TECHNISCHE
UNIVERSITÄT
DRESDEN

Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH)

Hauptseminar
Rechnerarchitektur und Programmierung

Adapteva Parallella:

Crowd-funded Low-budget Open-source HPC

Tobias Frust (tobias.frust@mailbox.tu-dresden.de)

Tutor: Ronny Brendel (ronny.brendel@tu-dresden.de)

17th July, 2015

ZIH
Zentrum für Informationsdienste
und Hochleistungsrechnen

Structure

- The Adapteva Parallella Platform
  - Company history of Adapteva
  - Motivation
  - The Parallella board
  - The Epiphany coprocessor
- Implementation of the 1D-FFT with filtering
  - Motivation
  - Implementation details
  - Performance results
- Comparison with current architectures - GPUs
- Assessing the Parallella's Potential for HPC

Company history of Adapteva

- Feb 2008: Founded by Andreas Oloffson
  - Goal: 10 times advancement in floating point processing energy efficiency
- Jun 2009: Tapeout of Epiphany-I prototype (65nm)
  - Secured 1.5M US$ in Series-A funding from Bittware
- May 2011: Sampled Epiphany-III (65nm) product
- Aug 2012: Demonstration of 50 GFLOPS/Watt efficiency at 28nm
- Oct 2012: Launch of Parallella kickstarter project
- Jul 2013: first Parallella boards shipped to Kickstarter backers
- Apr 2014: Completed shipping of all Parallella boards to Kickstarter backers

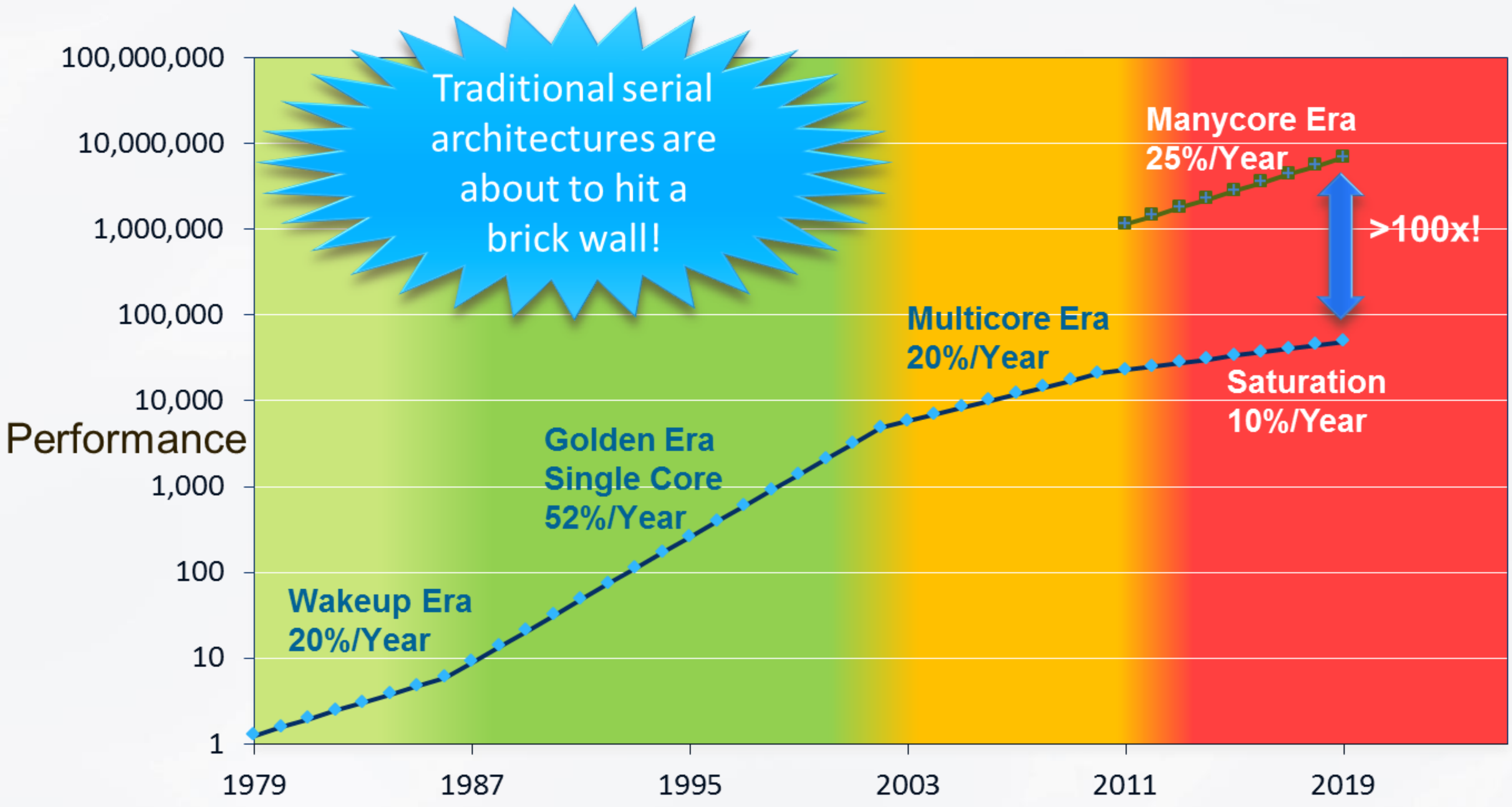
Parallella Kickstarter Program

- Slogan: *“The parallella project will make parallel computing accessible to everyone”*
- Kickstarter project gained 898,921$ from 4,965 backers
- Attributes:
  - **Open Access:** no NDAs or special access needed
  - **Open Source:** platform based on free open source development tools and libraries
  - **Affordable:** Parallella high performance computer at costs below 100$

