

# Zwischenpräsentation: Profilmodul Forschungsprojekt Grundlagen

Optimierung eines CT-Rekonstruktions-Programms im Hinblick  
auf die effiziente Nutzung eines Intel® Xeon Phi™

Jan Stephan

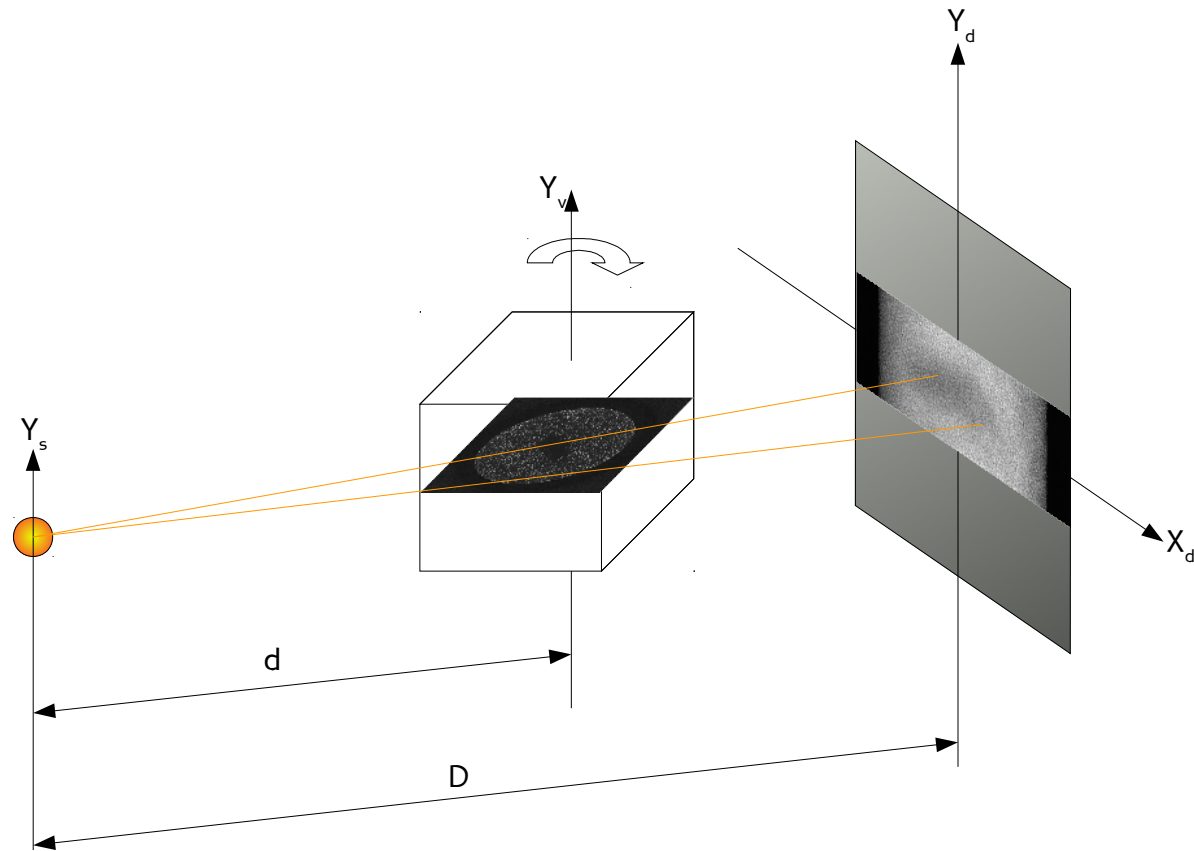
Betreuender Hochschullehrer: Prof. Dr. Wolfgang E. Nagel

