GR/Cosmology simulations on Next-generation Supercomputers

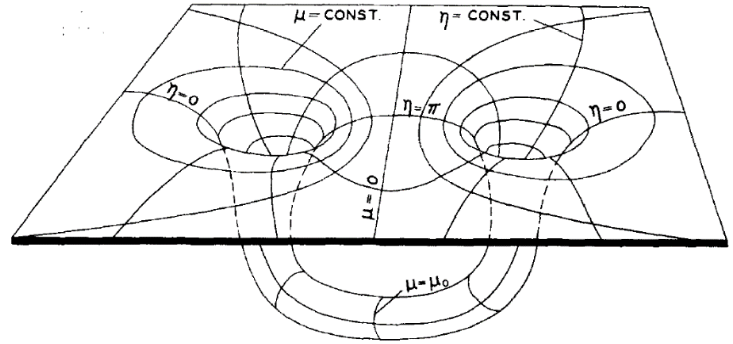
Hee Il Kim

다산데이타HPC연구소 / 서울대 천문학과

The 52nd GR and Cosmology Workshop, Sungkyunkwan Univ. 11/19/2016

Hahn & Lindquist (1964)

- Two black hole dynamics
  - Initial data: Misner's wormhole solution
  - Axial symmetry
  - Conformal decomposition???
  - Gauge condition: gaussian normal
  - 12 first order time evolution eqs
  - No clear results



$$ds_0^2 = \psi^4 \, d\bar{s}^2$$

$$\bar{\Delta}\psi - \tfrac{1}{8}\bar{R}\psi = 0$$

$$d\bar{s}^2 = d\mu^2 + d\eta^2 + \sin^2\eta \, d\varphi^2$$

$$\left[\frac{\partial^2}{\partial\mu^2} + \frac{1}{\sin\eta}\frac{\partial}{\partial\eta}\left(\sin\eta\frac{\partial}{\partial\eta}\right) + \frac{1}{\sin^2\eta}\frac{\partial^2}{\partial\varphi^2} - \frac{1}{4}\right]\psi = 0$$

$$\psi = a^{1/2}\sum_{n=-\infty}^{\infty}\left[\cosh(\mu + 2n\mu_0) - \cos\eta\right]^{-1/2}$$

$$g_{0\alpha} = -\delta_{0\alpha}$$

$$\frac{\partial\Gamma^0_{ij}}{\partial t} = \frac{\partial\Gamma^k_{ik}}{\partial x^j} - \frac{\partial\Gamma^k_{ij}}{\partial x^k} + \Gamma^l_{ik}\Gamma^k_{jl} - \Gamma^k_{ij}\Gamma^l_{kl} + g^{kl}(2\Gamma^0_{ik}\Gamma^0_{jl} - \Gamma^0_{ij}\Gamma^0_{kl})$$

$$\frac{\partial\Gamma^k_{ij}}{\partial t} = g^{kl}\left(\frac{\partial\Gamma^0_{il}}{\partial x^j} + \frac{\partial\Gamma^0_{jl}}{\partial x^i} - \frac{\partial\Gamma^0_{ij}}{\partial x^l} - 2\Gamma^m_{ij}\Gamma^0_{lm}\right)$$

IBM7090:

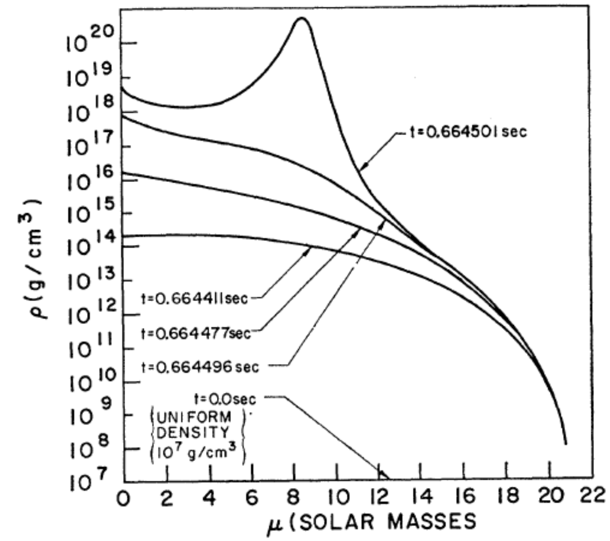
Clock Speed: ???