→ Close the knowledge gap in parallel programing

→ Democratize access to parallel computing

Motivation for Epiphany multicore architecture

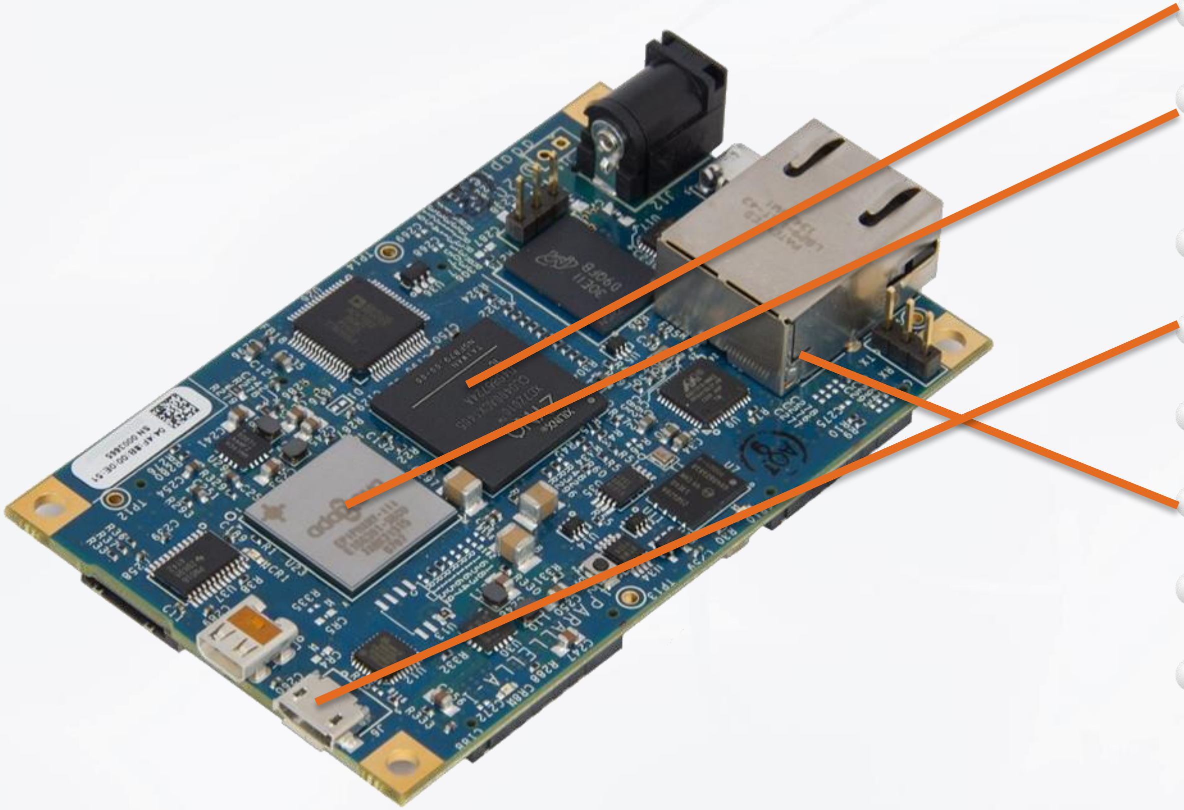


Memory Bandwidth Challenge

Programming Challenge

Scaling Challenge

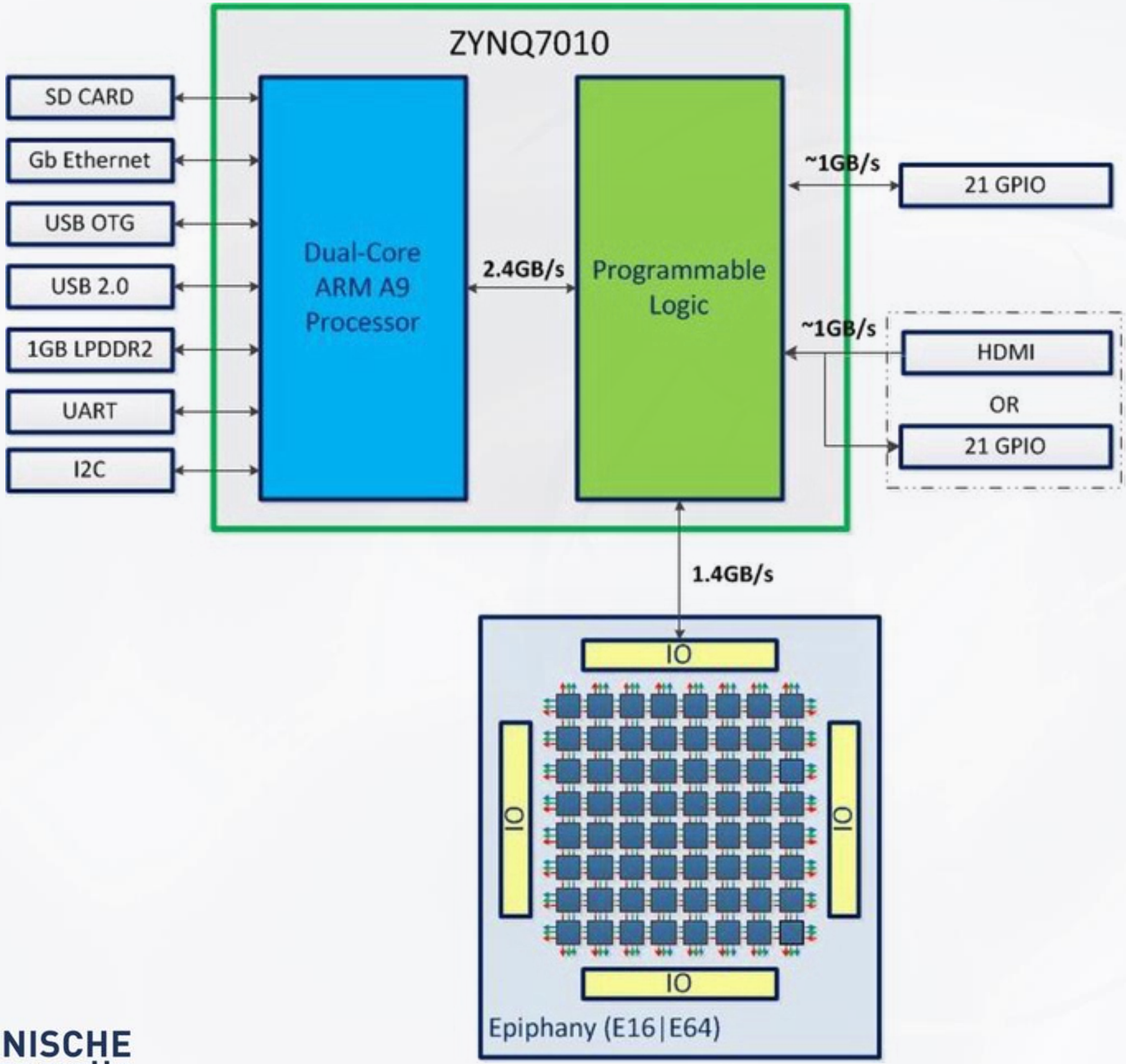
Parallella Board



Tech specs:

- Dual-core ARM A9
- 16-core Epiphany Coprocessor
- 1 GB RAM
- MicroSD Card
- USB 2.0
- Gigabit Ethernet
- Linux
