

Literaturverzeichnis

- 1) **Bengel, G.; et al.:** Masterkurs Parallele und Verteilte Systeme. Vieweg + Teubner, Wiesbaden, 2008.
- 2) **Breshears, C.:** The Art of Concurrency - A Thread Monkey's Guide to Writing Parallel Applications. O'Reilly, Beijing, 2009.
<http://oreilly.com/catalog/9780596521547>
- 3) **Cook, S.:** CUDA Programming - A Developer's Guide to Parallel Computing with GPUs. Morgan Kaufmann, Amsterdam, 2013.
- 4) **Farber, R.:** CUDA Application Design and Development. Morgan Kaufmann, Amsterdam, 2012.
- 5) **Fernando, R.:** GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics. Addison-Wesley, Boston, MA, 2004.
http://http.developer.nvidia.com/GPUGems/gpugems_part01.html
- 6) **Fernando, R.; Kilgard, M.J.:** The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics. Addison-Wesley, Boston, MA, 2003.
http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter01.html
- 7) **Gaster, B.R.; et al.:** Heterogeneous Computing with OpenCL. 2nd edition, Morgan Kaufmann, Amsterdam, 2013.
<http://store.elsevier.com/product.jsp?isbn=9780124058941>
- 8) **Grama, A.; et al.:** Introduction to Parallel Computing. Pearson Education, Harlow, England, 2nd edition, 2003.
- 9) **Herlihy, M.; Shavit, N.:** The Art of Multiprocessor Programming. Elsevier, Amsterdam, 2008.
<http://www.elsevierdirect.com/product.jsp?isbn=9780123705914>
<http://www.elsevierdirect.com/companion.jsp?ISBN=9780123705914>
- 10) **Hwu, W.W. (ed.):** GPU Computing Gems: Emerald Edition. Elsevier, Amsterdam, 2011.
- 11) **Kirk, D.B.; Hwu, W.W.:** Programming Massively Parallel Processors - A Hands-on Approach. Elsevier, Amsterdam, 2nd edition, 2012.
<http://www.elsevierdirect.com/product.jsp?isbn=9780123814722>
<http://booksite.elsevier.com/9780124159921/>
- 12) **Munshi, A.; et al.:** OpenCL Programming Guide. Addison-Wesley, Upper Saddle River, NJ, 2012.
- 13) **Nguyen, H.:** GPU Gems 3: Programming Techniques for High-Performance Graphics and General-Purpose Computation. Addison-Wesley, Boston, MA, 2007.
http://http.developer.nvidia.com/GPUGems3/gpugems3_part01.html

- 14) **Pharr, M.:** GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation. Addison-Wesley, Boston, MA, 2005.
http://http.developer.nvidia.com/GPUGems2/gpugems2_part01.html
- 15) **Pacheco, P. S.:** Parallel Programming with MPI. Morgan Kaufmann, San Francisco, CA, 1997.
<http://www.cs.usfca.edu/mpi>
- 16) **Pacheco, P. S.:** An Introduction to Parallel Programming. Morgan Kaufmann, Burlington, MA, 2011.
<http://www.mkp.com/pacheco>
- 17) **Quinn, M. J.:** Parallel Programming in C with MPI and OpenMP. McGraw-Hill, New York, 2004.
<http://fac-staff.seattleu.edu/quinnm/web>
- 18) **Sanders, J.; Kandrot, E.:** CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley, Upper Saddle River, NJ, 2011.
- 19) **Wilt, N.:** The CUDA Handbook - A Comprehensive Guide to GPU Programming. Addison-Wesley, Upper Saddle River, NJ, 2013.
<http://www.cudahandbook.com>

Informationen im *World Wide Web*:

- 1) **AMD Accelerated Parallel Processing (APP) SDK (formerly ATI Stream):**
<http://developer.amd.com/tools-and-sdks/opencl-zone/amd-accelerated-parallel-processing-app-sdk/>
- 2) **AMD Accelerated Parallel Processing (APP) SDK OpenCL Programming Guide:**
http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2013/07/AMD_Accelerated_Parallel_Processing_OpenCL_Programming_Guide-rev-2.7.pdf
- 3) **AMD Developer Central:** Tools, Resources, ...
<http://developer.amd.com>
- 4) **Brown Deer Technology:** OpenCL Tutorial: N-Body Simulation.
http://www.browndertechnology.com/docs/BDT_OpenCL_Tutorial_NBody-rev3.html
- 5) **FreeBookCentre.Net:**
<http://www.freebookcentre.net>
- 6) **General-Purpose Computation on Graphics Hardware:**
<http://gpgpu.org>
<http://gpgpu.org/tag/ati-stream>
<http://gpgpu.org/tag/opencl>

- 7) **GNU Scientific Library (GSL):**
<http://www.gnu.org/software/gsl>
http://www.gnu.org/software/gsl/manual/html_node
- 8) **GPU Resources:**
<http://cseweb.ucsd.edu/~baden/Doc/gpu.html>
- 9) **GPUSort: High Performance Sorting using Graphics Processors:**
<http://gamma.cs.unc.edu/GPUSORT>
<http://gamma.cs.unc.edu/GPUSORT/results.html>
- 10) **Intel OpenCL SDK:**
<https://software.intel.com/en-us/intel-opencl>
- 11) **Intel Xeon E5000er Prozessoren (Überblick, Spezifikationen, Technische Dokumente):**
<http://www.intel.de/content/www/de/de/processors/xeon/xeon-processor-5000-sequence.html>
- 12) **Mathematica GPU Computing:**
<http://www.wolfram.com/solutions/hpc>
<http://reference.wolfram.com/mathematica/ParallelTools/tutorial/Overview.html>
<http://www.nvidia.de/object/cuda-programming-mathematica-de.html>
- 13) **MATLAB GPU Computing:**
http://www.mathworks.de/products/parallel-computing/?s_cid=HP_FP_ML_parallel
<http://www.mathworks.de/discovery/matlab-gpu.html>
<http://developer.nvidia.com/cuda-tools-ecosystem#Numeric>
- 14) **Microsoft Parallel Computing Developer Center:**
<http://msdn.microsoft.com/de-de/concurrency/default>
- 15) **MIT Open Courseware: Applied Parallel Computing.**
<http://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-950-parallel-programming-for-multicore-machines-using-openmp-and-mpi-january-iap-2010/>
<http://ocw.mit.edu/courses/mathematics/18-337j-parallel-computing-fall-2011/index.htm>
<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-multicore-programming-primer-january-iap-2007/>
<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-895-theory-of-parallel-systems-sma-5509-fall-2003/>
- 16) **MPI standard:** The Message Passing Interface Standard.
<http://www.mpi-forum.org/docs/docs.html>,
<http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>
- 17) **Numerik-Bibliotheken:** BLAS/PBLAS/CUBLAS, LAPACK/PLAPACK/CULA/MAGMA, CUDPP, ScaLAPACK.
 - BLAS/PBLAS: (Parallel) Basic Linear Algebra Subprograms.
<http://www.netlib.org/blas/index.html>
http://www.netlib.org/scalapack/pblas_qref.html