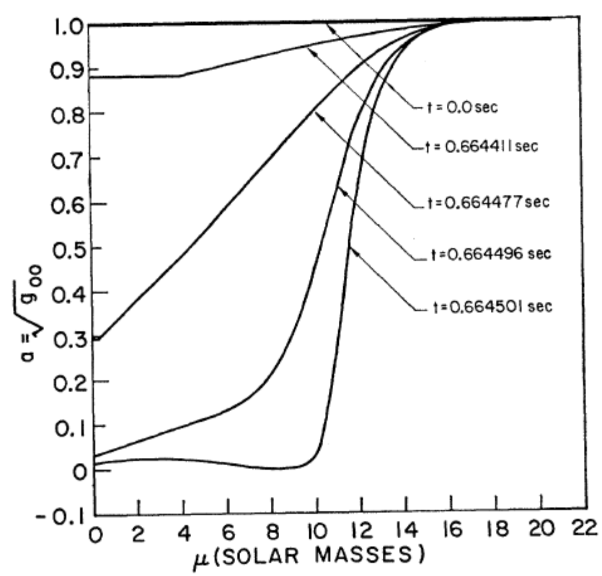
Microprocessor Peak Teraflops:
0.0000001 Tflops
Memory in TB ???

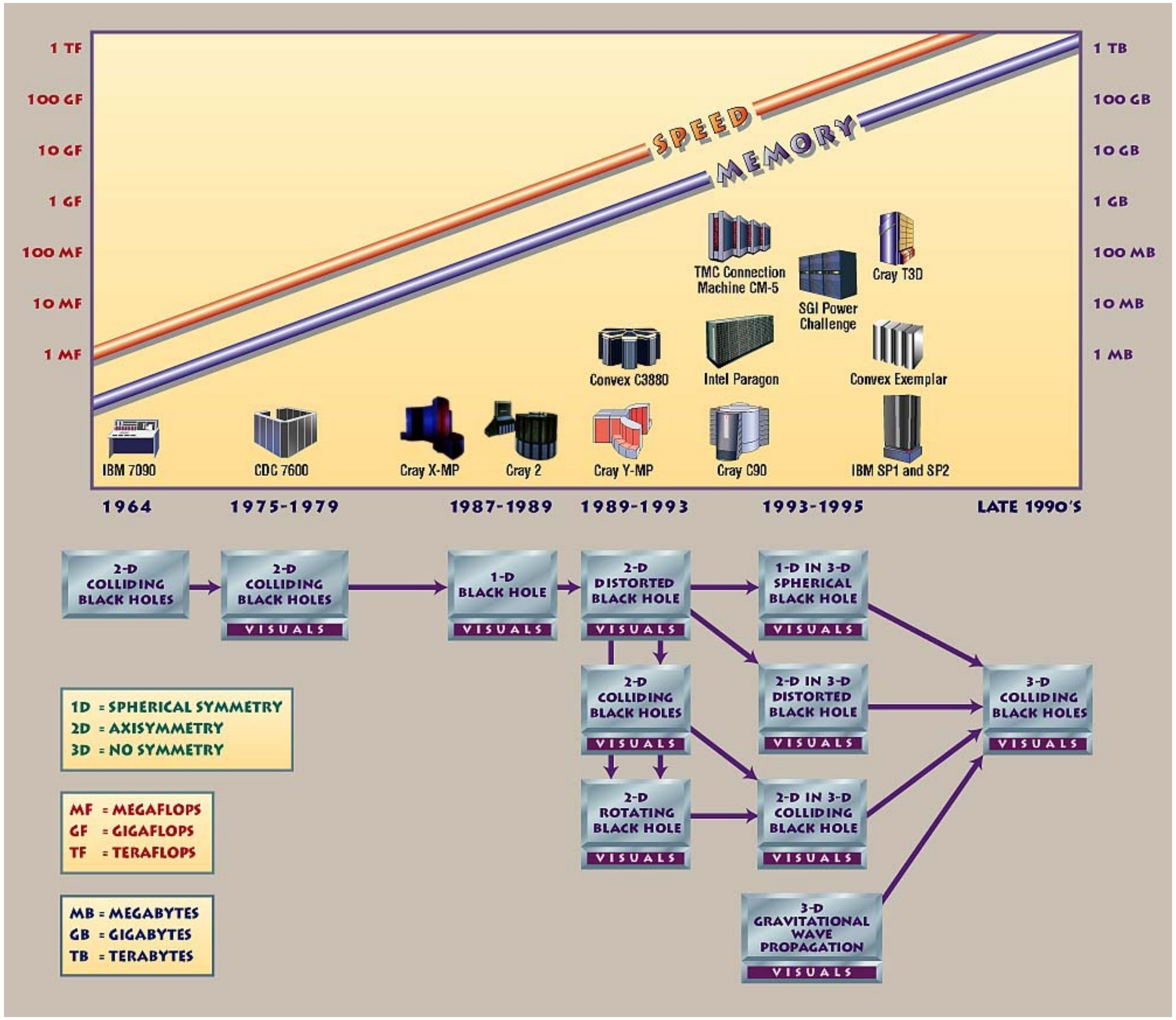
*** $ 2.9M ($ 24.2M in 2014)**

May & White (1966)

- Gravitational collapse of a star
  - Spherical symmetry (no rotation)
    - Simple ODE for initial data
  - Simple initial data
  - Simple equation of state (EoS)
  - Atmosphere: nothing
    - Lagrangian coordinates were taken
  - Shock: artificial viscosity
  - No microphysics: no dissipation, no neutrino transport, no hot nucleonic EoS, etc
  - Found the collapse conditions







2000 ~ Present: ???

- Highly nonlinear second order PDEs ~ very difficult to find solutions
- ~ 8 byte x 300 variables x 600^3 grid points ~ 500 GB