- Peak Performance: 26 GFLOP/s (single precision)

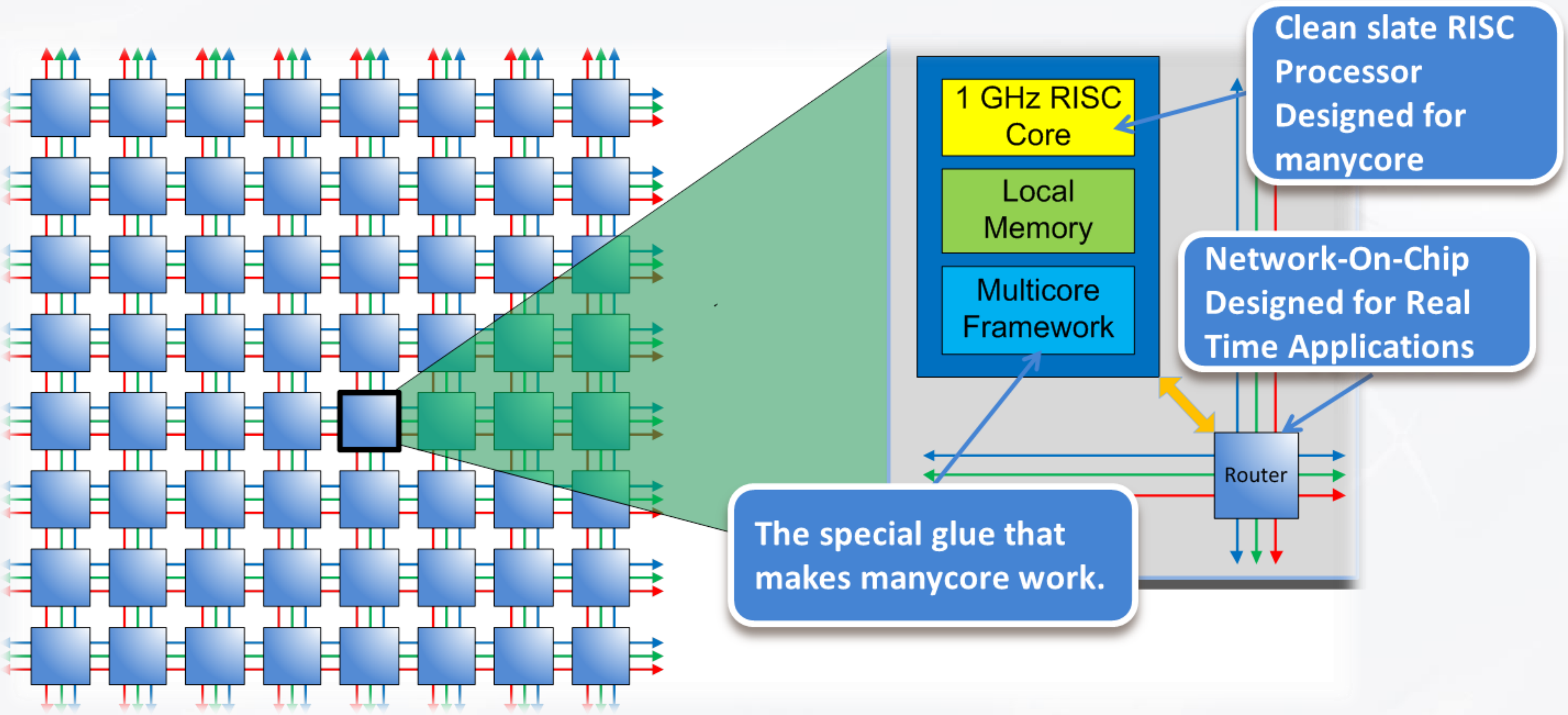
System overview of Parallella board



TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Zentrum für Informationsdienste und Hochleistungsrechnen

High level overview of the Epiphany architecture



Clean slate RISC Processor Designed for manycore

Network-On-Chip Designed for Real Time Applications

The special glue that makes manycore work.

