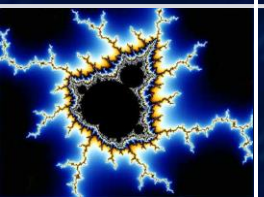
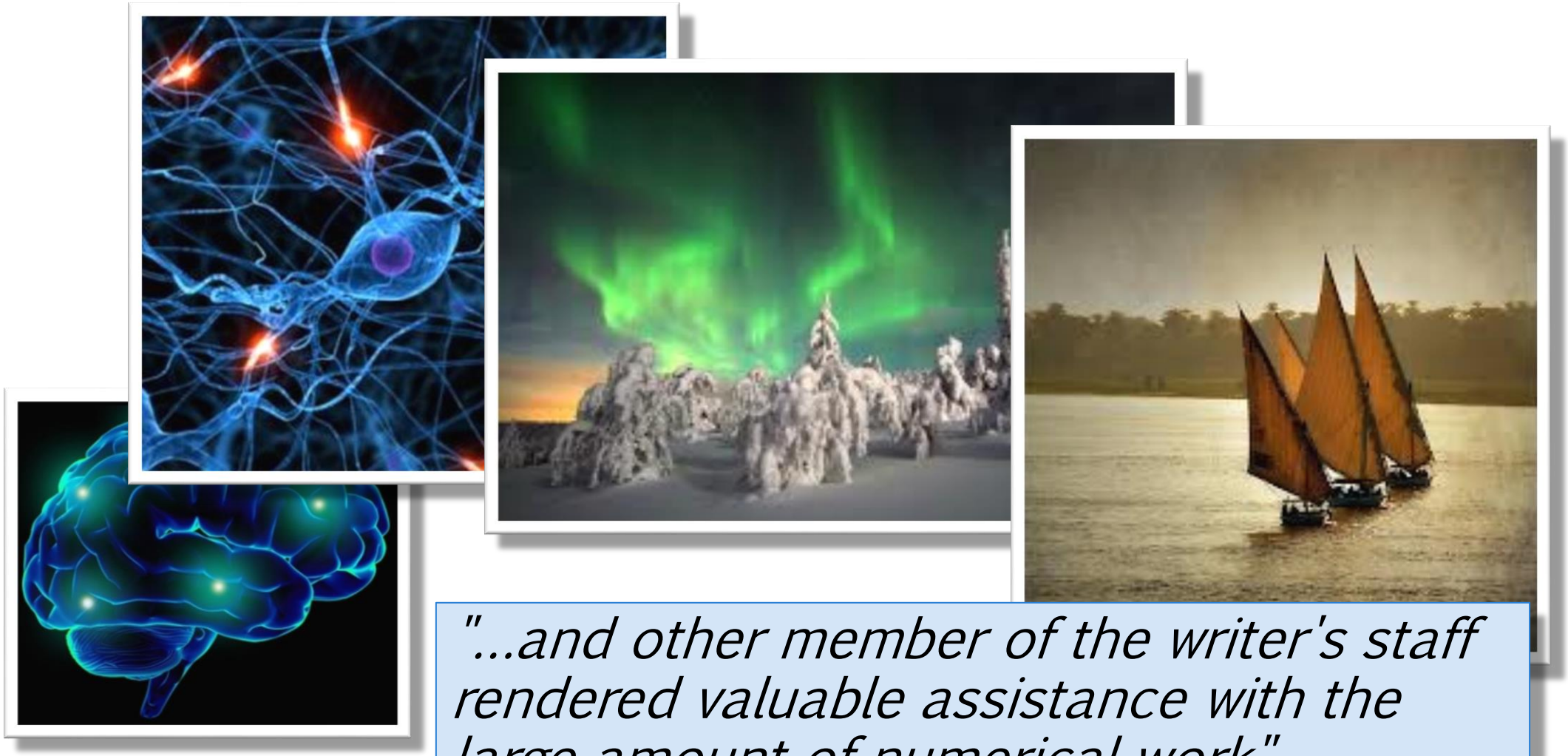
Thinking parallel: Multi-cores, virtual elasticity, and the application programmer