You need supercomputers and numerical relativists + Software developers, Engineers, Admins, etc

Our PCs:

- 2.9 x 8 x 8 x 2 ~ 0.4Tflops ???
- Your smartphone ~ 5Gflops

Cf. TachyonII (KISTI) ~ 300 Tflops

5th KISTI supercom ~ 30 Pflops

Goals of NR/Cosmology?

- You already know!
- Multi-scale / Multi-physics
  - NR
    - E.g. GRB, Supernova explosion:
    - 10 km vs 100 AU
    - Strong int. (Nucleonic+QCD), Weak int. (neutrino int + transport)
    - EM radiative transport, …
  - Cosmology:
    - 6000 Mpc vs 100kpc
    - Dynamics between galaxies, DM
    - Star formation, Supernova feedback, …
- How big computational resources required?
  - Simply the Bigger is the Better
  - Current goal: Exascale computing (ExaFlops, ExaByte memory, ExaByte storage, …)

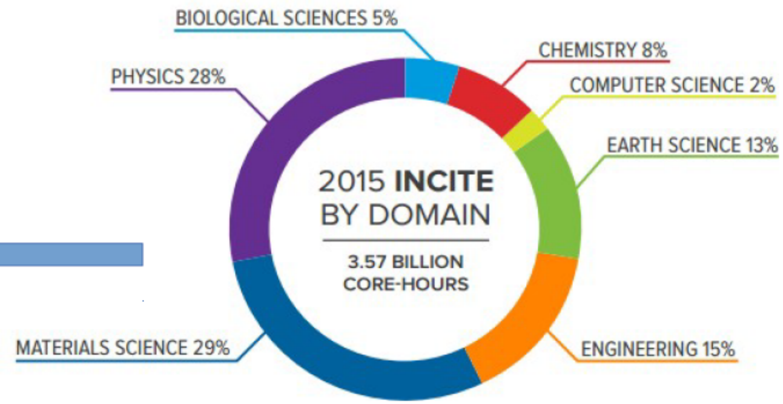
How to achieve Exa-scale?