Coprocessor to ARM/Intel CPU

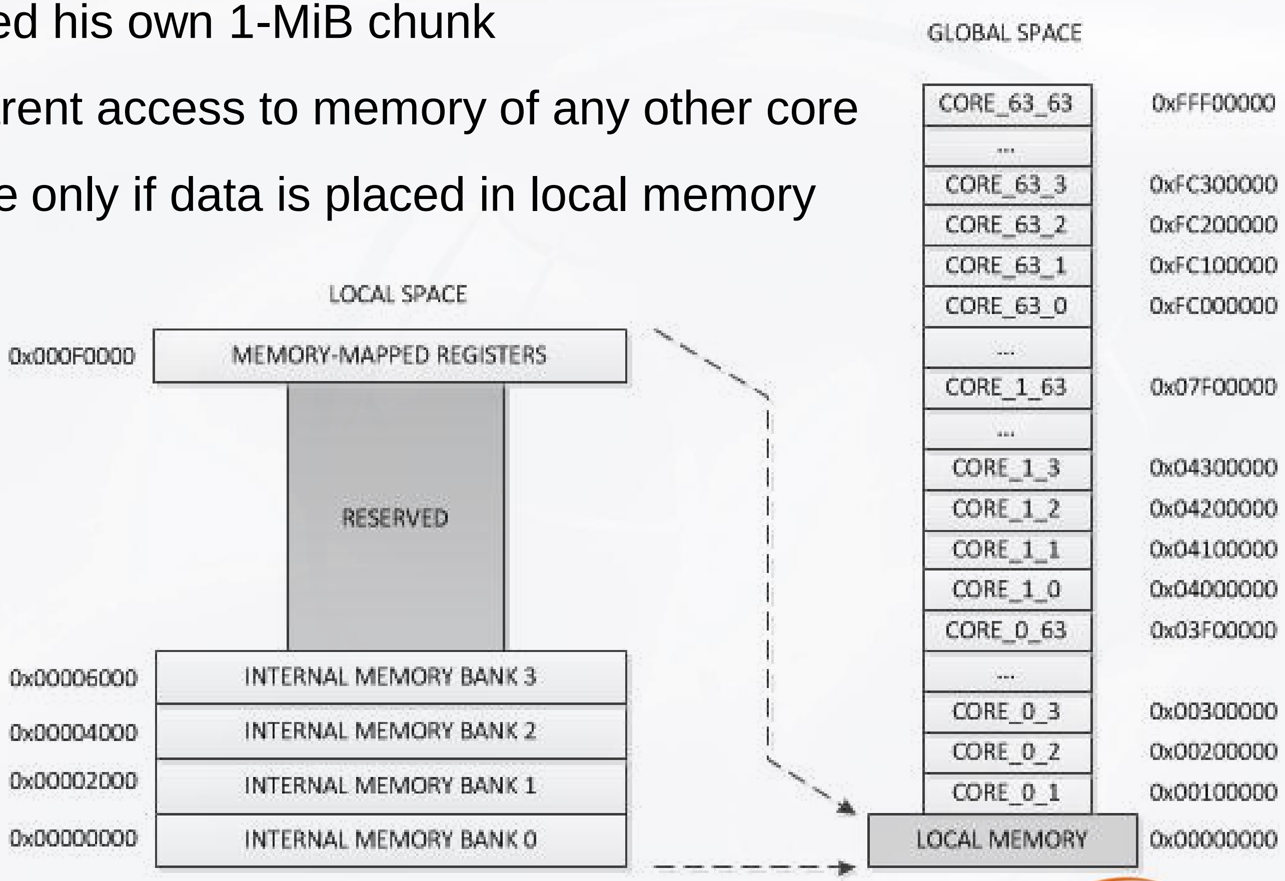
25mW per core

Ease To Use

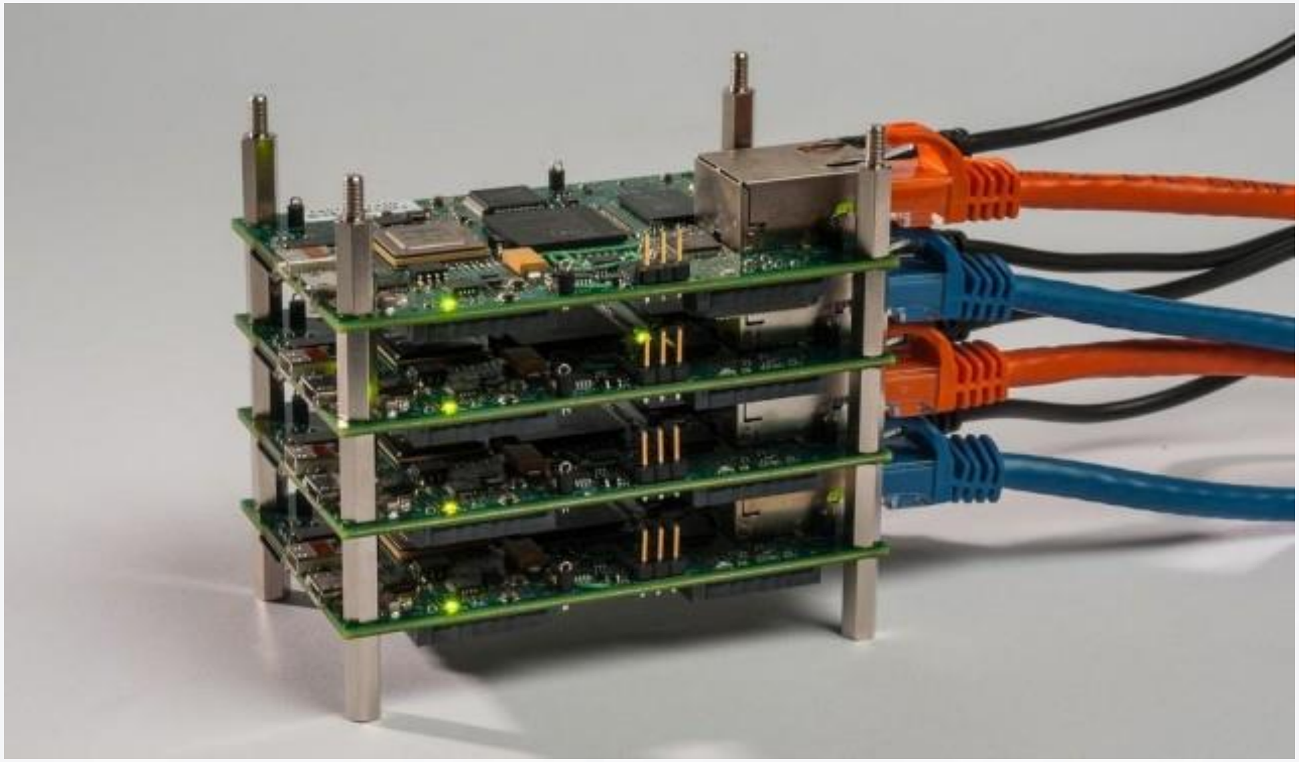
Memory scheme of the Epiphany architecture

Attributes:

- flat 32 bit address space split into 4096 1-MiB chunks
- Each core is assigned his own 1-MiB chunk
  - But has transparent access to memory of any other core