Geir Horn

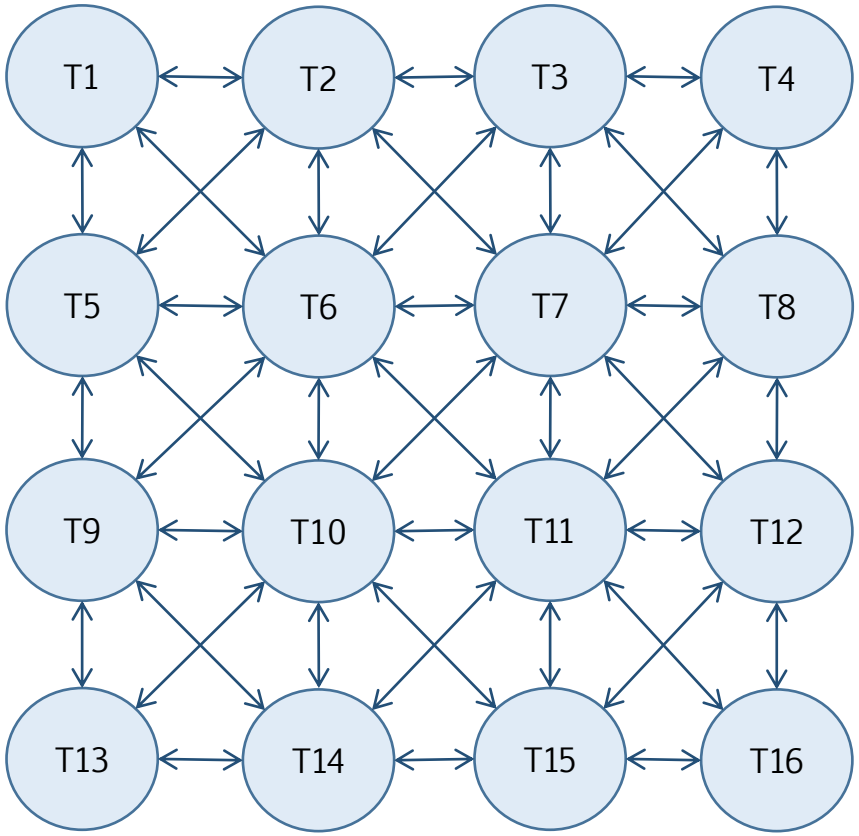


"...and other member of the writer's staff rendered valuable assistance with the large amount of numerical work"

- H. E. Hurst[2]

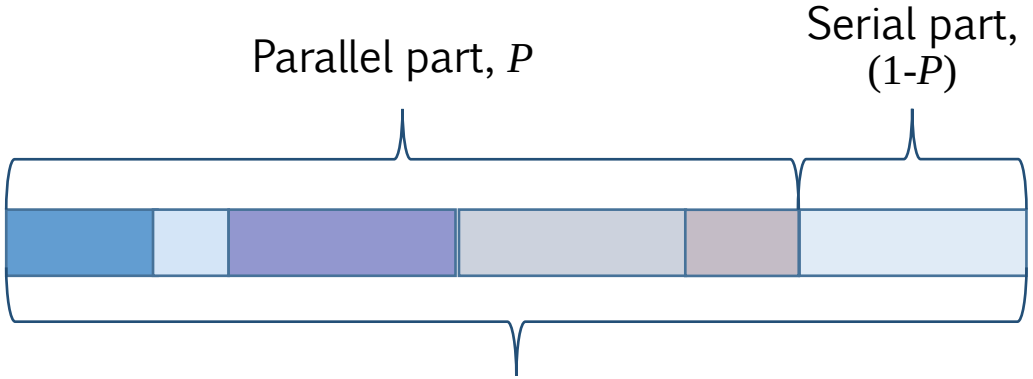
[1] Suzana Herculano-Houzel, 'The human brain in numbers: a linearly scaled-up primate brain', *Front. Hum. Neurosci.*, vol. 3, p. Article 31, Nov. 2009.
[2] Harold Edwin Hurst, 'Long-term storage capacity of reservoirs', *Transactions of the American Society of Civil Engineers*, pp. 770–799, 1951.