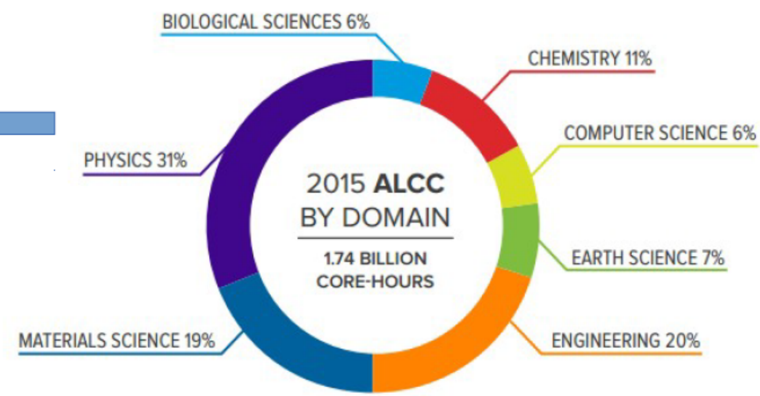
- Money makes it all ~~
- But for Green/Economic/Efficient computing
- Multi core CPU + offloading accelerators (GPU, MIC, …) + fast/efficient cache/pipelining + better instructions
- Fast memory: 3D-Xpoint ???
- Fast/stable storage with Parallel filesystem showing good scalability
- Low latency / high bandwidth interconnects + Good network topologies
- Advanced cooling system: immersion cooling,…
- Better software parallelizations
- Improved numerical codes / physics
- …

Finkel (2016)

Mira by Domain

https://www.alcf.anl.gov/files/alcfscibro2015.pdf





Finkel(2016)

Common Algorithm Classes in HPC

Algorithm / Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFTs)	Particle Methods	Structured Grids	Unstructured or AMR Grids	Data Intensive
Accelerator Science		X	X	X	X	X	
Astrophysics	X	X	X	X	X	X	X
Chemistry	X	X	X	X			X
Climate			X		X	X	X
Combustion					X	X	X
Fusion	X	X		X	X	X	X
Lattice Gauge		X	X	X	X		
Material Science	X		X	X	X		

http://crd.lbl.gov/assets/pubs_presos/CDS/ATG/WassermanSOTON.pdf

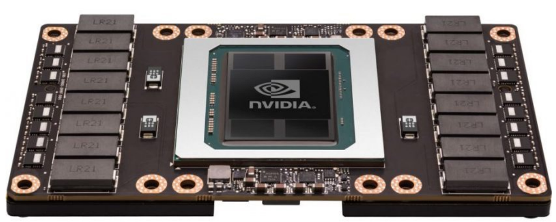
Parallelization (Load sharing)

- Types of parallelism
- ✔ Parallelism across nodes (using MPI, etc.)
- ✔ Parallelism across sockets within a node [Not applicable to the BG/Q, KNL, etc.]
- ✔ Parallelism across cores within each socket
- ✔ Parallelism across pipelines within each core (i.e. instruction-level parallelism)
- ✔ Parallelism across vector lanes within each pipeline (i.e. SIMD)
- ✔ Using instructions that perform multiple operations simultaneously (e.g. FMA)

CPU and accelerators

- CPUs:
  - Early 2000: dual core
  - 2011: AMD Bulldozer 32 core
  - 2016: Intel 24 core
- CPU instructions:
  - SIMD: SSE → AVX → AVX2 + AVX512
  - SIMT: GPU
- Offloading accelerators
  - GPUs: NVIDIA, AMD
  - MIC: Intel Many Integrated Core
    - Offloading mode
    - Native mode: Knights Coner, Knights Landing (KNL)
    - MCDRAM + DDR4: KNL (AVX512)

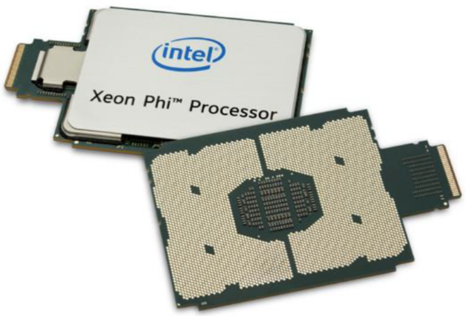
More on GPUs



GPU	Kepler GK110	Maxwell GM200	Pascal GP100
Compute Capability	3.5	5.3	6.0
Threads / Warp	32	32	32
Max Warps / Multiprocessor	64	64	64
Max Threads / Multiprocessor	2048	2048	2048
Max Thread Blocks / Multiprocessor	16	32	32
Max 32-bit Registers / SM	65536	65536	65536
Max Registers / Block	65536	32768	65536
Max Registers / Thread	255	255	255
Max Thread Block Size	1024	1024	1024
CUDA Cores / SM	192	128	64
Shared Memory Size / SM Configurations (bytes)	16K/32K/48K	96K	64K