- Optimal performance only if data is placed in local memory banks



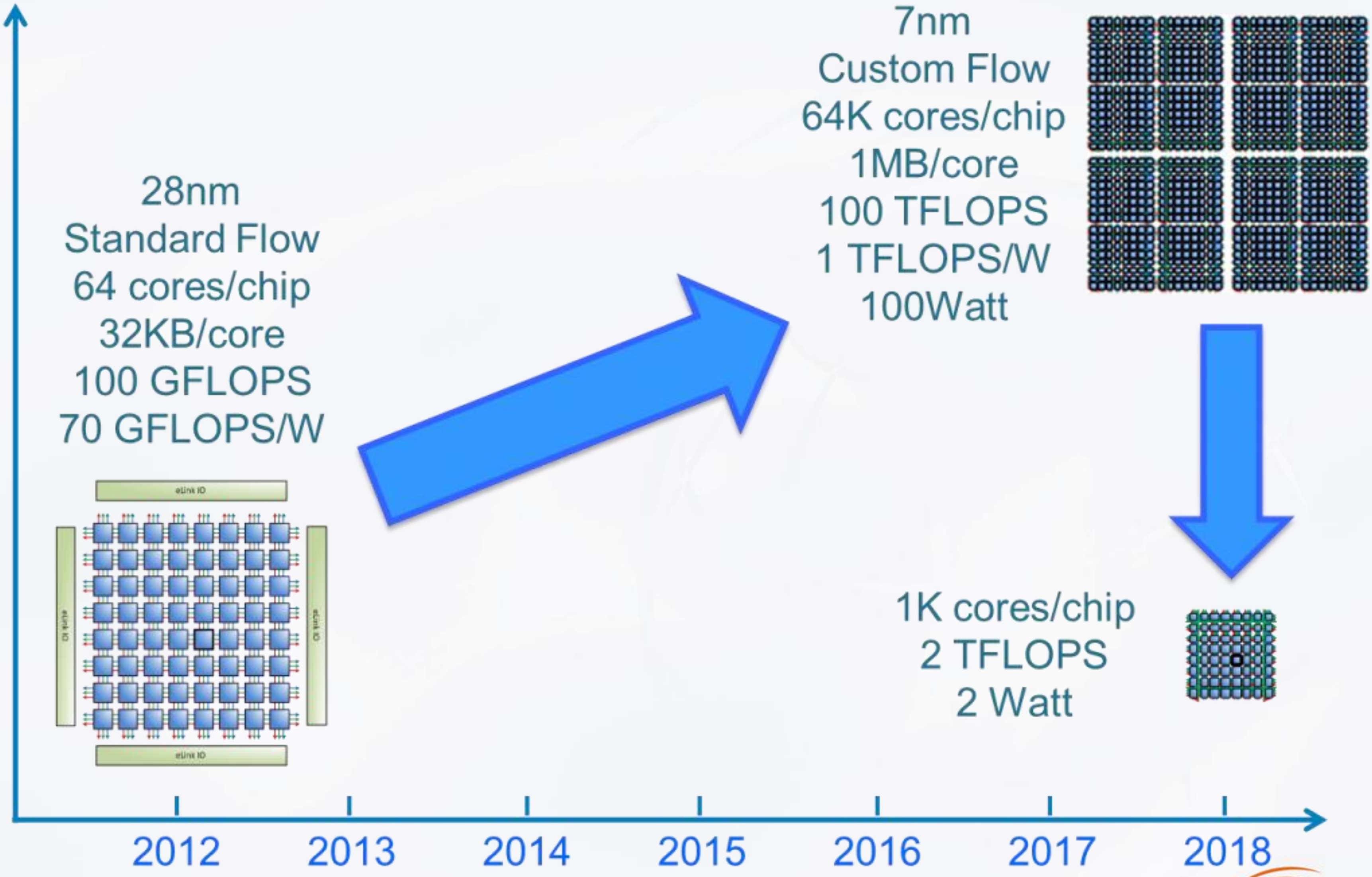
Example: Parallella as a cluster



- Boards can be combined to one cluster
- Gigabit-Ethernet interconnect

→ With thousands of boards put together you can build a supercomputer at low cost