The basics



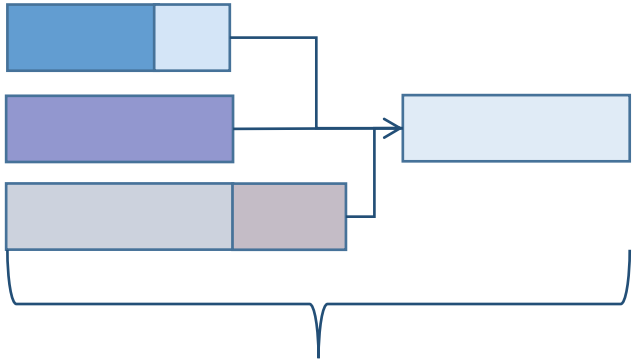
Task Interaction Graph (TIG)

Sequential scheduling (one processor)



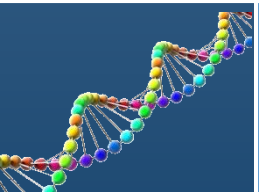
Makespan = $T_{sequential}$

Parallel scheduling (three processors)



$$\text{Makespan} = T_{parallel} + T_{serial}$$

$$\text{Speedup} = T_{sequential} \Big/ \left(T_{parallel} + T_{serial} \right)$$



Speedup versus Scalability

- Ideal speedup = number of processors = $|\mathrm{N}|$
- Amdahl's law[3]:

$$\text{Speedup} = \frac{1}{(1 - P) + \frac{P}{|\mathrm{N}|}}$$

- Max speedup as $|\mathrm{N}| \to \infty$ is

$$\frac{1}{1 - P}$$

- Example: With $P = 90\%$ the max speedup is 10

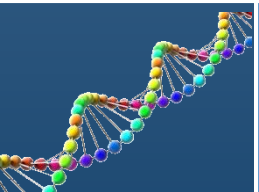
- Alternatively: Keep run time fixed, but increase problem size with the number of processors
- Hypothetical sequential run time

$$T_{\text{sequential}} = T_{\text{serial}} + |\mathrm{N}| \times T_{\text{parallel}}$$

- Gustafson-Barsis' law[4]: the scaled speedup is

$$\begin{aligned} \mathrm{SS} &= T_{\text{sequential}} \Big/ \left(T_{\text{parallel}} + T_{\text{serial}}\right) \\ &= \left(T_{\text{serial}} + |\mathrm{N}| \times T_{\text{parallel}}\right) \Big/ \left(T_{\text{parallel}} + T_{\text{serial}}\right) \\ &= |\mathrm{N}| - \alpha \left(|\mathrm{N}| - 1\right) \end{aligned}$$

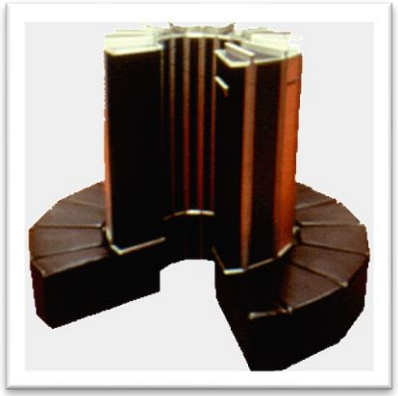
with $\alpha = T_{\text{serial}} \Big/ \left(T_{\text{parallel}} + T_{\text{serial}}\right)$



[3] Gene M. Amdahl, 'Validity of the single processor approach to achieving large scale computing capabilities', in *Proceedings of the AFIPS spring joint computer conference*, Conference location: Atlantic City, New Jersey, USA, 1967, pp. 483–485.
[4] John L. Gustafson, 'Reevaluating Amdahl's law', *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, May 1988.

Parallel *digital* computers

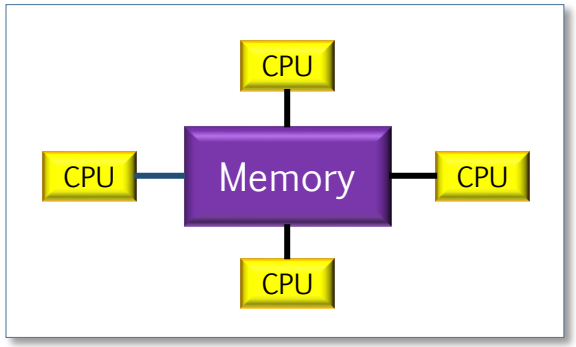
Burroughs' D825
4 processors
(1962)



Cray-1 (1976)
Cray X-MP (1980)