- GPUDirect 1/2/3: Peer-to-Peer, GPUDirect for RDMA
- NVLINK: IBM+NVIDIA

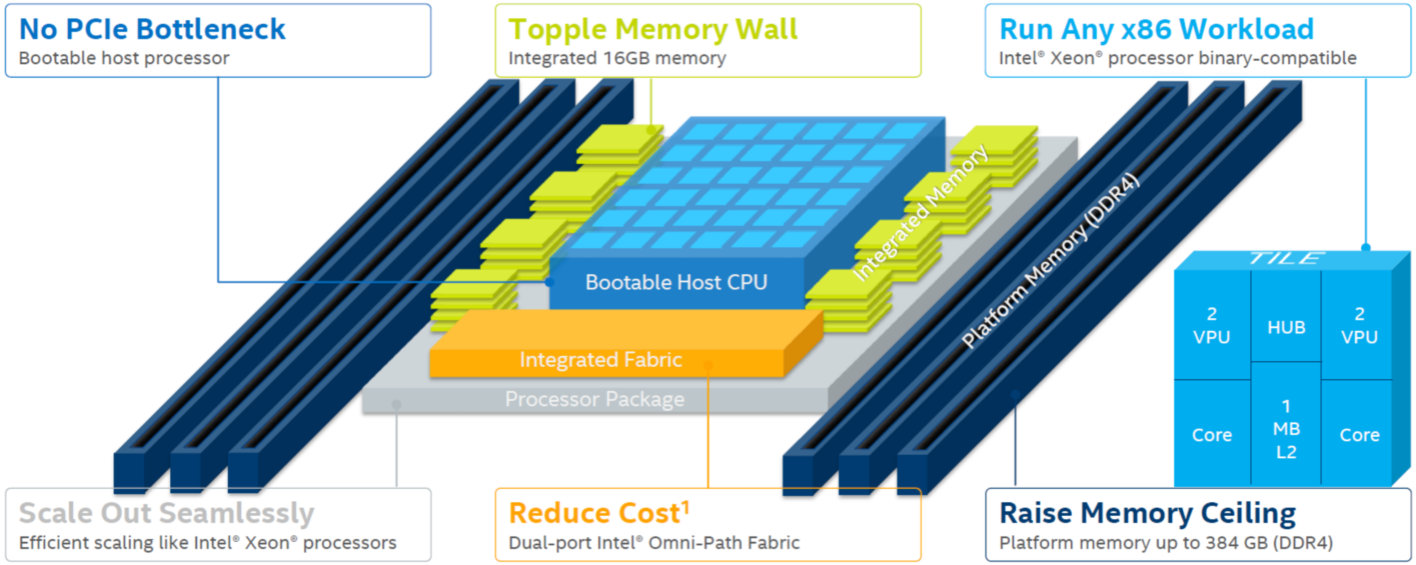
MIC



Intel® Xeon Phi™ Product Family

The transformative path to deeper insight and innovation for research and commercial applications in science, visualization, and analytics.

- **Highly-parallel performance**: Up to 72 cores with deep out-of-order buffers, Intel® Advanced Vector Extensions 512 and 3x single-thread performance compared to the previous generation product
- **Power-efficient architecture**: Delivers significantly more compute per unit of energy consumed
- **Simplified code modernization**: Reduce programming efforts and downtime by sharing code and developer base with Intel® Xeon® processors
- **Seamless IT manageability**: Common x86 architecture delivers best utilization across any workload
- **Future-ready code investment**: Code is flexible, portable, and reusable into the future as it is optimized for a general-purpose architecture using open standards

