a portion of shared data at any point in time (except of read-only data)
 - ODC² switches to *write-through* policy to keep memory updated
- After leaving the critical region shared data is invalidated
 - Reloading required at next access

Mutexes, critical sections

... Accesses to private data only ...

`lock(critical_section);`

`enter_shared_mode();`

... Accesses to private and shared data ...

`exit_shared_mode();`

`unlock(critical_section);`

... Accesses to private data only ...

Barriers

`enter_shared_mode();`

... Accesses to private and shared data ...

`exit_shared_mode();`

`barrier_wait(barrier_1);`

`enter_shared_mode();`

... Accesses to private and shared data ...

`exit_shared_mode();`

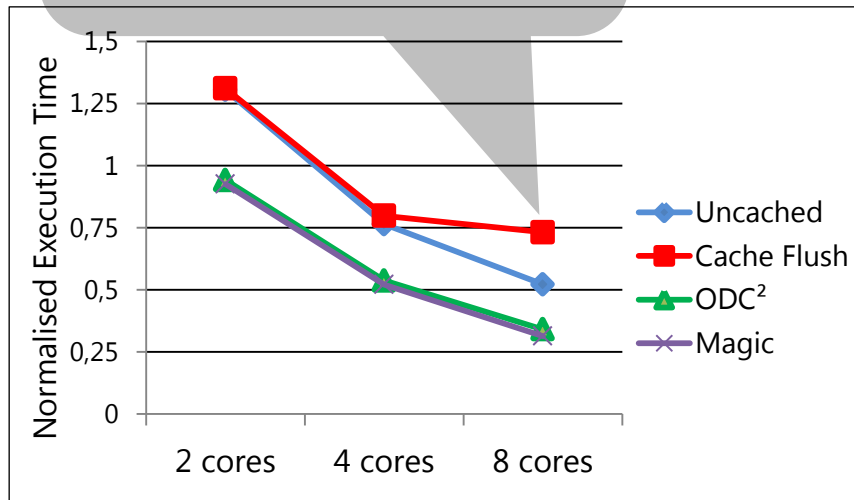
`barrier_wait(barrier_2);`

ODC² operates in Shared Mode.

Impact on timing analysability:

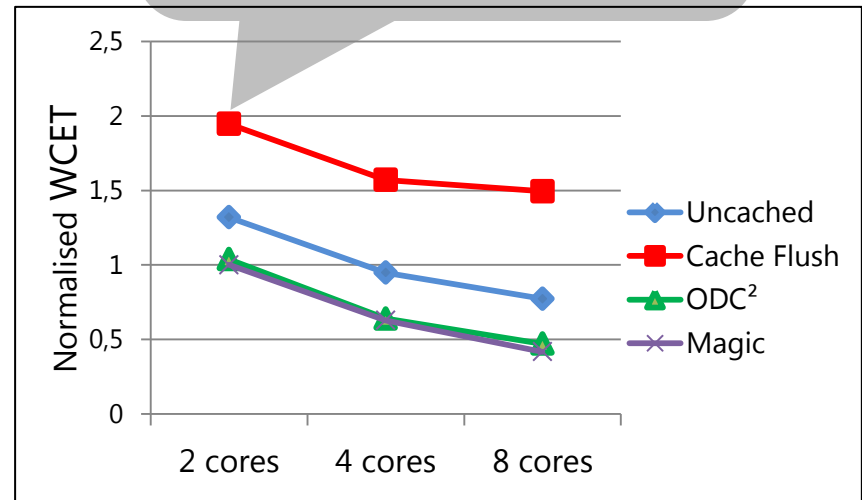
- No external modifications of cache lines