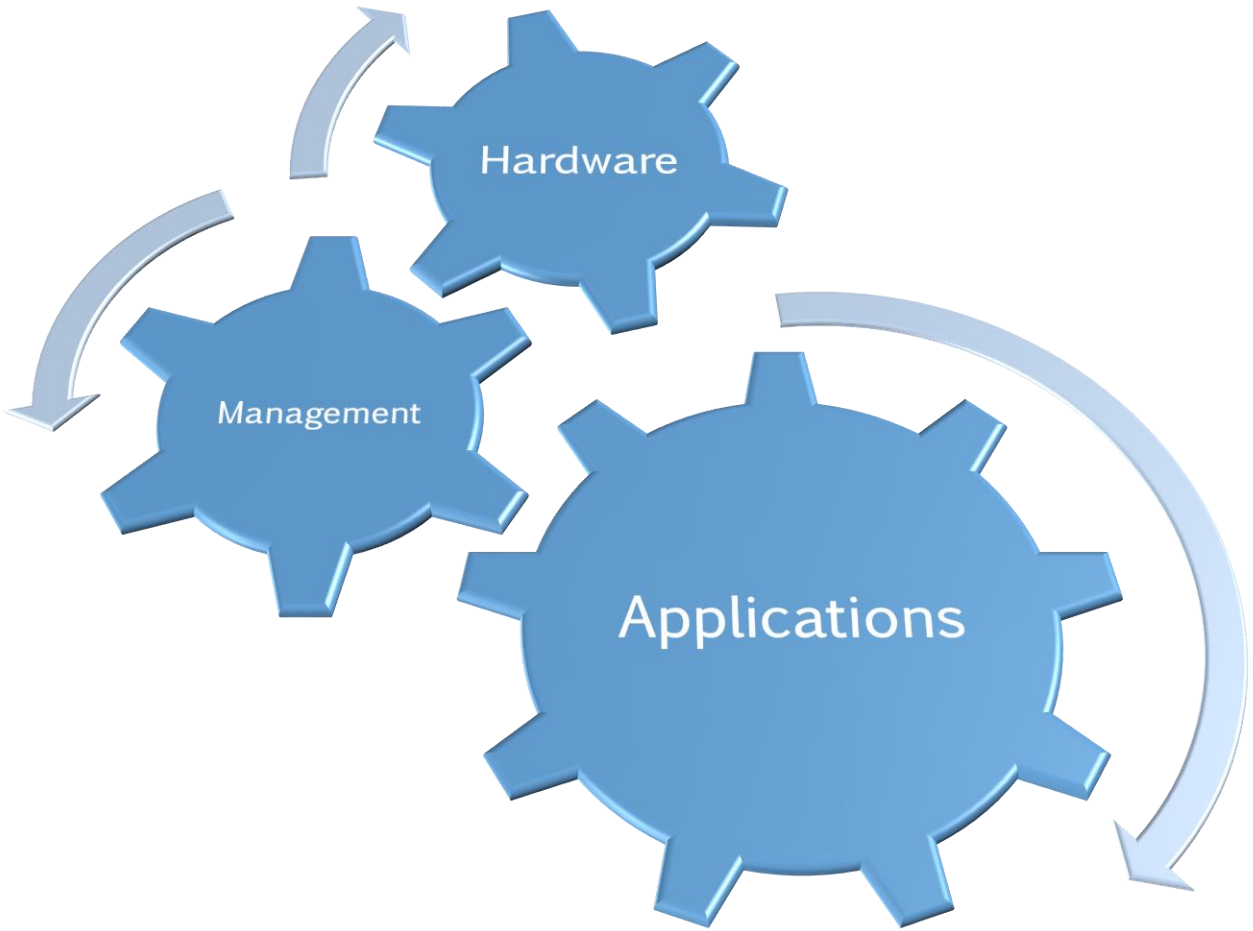
Distributed computing

- Massive Parallel Processing = Dedicated interconnect
- Cluster Computing = Ethernet + TCP/IP
- GRID computing = Internet
- Cloud = everything virtualised



Shared memory

Parallel computing†

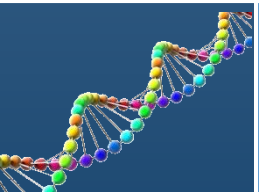


Hardware
- The execution platform

Management software
- Operating System (OS)
- Platform control
- Management middleware
- Development support

Applications
- The essential value

† High Performance Computing, Many Task Computing, maybe High Throughput Computing, not Dedicated Computing