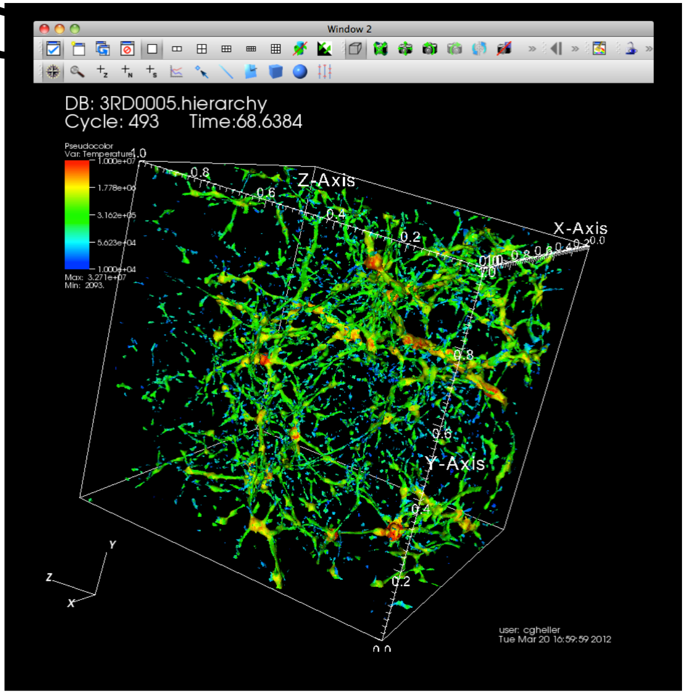


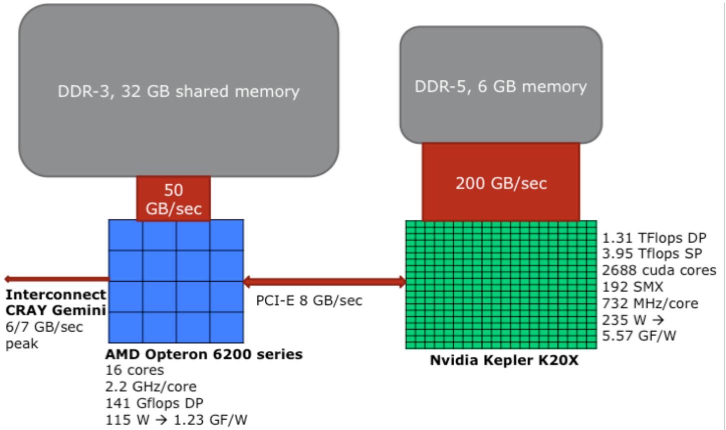
You need accelerators?

- CPU bound vs. Memory bound
- Grid-based vs. Particle-based
- High performance vs. High throughput