Betreuer: Dr.-Ing. André Bieberle

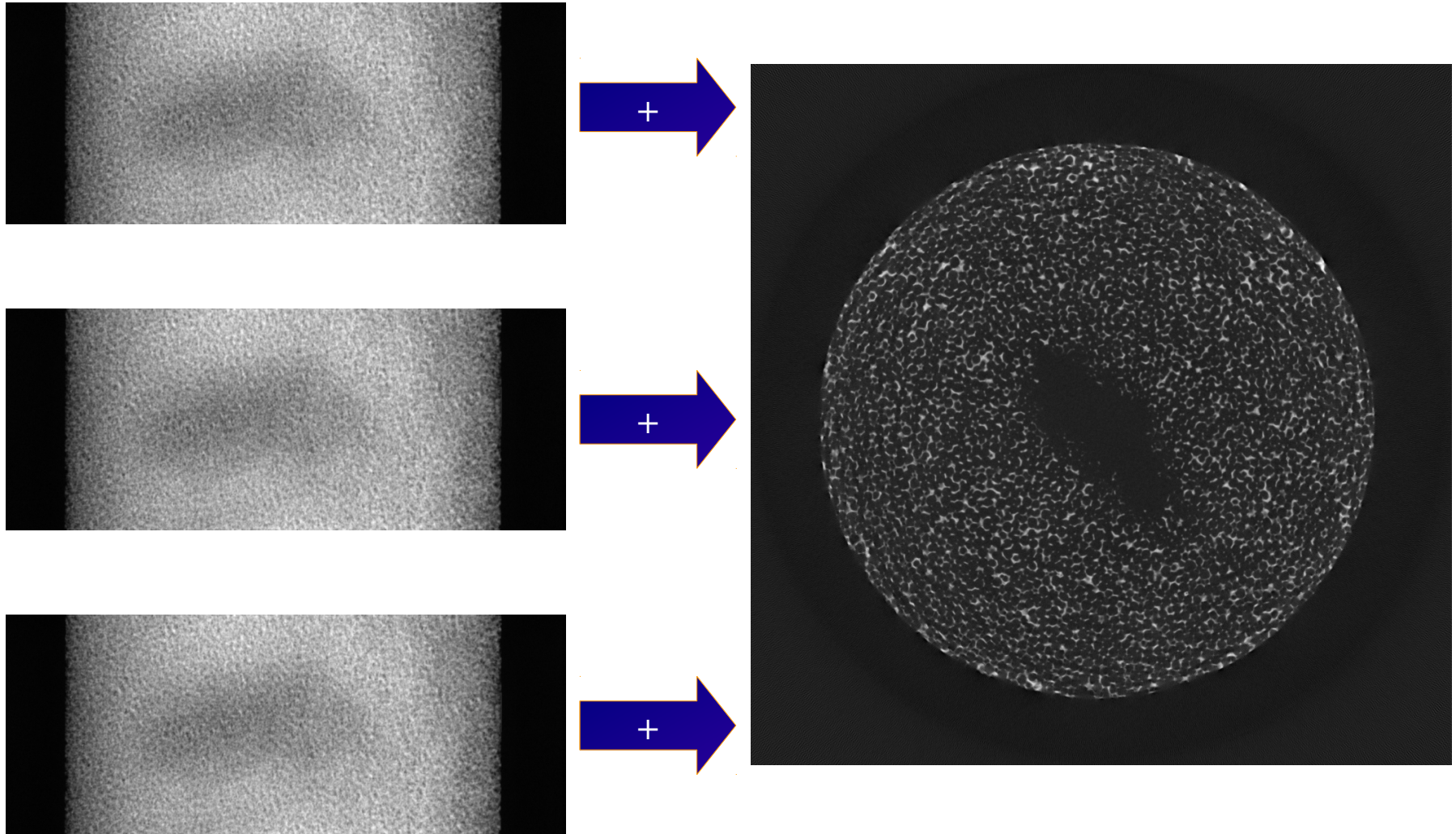
Ronny Brendel

Olaf Krzikalla

- Einleitung
  - Computertomographie
  - Der Feldkamp-Davis-Kress-Algorithmus
  - Intel® Xeon Phi™: Knights Landing
- Umsetzung
- Leistungsmessung
- Ausblick



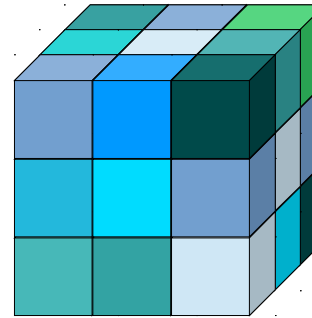
# Computertomographie



# Der Feldkamp-Davis-Kress-Algorithmus

---

- entwickelt 1984[1] von Feldkamp, Davis und Kress (FDK)
- gefilterte Rückprojektion
- für jedes Voxel unabhängig lösbar  
→ *embarrassingly parallel*