- CUBLAS: CUDA/GPU accelerated BLAS.
<http://docs.nvidia.com/cuda/cublas/index.html>
 - LAPACK/PLAPACK: (Parallel) Linear Algebra Package.
<http://www.netlib.org/lapack/index.html>
<http://www.cs.utexas.edu/~plapack>
 - CULA: CUDA/GPU accelerated LAPACK.
<http://www.culatools.com>
 - MAGMA: Matrix Algebra on GPU and Multicore Architectures.
<http://icl.cs.utk.edu/magma>
 - CUDPP: CUDA Data Parallel Primitives Library.
<http://code.google.com/p/cudpp>
 - ScaLAPACK: Scalable LAPACK.
<http://www.netlib.org/scalapack/index.html>
- 18) **Nvidia CUDA:**
<http://docs.nvidia.com/cuda/index.html>
<https://developer.nvidia.com/cuda-toolkit>
http://www.nvidia.com/object/cuda_home_new.html
- 19) **Nvidia Developer Web Site:**
<https://developer.nvidia.com>
- 20) **Nvidia GPU Computing Webinars:**
<https://developer.nvidia.com/gpu-computing-webinars>
- 21) **Nvidia GPU Programming Guide:**
<https://developer.nvidia.com/nvidia-gpu-programming-guide>
http://developer.download.nvidia.com/GPU_Programming_Guide/GPU_Programming_Guide_G80.pdf
- 22) **Nvidia Tesla Cxxxx (Overview, Specifications, Drivers & Downloads, ...):**
<http://www.nvidia.com/object/tesla-workstations.html>
<http://www.nvidia.com/object/tesla-supercomputing-solutions.html>
- 23) **OpenACC:**
<http://www.openacc.org>
<https://developer.nvidia.com/openacc>
<https://developer.nvidia.com/content/openacc-example-part-1>
<https://developer.nvidia.com/content/openacc-example-part-2>
- 24) **OpenCL:**
<http://www.khronos.org/opencvl>
<http://developer.nvidia.com/opencvl>
<http://developer.amd.com/tools-and-sdks/opencvl-zone/>
- 25) **OpenCL Programming Guide:**
http://www.nvidia.com/content/cudazone/download/OpenCL/NVIDIA_OpenCL_ProgrammingGuide.pdf
- 26) **OpenCL Quick Reference Card:**
<http://www.khronos.org/files/opencvl20-quick-reference-card.pdf>

- 27) **OpenCL Spezifikation:**
<http://www.khronos.org/registry/cl>
- 28) **OpenGL and OpenCL Debugger:**
<http://www.gremedy.com>
- 29) **Open MPI:** Open Source High Performance Computing.
<http://www.open-mpi.org>
- 30) **OpenMP.org:** OpenMP Application Program Interface. Version 4.5, November 2015.
<http://openmp.org>
<http://openmp.org/wp/openmp-specifications>
<http://www.openmp.org/mp-documents/openmp-4.5.pdf>
- 31) **Wikipedia:**
ATI-Stream: <http://de.wikipedia.org/wiki/ATI-Stream>
CUDA: <http://de.wikipedia.org/wiki/CUDA>
GPGPU: <http://de.wikipedia.org/wiki/GPGPU>
Grafikprozessor: <http://de.wikipedia.org/wiki/Grafikprozessor>
OpenACC: <http://en.wikipedia.org/wiki/OpenACC>
OpenCL: <http://de.wikipedia.org/wiki/OpenCL>

Software im *World Wide Web*:

- 1) **Accelerated Parallel Processing (APP) SDK:**
<http://developer.amd.com/tools/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk>
- 2) **Intel C/C++ Compiler:** Kann für Linux für nicht kommerziellen Einsatz kostenfrei lizenziert werden.
<http://software.intel.com/en-us/articles/non-commercial-software-development/>
<https://software.intel.com/en-us/intel-education-offerings/>
(Nachdem Sie die Lizenzbedingungen akzeptiert haben, können Sie unter anderem den "Intel C++ Compiler for Linux" herunterladen. Bevor der Compiler tatsächlich heruntergeladen werden kann, müssen Sie sich noch mit Ihrer *E-Mail*-Adresse registrieren. An diese Adresse wird dann eine Lizenzdatei geschickt, die Sie in einem beliebigen Verzeichnis speichern können. Den Pfadnamen zu dem Verzeichnis speichern Sie in der Umgebungsvariablen "INTEL_LICENSE_FILE", z. B. mit "setenv INTEL_LICENSE_FILE /home/fd1026/intel_lic".)
- 3) **Intel SDK for OpenCL Applications:**
<https://software.intel.com/en-us/intel-openccl>
- 4) **Nvidia CUDA Download:**
<https://developer.nvidia.com/cuda-downloads>
- 5) **NVIDIA CUDA SDK Code Samples:**
<http://docs.nvidia.com/cuda/cuda-samples/index.html>