GPU apps : RAMSES

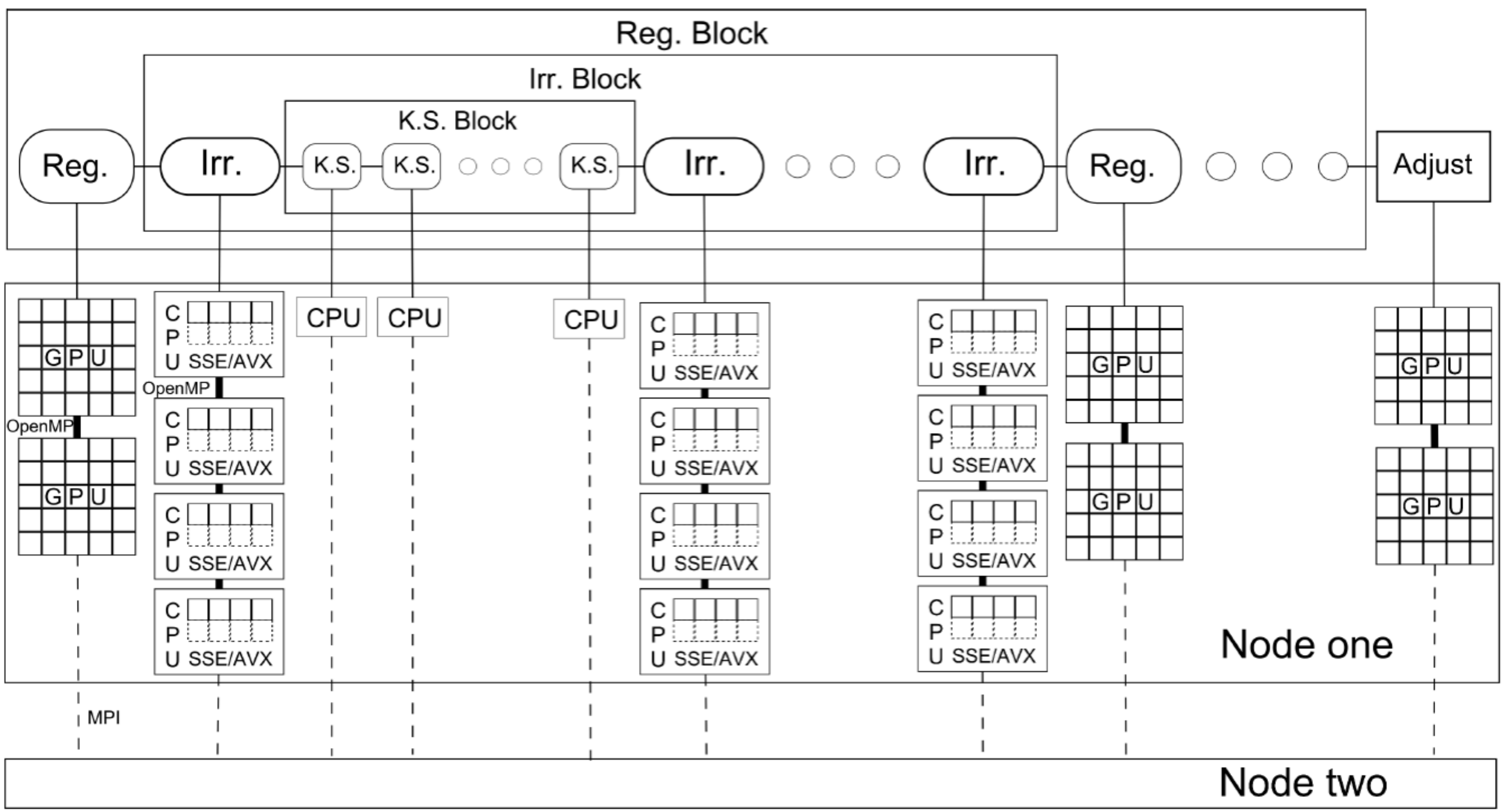
- http://www.ics.uzh.ch/~teyssier/ramses/RAMSES.html
- 3D hybrid (MPI+OpenMP) Eulerian AMR code.
- The code solves:
  - Dark matter N-body particle-mesh technique.
  - Poisson equation is solved using a multigrid technique.
  - Hydrodynamics: High resolution shock capturing
  - MHD
  - radiative cooling, ionization,
  - Star formation feedback
  - ...





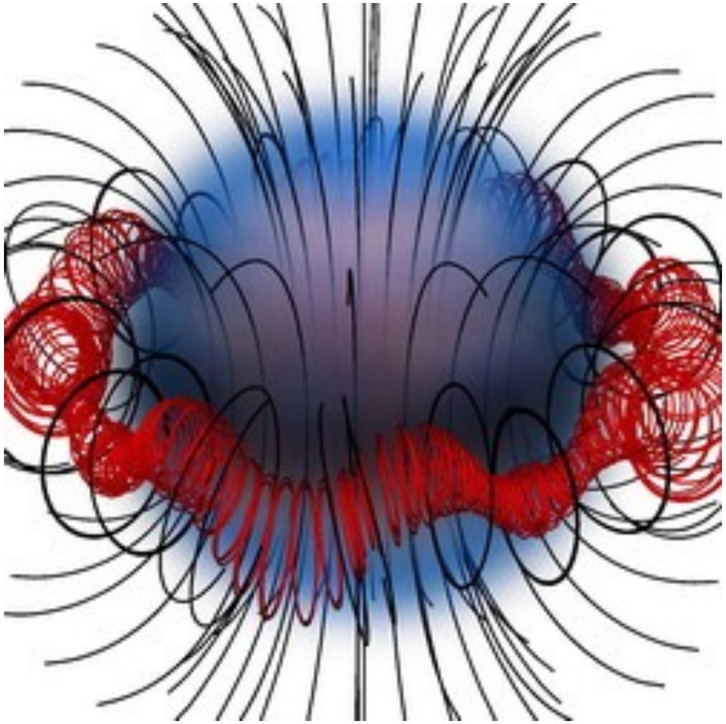
NBODY6++GPU: Ready for the gravitational million-body problem