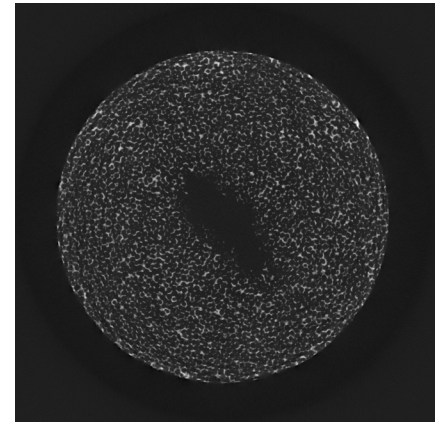
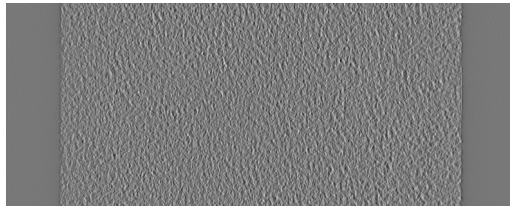
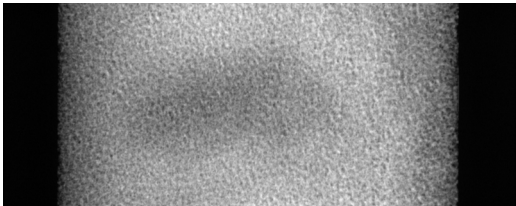
# Der Feldkamp-Davis-Kress-Algorithmus

---

- Xu et al.[2], 2004: vermutlich OpenGL
- Li et al.[3], 2005: FPGA
- Knaup et al.[4], 2006: PowerPC
- Scherl et al.[5], 2007: CUDA
- Domonkos et al.[6], 2009: OpenCL
- Hofmann et al.[7], 2012: Intel© Xeon Phi™: Knights Corner

- dritte Xeon-Phi™-Generation
- *Many-Integrated-Core*-Architektur (MIC)
  - 64 – 72 Kerne
  - 1,30 – 1,50 GHz
  - 16GiB Cache
- AVX-512-Befehlssatz
  - parallele Verarbeitung von acht 64bit-Zahlen / 16 32bit-Zahlen pro Instruktion und Kern







FDK-Pseudocode:

```
foreach (x, y, z) in volume {  
    p_x = // Volumenkoordinaten in Projektionskoordinaten  
    p_y = // umrechnen, unter Berücksichtigung des Projektionswinkels  
  
    det_val = projection(p_x, p_y);  
    (x, y, z) += det_val;  
}
```