Future ideas for Epiphany architecture



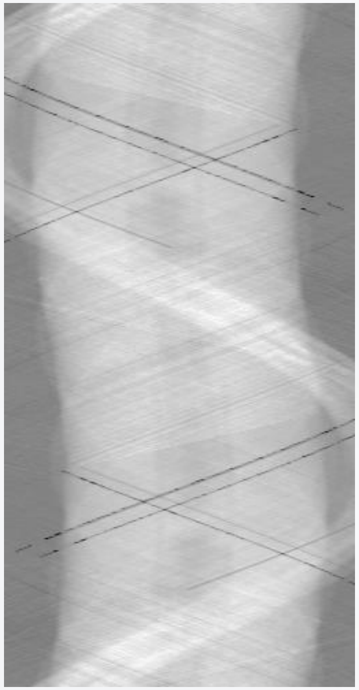
28nm
Standard Flow
64 cores/chip
32KB/core
100 GFLOPS
70 GFLOPS/W

7nm
Custom Flow
64K cores/chip
1MB/core
100 TFLOPS
1 TFLOPS/W
100Watt

1K cores/chip
2 TFLOPS
2 Watt

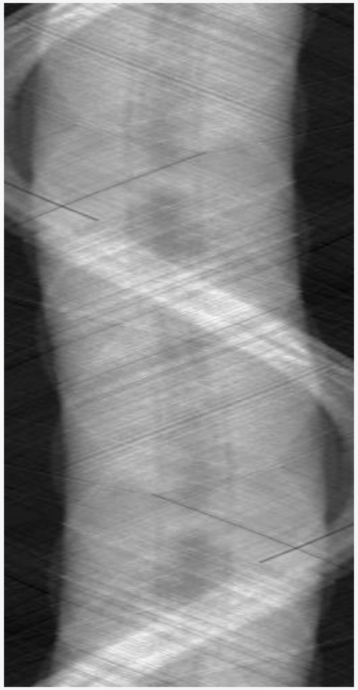
Motivation: Implementation of the 1D-FFT on Epiphany

- Fast Fourier Transform is basic operation for many applications
- Algorithm to compute the discrete fourier transform
- Concrete example: Filtered backprojection as CT-Reconstruction algorithm