Write-through on all data leads to high memory pressure
node must

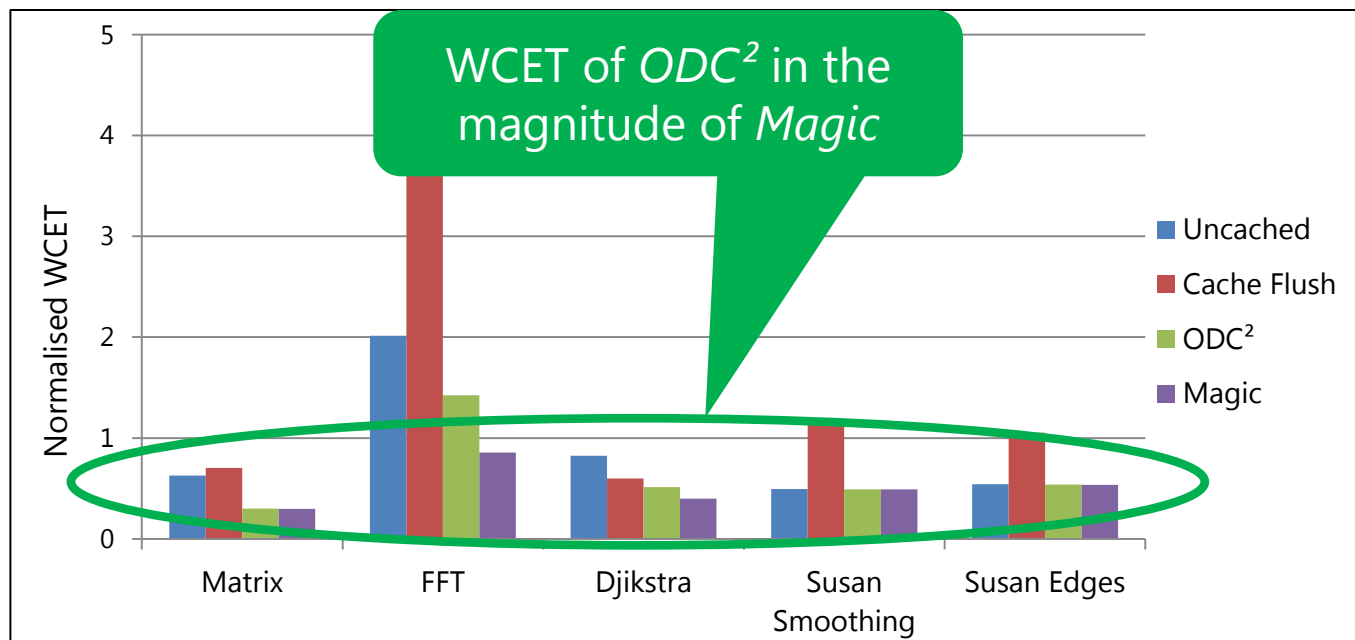
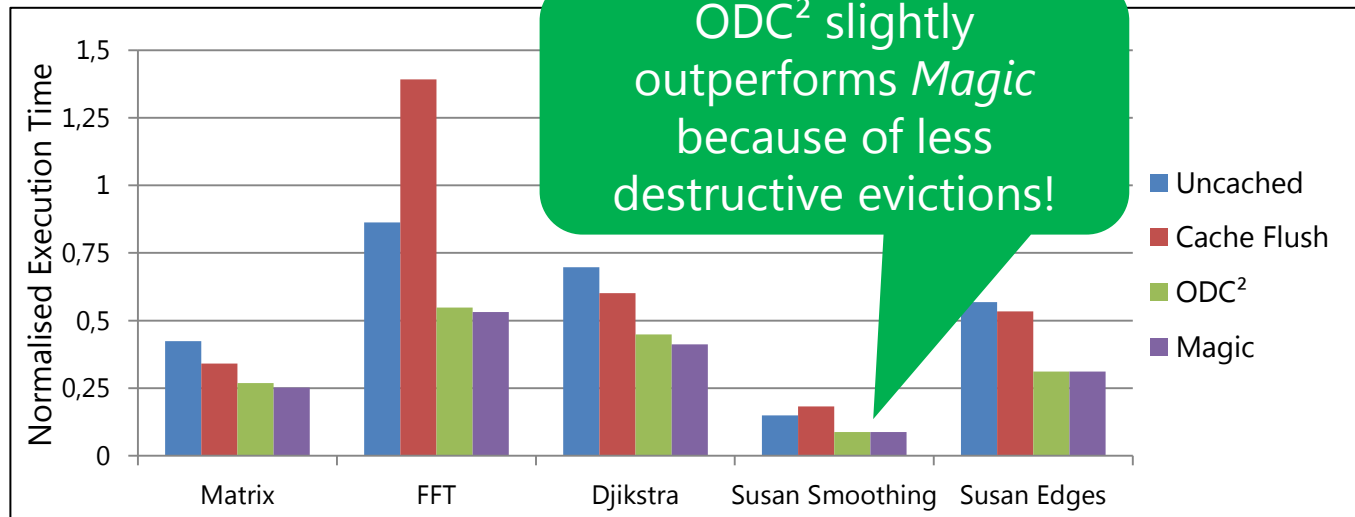