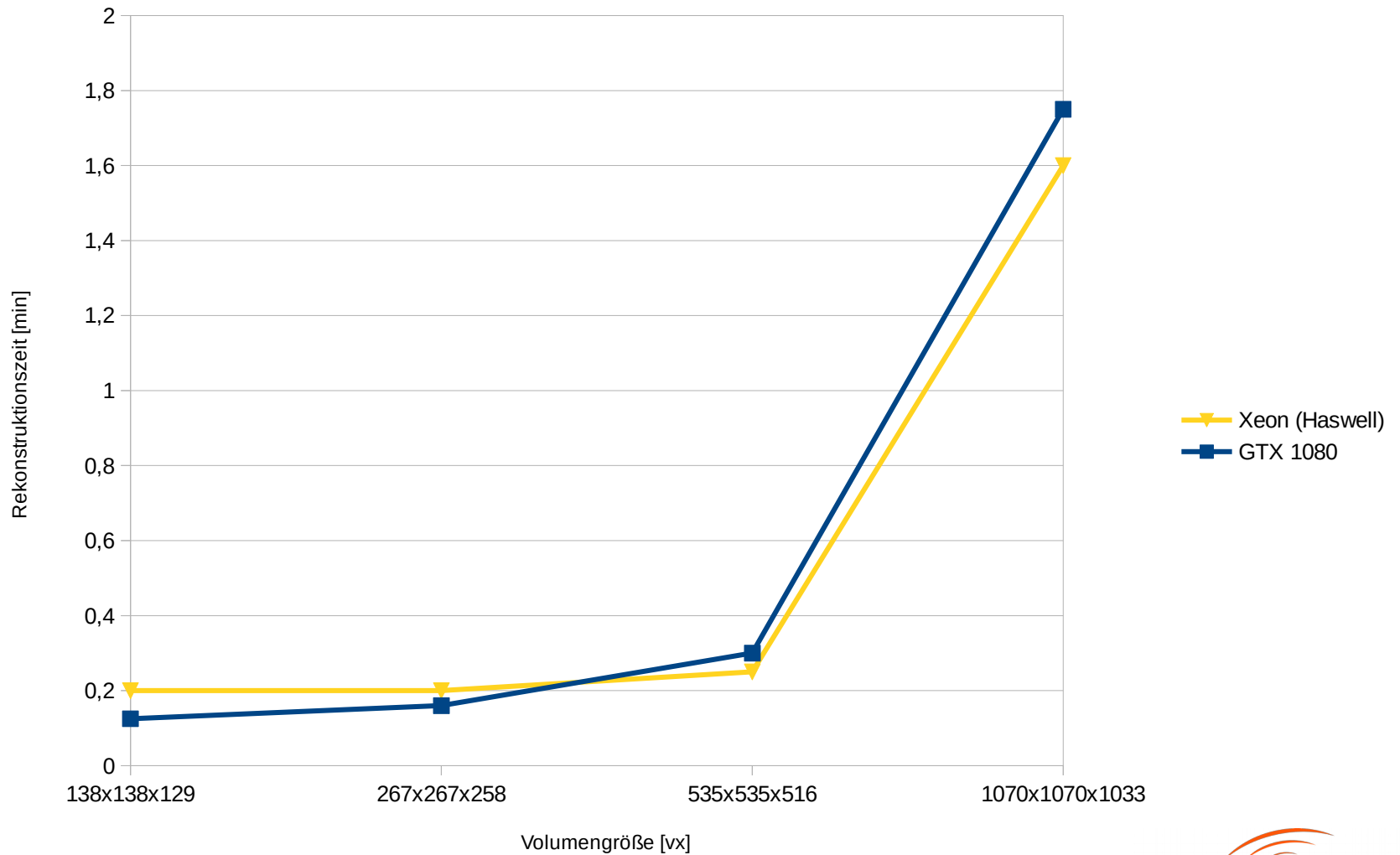
```
for(auto z = 0u; z < dim_z; ++z)
{
    for(auto y = 0u; y < dim_y; ++y)
    {
        for(auto x = 0u; x < dim_x; ++x)
        {
            /*
             * Rückprojektion...
             */
        }
    }
}
```

```
#pragma omp parallel for collapse(3)
for(auto z = 0u; z < dim_z; ++z)
{
    for(auto y = 0u; y < dim_y; ++y)
    {
        for(auto x = 0u; x < dim_x; ++x)
        {
            /*
             * Rückprojektion...
             */
        }
    }
}
```