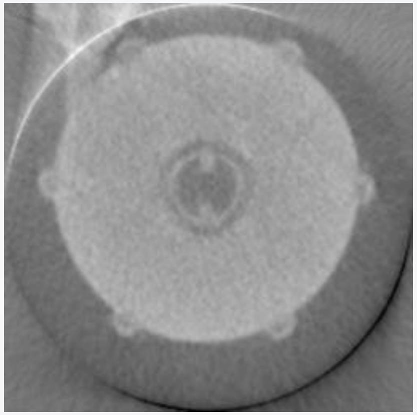


FFT + Filtering

+ Inverse FFT



Backprojection



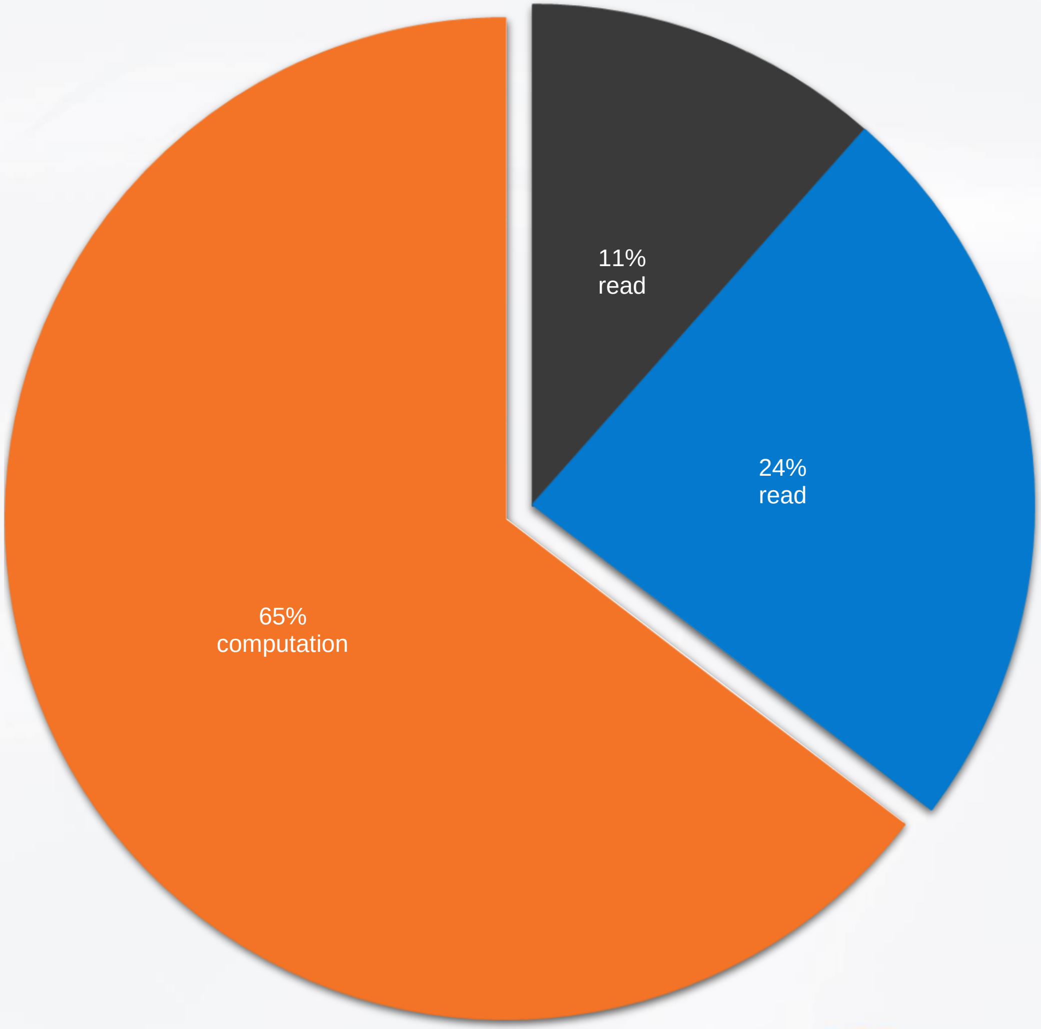
Implementation on Parallella

Details of implementation

Data transfer vs calculation time (1 core)

- Data transfer is the main bottleneck
- To increase performance, more calculation per memory transfer is necessary
- Due to Amdahls law speedup cannot exceed 2.5 with 16 cores
- Maximum speedup of 2.9 possible

→TODO: reason



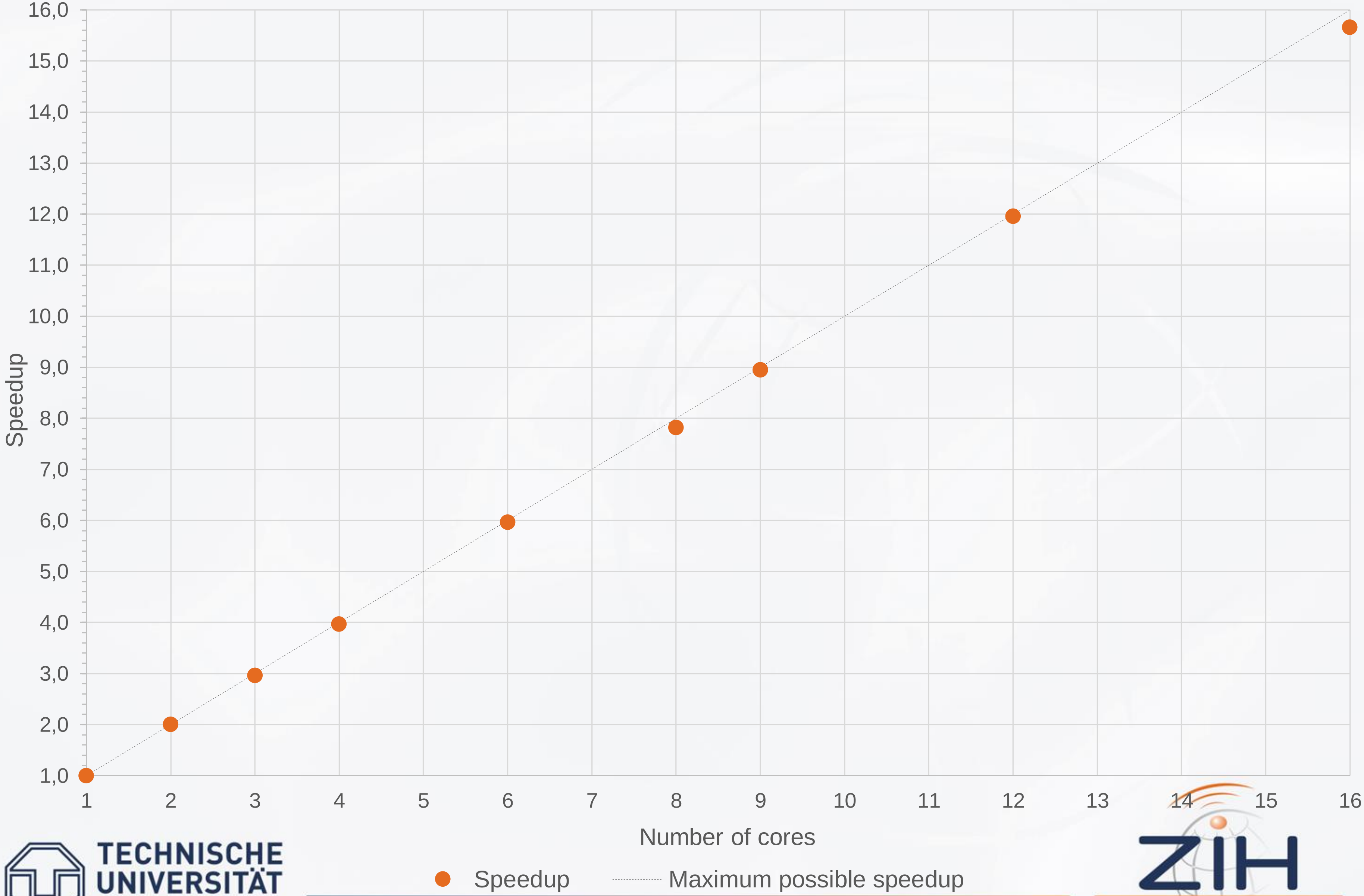
Performance results

- Results measured for 500 FFTs of size 256

	1 core	**16 cores**
Write time	70 ms	72 ms
Computation time	394 ms	25 ms
Read time	146 ms	148 ms
Number of floating point operations	$13{,}9 \cdot 10^6$	$13{,}9 \cdot 10^6$
MFLOP/s in calculation part	35 MFLOP/s	556 MFLOP/s
MFLOP/s in whole program	23 MFLOP/s	57 MFLOP/s
Transfer rate: write	14,6 MByte/s	14,2 MByte/s
Transfer rate: read	7,01 MByte/s	6,9 MByte/s