- arXiv:1504.03687_
- https://github.com/lwang-astro/betanb6pp.git



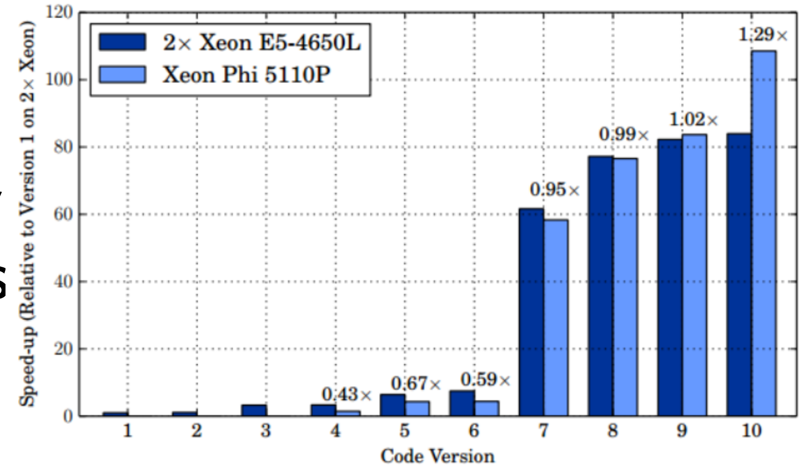
Horizon: GPU supported GRMHD code

- Lasky, P.D., Zink, B., Kokkotas, K.D., Glampedakis, K. (2011) *APJ*, *735*, L20. arXiv:1105.189
- Instability of poloidal fields of a magnetar: to explain the origin of a Soft Gamma Ray (SGR) repeater
- Black lines: poloidal fields
- Red lines: fields near zero magnetic fields
- http://vimeo.com/22986248



MIC applications

- Still limited compared to GPUs
- But all MPI codes can run smoothly on KNLs
- Cambridge's COSMOS becomes Intel Parallel Computing Center
- Reasonable direct N-body codes
- 100x performance gain for MODAL, a CMBR analysis code



Version	Processor (s)	Coprocessor (s)	Comment
1	2887.0	-	Original code.
2	2610.0	-	Loop simplification.
3	882.0	-	Intel® MKL integration routines and function inlining.
4	865.9	1991.6	Flattened loops and introduced OpenMP threads.
5	450.6	667.9	Loop reordering and manual nested threading.
6	385.6	655.0	Blocked version of the loop (for cache).
7	46.9	49.5	Numerical integration routine (Trapezium Rule).
8	37.4	37.7	Reduction with DGEMM.
9	35.1	34.5	Data alignment (for vectorization).
10	34.3	26.6	Tuning of software prefetching distances.

Pros & Cons

- GPU
  - P: No software tuning required
  - P: Nice hardware performance
  - C: Coding is not simple.
  - C: weak paralleization between nodes/gpus
    - Cf. Baidu claims nice scalability upto 128 GPUs
- MIC
  - P: Running of MPI codes
  - C: tricky code optimizations for performance

My projects on KNL

- NR using Einstein Toolkit
  - No KNL optimized yet
  - But KNC / OpenCL
  - Nice tools for cpu/memory bindings
  - Half of Opts. Will
  - Rely on Automatic AVX-512 at the moment
  - LoopControl, Vectors, etc
  - Ongoing yet.
  - Singlie KNL
    - Rotating star dynamics
- Why not for your projects?
  - If you have MPI/GPU-parallelized codes.
  - If program is not big, you may use OpenMP. It's easy & simple!
  - If you're lazy, you may rely on Automatic parallelization
    - But wailt for a while ^^

Get ready for Supercomputing
^^

Thanks!