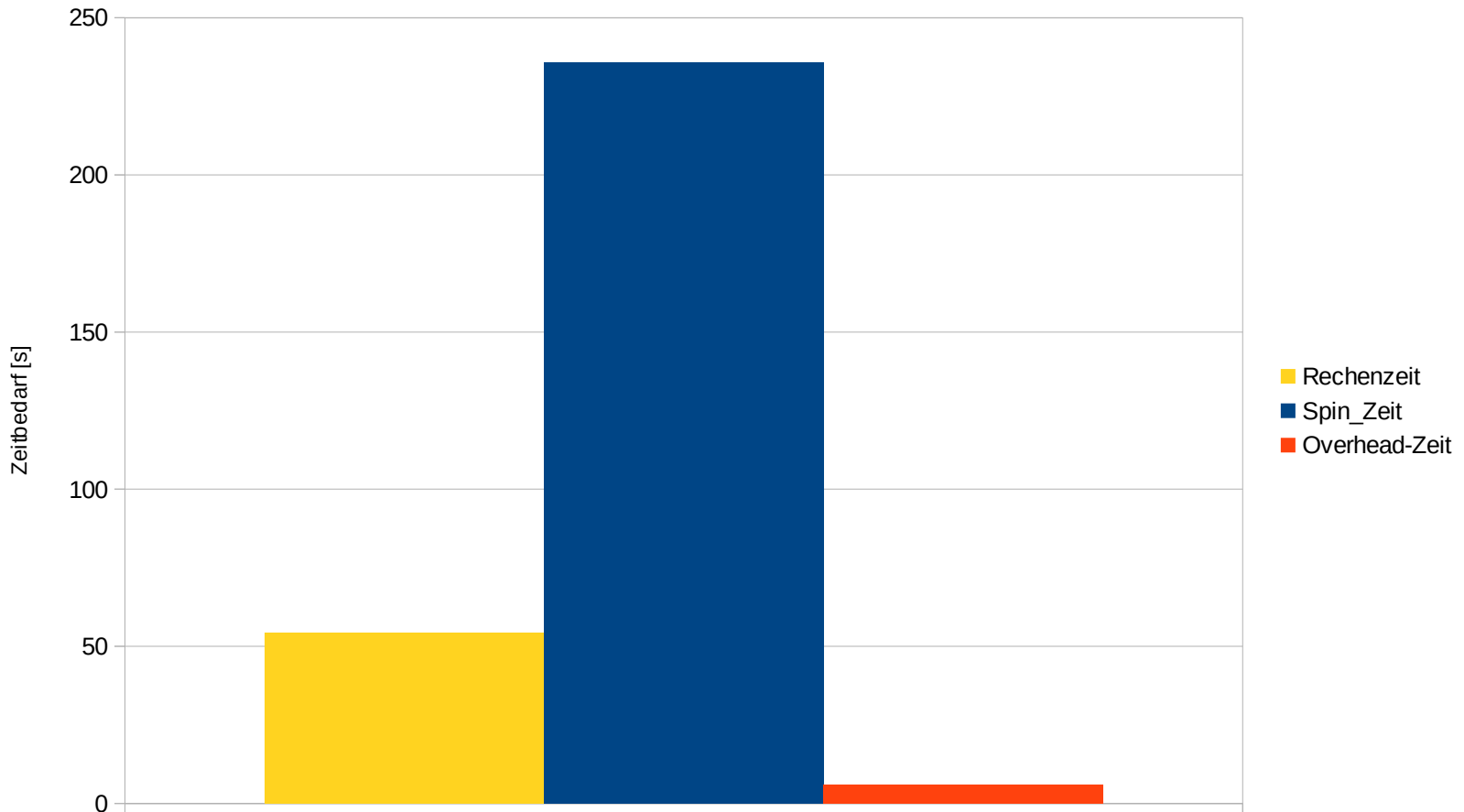
- Filterung: cuFFT → FFTW
- Beibehaltung der Pipelinestruktur
- Compiler: gcc 5.3.0

	Xeon E5/E7 v3 (Haswell)	GTX 1080
Taktfrequenz	2500 MHz	1607 MHz
(GPU-)RAM	64 GiB	8 GiB
(CUDA-)Kerne	24	2560
rekonstruiertes Volumen	eine Schicht	komplettes Volumen