- Small memory transfer rate to local memory banks
- Read = read + write → half performance compared to write

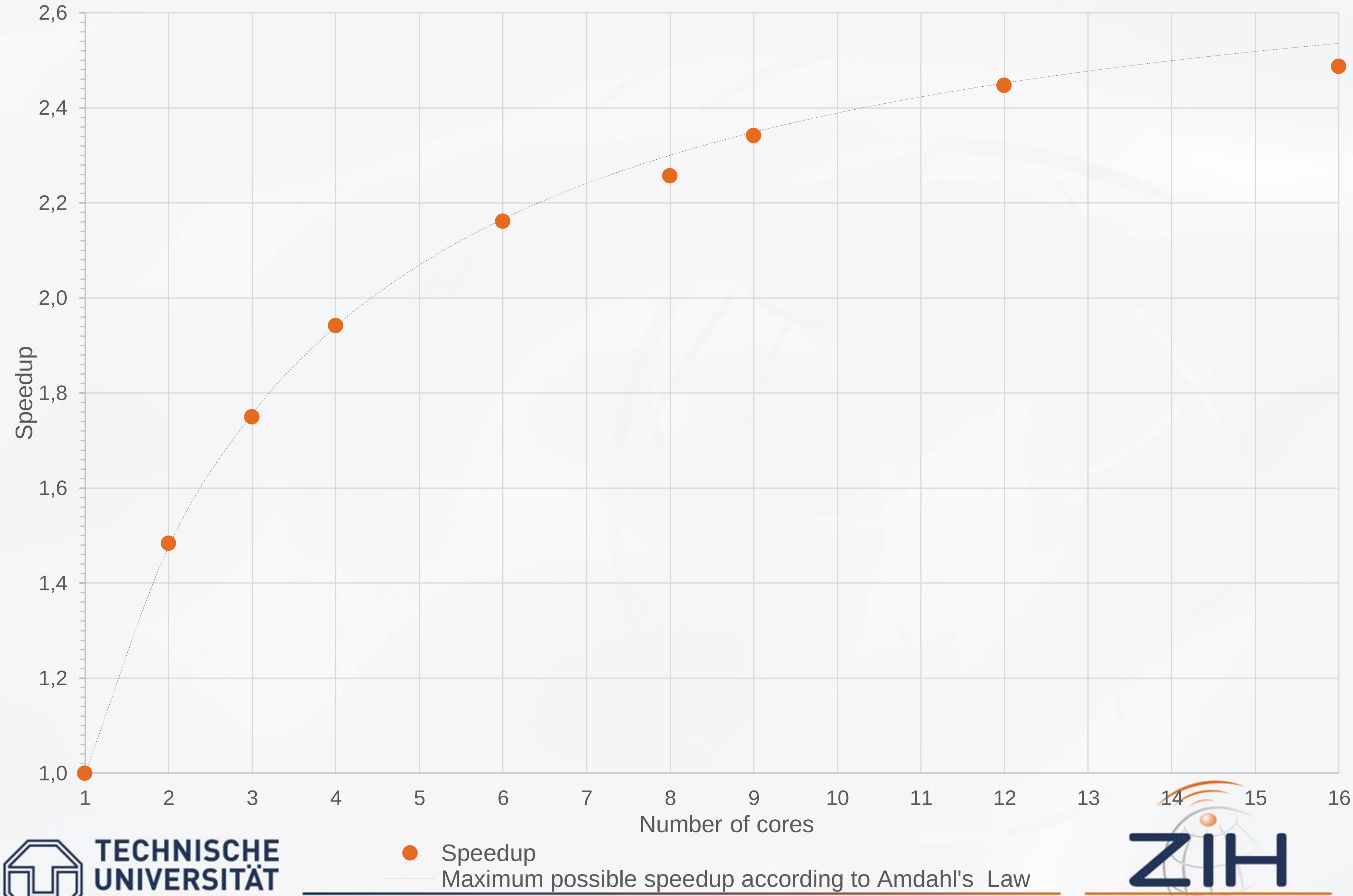
Speedup of parallelisable calculation part



TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Zentrum für Informationsdienste und Hochleistungsrechnen

Overall Speedup



Comparison with actual architecture - GPUs

- Comparison with *Intel Core i7 2600* combined with *Nvidia Geforce 750Ti* in terms of energy efficiency
- Identical implementation with use of *cufft*-Library from Nvidia

Peak Performance Single Precision	1,472 TFLOP/s
Number of Streaming Multiprocessors (SMs)	5
Number of cuda cores	640
Memory Bandwidth	86,4 GByte/s
Architecture	Maxwell
Memory bus interface	128 Bit
Manufacturing Process	TSMC 28nm
Thermal Design Power (TDP)	60 Watt
Launch Date	02/18/14
Die size	148 mm^2

Comparison of Parallella with GeForce 750 Ti

- To stay fair comparison in terms of energy efficieny and chip area