More beyond Moore

Gordon E. Moore's law[5]:

"...the number of transistors on a chip will double every 18 months"

Mostly process scaling with 0.7 every 24 months from 0.8 μm in 1992 until ~2000

- Physical distance problems with scaling
- Voltage supply scaling and increased leaks
- Clock frequency: maximum around 5GHz
- Power density wall and heat
- Design complexity: exponential design team growth

[5] Gordon E. Moore, 'Cramming More Components Onto Integrated Circuits', *Electronics*, pp. 114–117, Apr. 1965.
[6] Jeff Parkhurst, John Darringer, and Bill Grundmann, 'From single core to multi-core: preparing for a new exponential', in *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design (ICCAD '06)*, Conference location: San Jose, California, USA, 2006, pp. 67–72.

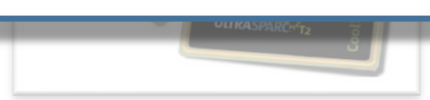
...on heterogeneous cores

Intel & AMD x86
dual-cores
(2005)

"Assuming that this trend will follow Moore's Law scaling, mainstream systems will contain over 10 processing cores by the end of the decade, yielding unprecedented theoretical peak performance"

Justin Rattner
Vice president and chief technology officer, Intel
(2005)[10]

simultaneous
multithreading
(2004)[8]



SUN UltraSparc T2
8 cores, 64 threads
(2007)[9]

[7] J. M. Tendler *et al*, 'POWER4 system microarchitecture', *IBM Journal of Research and Development*, vol. 46, no. 1, pp. 5–25, Jan. 2002.
[8] B. Sinharoy *et al*, 'POWER5 system microarchitecture', *IBM Journal of Research and Development*, vol. 49, no. 4.5, pp. 505–521, Jul. 2005
[9] Umesh Gajanan Nawathe *et al*, 'An 8-Core 64-Thread 64b Power-Efficient SPARC SoC', in *Digest of Technical Papers from the IEEE International Solid-State Circuits Conference (ISSCC 2007)*, Conference location: San Francisco, CA, USA, 2007, pp. 5.7: 108–110.
[10] Justin Rattner, 'Multi-core to the masses', in *Proceedings of the 14th International Conference on Parallel Architectures and Compilation Techniques (PACT 2005)*, Conference location: Saint Louis, Missouri, USA, 2005, p. 3.

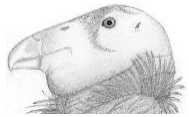
Hardware: The future

- Dedicated *accelerators*† or proven FPGA templates or design patterns
- Proven library of modules with predictable performance[15]
  - Parameterised interconnects
  - Standard adapters and interfaces
  - Reusable building blocks
- Asymmetric speedup ≥ symmetric speedup[16]
- "*Software will take a more prominent role in the multi-core era*"[6]
- "*...the programming model for heterogeneous architectures is much more complicated*"[17]

† "*Technology based on "manycore" will employ 100s to 1000s of CPU cores per chip (by2011)*"[14]

[14] John Shalf, 'The new landscape of parallel computer architecture', *Journal of Physics: Conference Series*, vol. 78, p. 012066, Jul. 2007.
[15] John A Darringer, 'Multi-core design automation challenges', in *Proceedings of the 44th annual Design Automation Conference (DAC '07)*, San Diego, California, USA, 2007, pp. 760–764.
[16] Mark D. Hill and Michael R. Marty, 'Amdahl's Law in the Multicore Era', *Computer*, vol. 41, no. 7, pp. 33–38, Jul. 2008.
[17] Geoffrey Blake *et al.* 'A survey of multicore processors', *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 26–37, Nov. 2009.

Operating Systems

- "No current OS is really multithreaded"[a]
- New approaches to operating systems
  - XtreemOS[b] =  + 
  - S(o)SS = Service Oriented Operating Systems[18]
  - Tessllation[19]
  - FUSE: OS support for easy HW accelerator integration[20]
- Native hypervisors + microkernels
- Real-time OS[d]

a Lutz Schubert, Universität Ulm

b http://www.xtreemos.eu/ and http://research.cs.wisc.edu/condor/

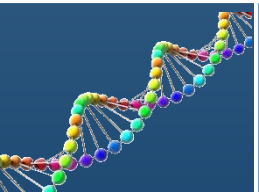
c http://www.soos-project.eu/

d QNX Neutrino RTOS (http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html) or INTEGRITY (http://www.ghs.com/products/rtos/integrity.html)