Impact on performance:

- Forced invalidation increases cache-miss rate
 - Latency due to cache invalidation
 - Invalidation of cache lines
- 3DPP not optimized for dual-core, better results with more cores

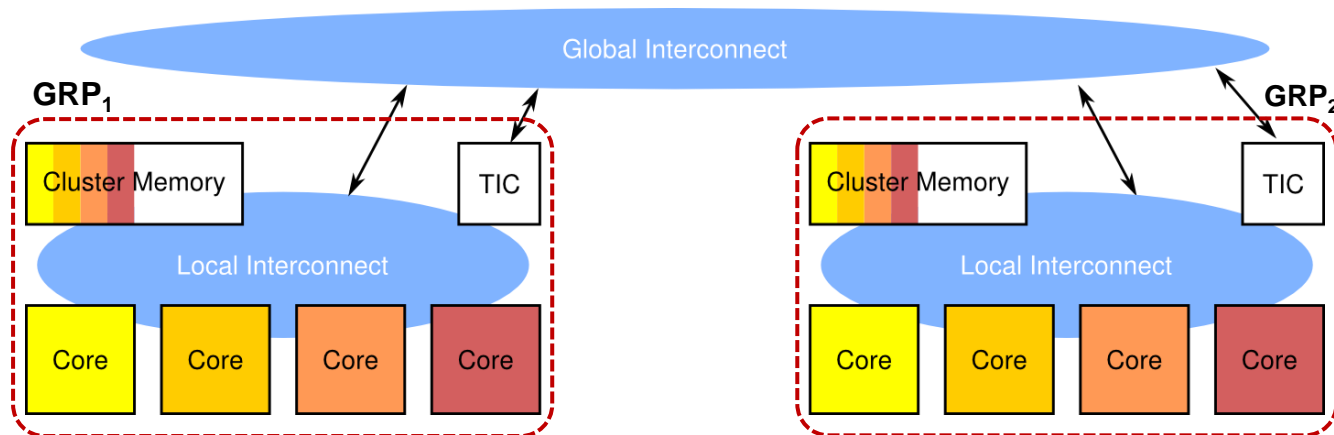


3DPP using a tree network, *Uncached* implements bypassing of cache for shared data, *Cache Flush* implements invalidation of complete cache at synchronisation points.



Several (micro) benchmarks:
Uncached implements bypassing of cache for shared data,
Cache Flush implements invalidation of complete cache at synchronisation points.

- Global physical **shared address space**
- Logical address space equal for each core
 - Private memory
 - No coherency needed
 - Shared memory
 - ODC² mechanism used
 - Used to communicate processes within GRP and applications among GRPs
 - OS functions for inter-partition communication
 - Memory protection mechanisms



- The parMERASA architecture provides time compositionality to reduce the WCET estimates
 - The impact of system integration on WCET estimates is considered as an additive factor ($+\Delta_{\text{inter}}$)
 - Parallel Software Partitions (pSWP) at software level
 - Guaranteed Resource Partitions (GRP) at hardware level
 - Predictable memory hierarchy including a novel cache coherence mechanism
- The techniques presented have been implemented into a simulator infrastructure
- **The parMERASA architecture allows executing hard real-time systems with low development overhead**

Predictable Many-core Processor

Eduardo Quiñones

(BSC)

Sascha Uhrig

(Technical University of Dortmund)

parMERASA Dissemination Event,
Barcelona, 23rd September 2014



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación



technische universität
dortmund



Universität
Augsburg
University