# Leistungsmessung – Vergleich mit CUDA



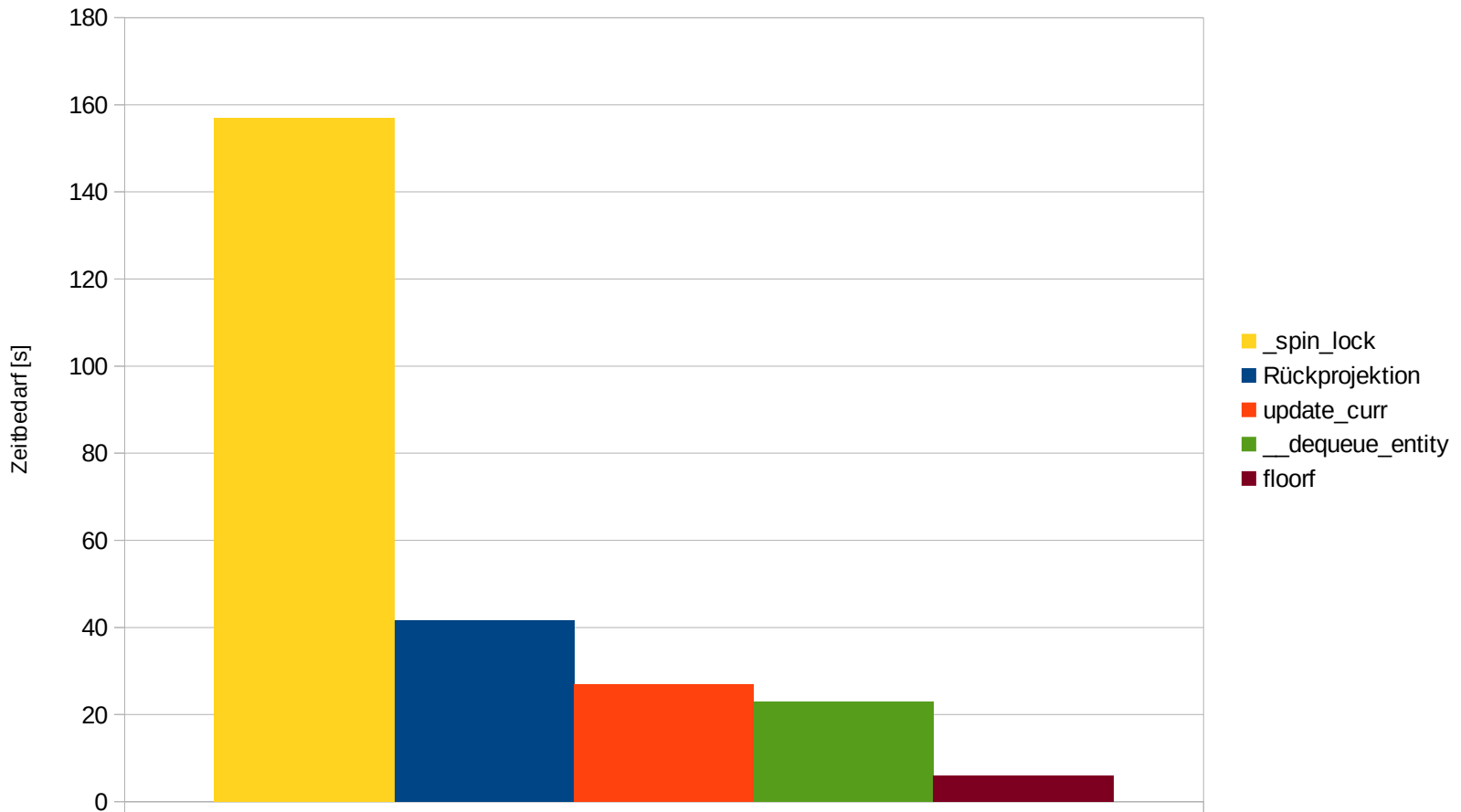
# Leistungsmessung - Rechenzeitaufteilung