[18] Lutz Schubert *et al.* 'Service-oriented operating systems: future workspaces', *IEEE Wireless Communications*, vol. 16, no. 3, pp. 42–50, Jun. 2009.
[19] Krste Asanovic *et al.* 'A view of the parallel computing landscape', *Communications of the ACM*, vol. 52, no. 10, pp. 56–67, Oct. 2009.
[20] Aws Ismail and Lesley Shannon, 'FUSE: Front-End User Framework for O/S Abstraction of Hardware Accelerators', in *Proceedings of the 19th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM 2011)*, Salt Lake City, Utha, USA, 2011, pp. 170–177.

High Throughput Mass Market

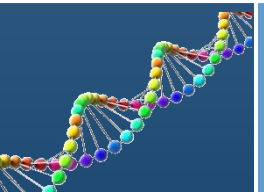
- Desktop
  - Legacy software "does the job"
  - Current software developed for single-core
  - Per-application performance is important (only) if the load consists of only a few applications or if there are performance-critical applications[33]
- Servers
  - Database and web servers are designed for high throughput
  - Idle time can be masked by multi-threading[33]
- Large parallel application part = more cores, otherwise more complex cores[16]



[30] Jacques A. Pienaar *et al.* 'MDR: performance model driven runtime for heterogeneous parallel platforms', in *Proceedings of the international conference on Supercomputing (ICS '11)*, Tucson, Arizona, USA, 2011, pp. 225–234.
[31] Anders Logg *et al.* 'Past and Future Perspectives on Scientific Software', in *Simula Research Laboratory*, vol. Part II, Aslak Tveito *et al.* (Eds.) Springer, 2010, pp. 321–362.
[32] Michela Becchi *et al.* 'Enabling Legacy Applications on Heterogeneous Platforms', in *Proceedings of the 2nd USENIX Workshop on Hot Topics in Parallelism (HotPar 2010)*, Berkeley, CA, USA, 2010
[33] Angela C. Sodan *et al.* 'Parallelism via Multithreaded and Multicore CPUs', *Computer*, vol. 43, no. 3, pp. 24–32, Mar. 2010.