Messsystem: Intel Xeon E5/E7 v3 (Haswell), 24 Kerne (Hyperthreading), 2,5GHz

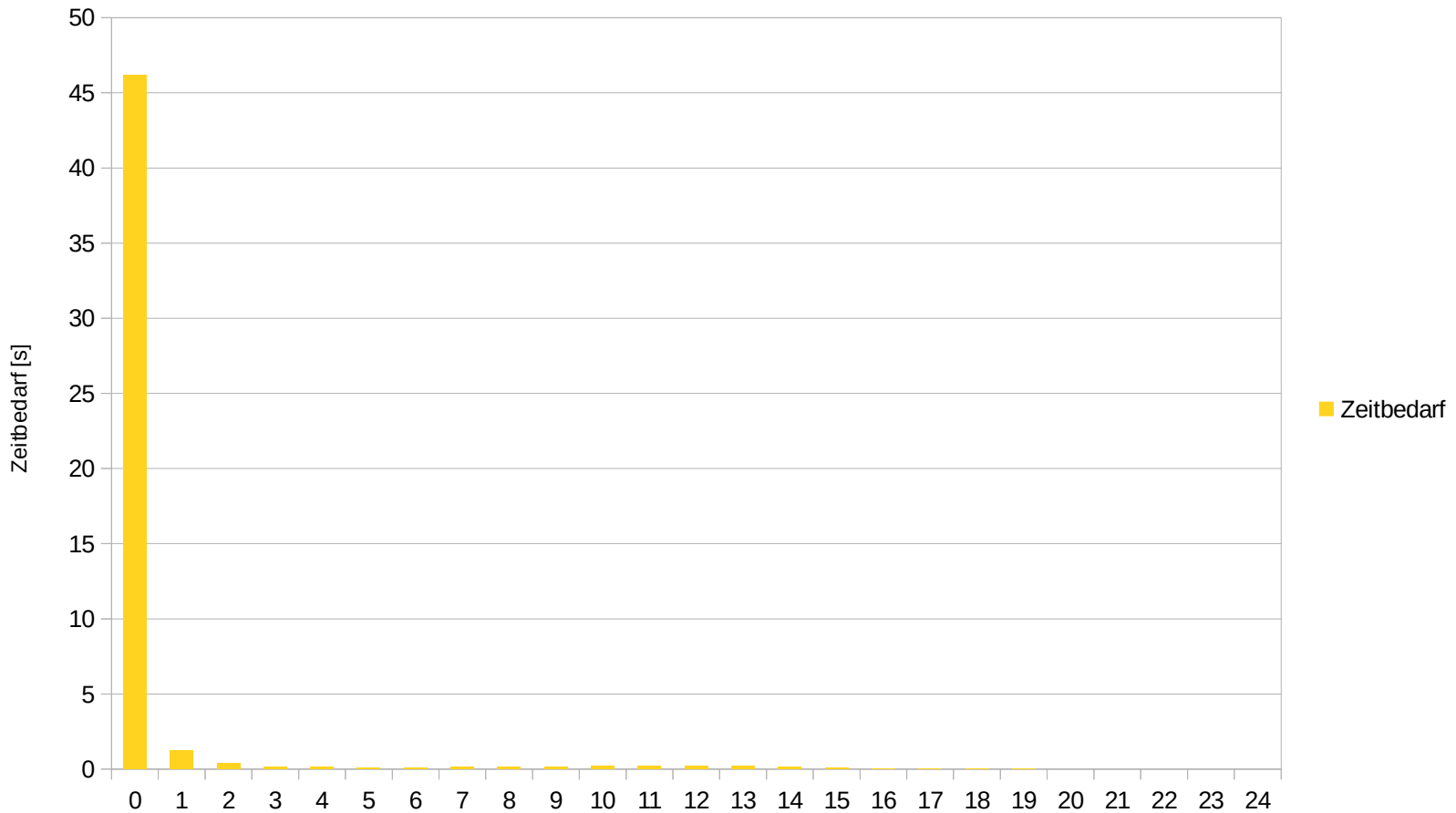


# Leistungsmessung – Top-5-Funktionen



Messsystem: Intel Xeon E5/E7 v3 (Haswell), 24 Kerne (Hyperthreading), 2,5GHz

# Leistungsmessung - Kernauslastung



Messsystem: Intel Xeon E5/E7 v3 (Haswell), 24 Kerne (Hyperthreading), 2,5GHz

- Verringerung der Spin-Zeit
- höhere Kernauslastung
- Compilervergleich (Intel, gcc, clang)
- Vektorisierung (automatisch / händisch)

---

# Vielen Dank!



- [1] L. A. Feldkamp, L. C. Davis & J. W. Kress: *Practical cone-beam algorithm*, Journal of the Optical Society of America A, Vol. 1, Issue 6, pp. 612 – 619, 1984
- [2] Fang Xu, Klaus Müller: *Ultra-Fast 3D Filtered Backprojection on Commodity Graphics Hardware*, IEEE International Symposium on Biomedical Imaging, April 2004
- [3] Jianchun Li, Christos Papachristou, Raj Shekhar: *An FPGA-Based Computing Platform for Real-Time 3D Medical Imaging and its Application to Cone-Beam CT Reconstruction*, Journal of Imaging Science and Technology, Vol 49 Nr. 3, S. 237 – 245(9), Juli 2005
- [4] Michael Knaup, Sven Steckmann, Olivier Bockenbach, Marc Kachelrieß: *Tomographic image reconstruction using the cell broadband engine (CBE) general purpose hardware*, Proc. SPIE 6498, Computational Imaging V, 64980P, Februar 2007
- [5] Holger Scherl, Benjamin Keck, Markus Kowarschik, Joachim Hornegger: *Fast GPU-Based CT Reconstruction using the Common Unified Device Architecture (CUDA)*, 2007 IEEE Nuclear Science Symposium Conference Record, S. 4464 - 4466, Oktober 2007
- [6] Balázs Domonkos, Gábor Jakab: *A programming model for GPU-based parallel Computing with scalability and abstraction*, SCCG '09 Proceedings of the 25th Spring Conference on Computer Graphics, S. 103 – 111, April 2009

- [7] Johannes Hofmann, Jan Treibig, Georg Hager, Gerhard Wellein: *Performance Engineering for a Medical Imaging Application on the Intel Xeon Phi Accelerator*, Architecture of Computing Systems (ARCS), 2014 Workshop Proceedings, Februar 2014