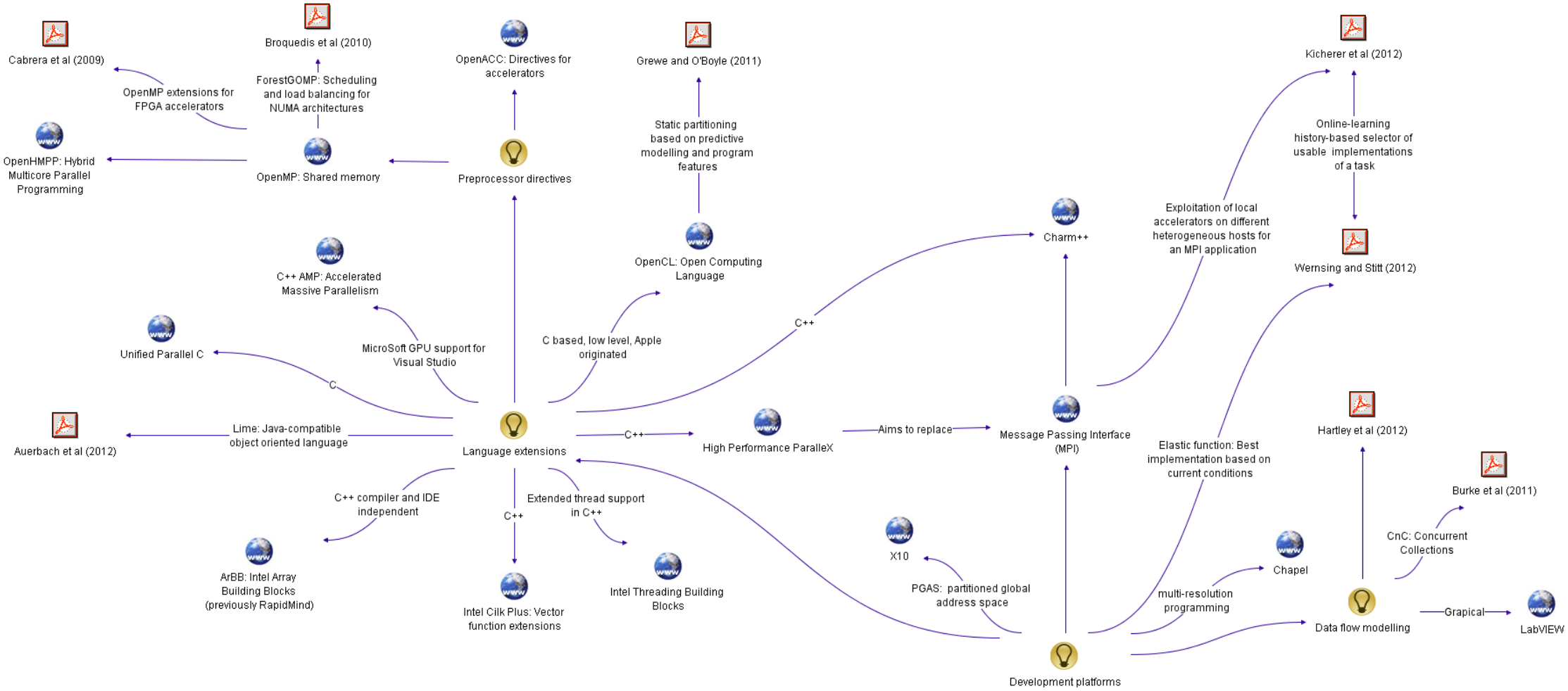
The future

- Pin-limits: no technology in sight![33]
- Three scenarios[35]
  - "Drop the ball" ⇒ Cloud computing
  - "Niche markets" ⇒ Multimedia, gaming…
  - "Scalable, dependable, software development"[13]
- Urgently needed: *"Parallel computing for all"* ™
  - Standardised, industry accepted development platforms
  - Education of programmers on these platforms
  - Requirements engineering for parallelism



[35] David Patterson, 'The trouble with multi-core', *IEEE Spectrum*, vol. 47, no. 7, pp. 28–32, 53–54, Jul. 2010.

Development support



Thinking parallel:
Multi-cores, virtual elasticity,
and the application programmer

http://www.open-mpi.org/
http://openmp.org
http://www.openhmpp.org
http://www.openacc-standard.org/
http://www.khronos.org/opencl/

http://upc.lbl.gov/
http://threadingbuildingblocks.org/
http://software.intel.com/en-us/articles/intel-cilk-plus/
http://software.intel.com/en-us/articles/intel-array-building-blocks/
http://msdn.microsoft.com/en-us/library/hh265137%28v=vs.110%29

http://charm.cs.uiuc.edu/research/charm/
http://stellar.cct.lsu.edu/tag/parallex/
http://www.ni.com/labview/
http://chapel.cray.com/
http://x10-lang.org/

Management software: The future

- Operating systems
  - Thin generic access layers
  - Automatic adaptive schedulers learning the characteristics of the execution platform and the application
- Optimised computation kernels
  - BLAS, LINPACK, Diffpack, FFT, Boost,…
  - Learning based implementation selectors[26] and elastic computing[27]
- Development support
  - Methodology: Thinking parallel, patterns[28]
  - Languages
    - Enhanced or new languages, e.g. Lime[29]
    - Model Driven Development and Runtime[30]
  - Integrated Development Environments
  - Cross – and just-in-time compilers[31]
- Tools to assist conversion of legacy software[32]

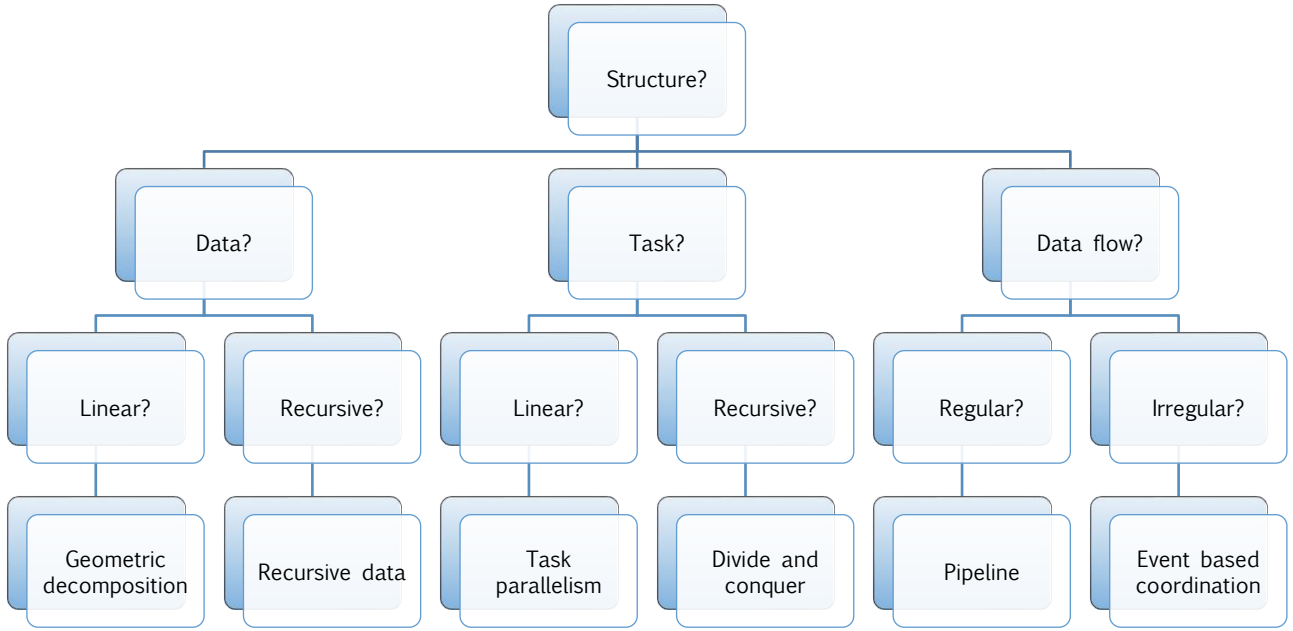
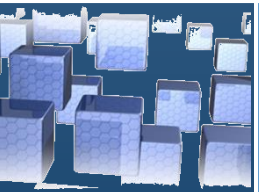


Illustration from [28]

[24] Chi-Keung Luk *et al.* 'Qilin: Exploiting parallelism on heterogeneous multiprocessors with adaptive mapping', in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42)*, New York, NY, USA, 2009, pp. 45–55.
[25] Cédric Augonnet and Raymond Namyst, 'A Unified Runtime System for Heterogeneous Multi-core Architectures', in *Proceedings of the Euro-Par 2008 Workshops on Parallel Processing*, Conference location: Las Palmas de Gran Canaria, Spain, 2008, vol. 5415, pp. 174–183.
[26] Mario Kicherer *et al.* 'Seamlessly portable applications: Managing the diversity of modern heterogeneous systems', *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 8, no. 4, pp. 42:1–20, Jan. 2012.

Paradigm: The ACTOR model[36]

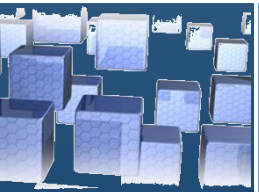
- Mathematical model of concurrent computation
  - Dynamic creation of actors
  - Asynchronous, unordered message passing
  - Concurrent computation
  - Dynamic topology – addresses in messages
- Service oriented
- No threads – no shared memory
- Legacy compliant



[36] Carl Hewitt, Peter Bishop, Richard Steiger: "A Universal Modular ACTOR Formalism for Artificial Intelligence", Proceedings of the 3rd International Joint Conference on Artificial Intelligence, pp. 235–245, San Francisco, CA, USA, Morgan Kaufmann Publishers, 1973

Virtual elasticity

- Migrating actors
- Location: Data ↔ Algorithms
- Scheduling of actors
  - Virtual threads
  - Bin packing
  - Communication aware
- Integrated interconnect



In a soldier's stance, I aimed my hand
At the mongrel dogs who teach
Fearing not that I'd become my enemy
In the instant that I preach
My pathway led by confusion boats
Mutiny from stern to bow
Ah, but I was so much older then
I'm younger than that now

Yes, my guard stood hard when abstract threats
Too noble to neglect
Deceived me into thinking
I had something to protect
Good and bad, I define these terms
Quite clear, no doubt, somehow
Ah, but I was so much older then
I'm younger than that now

Bob Dylan
My back pages

Geir Horn
Geir.Horn@mn.uio.no
+47 93 05 93 35