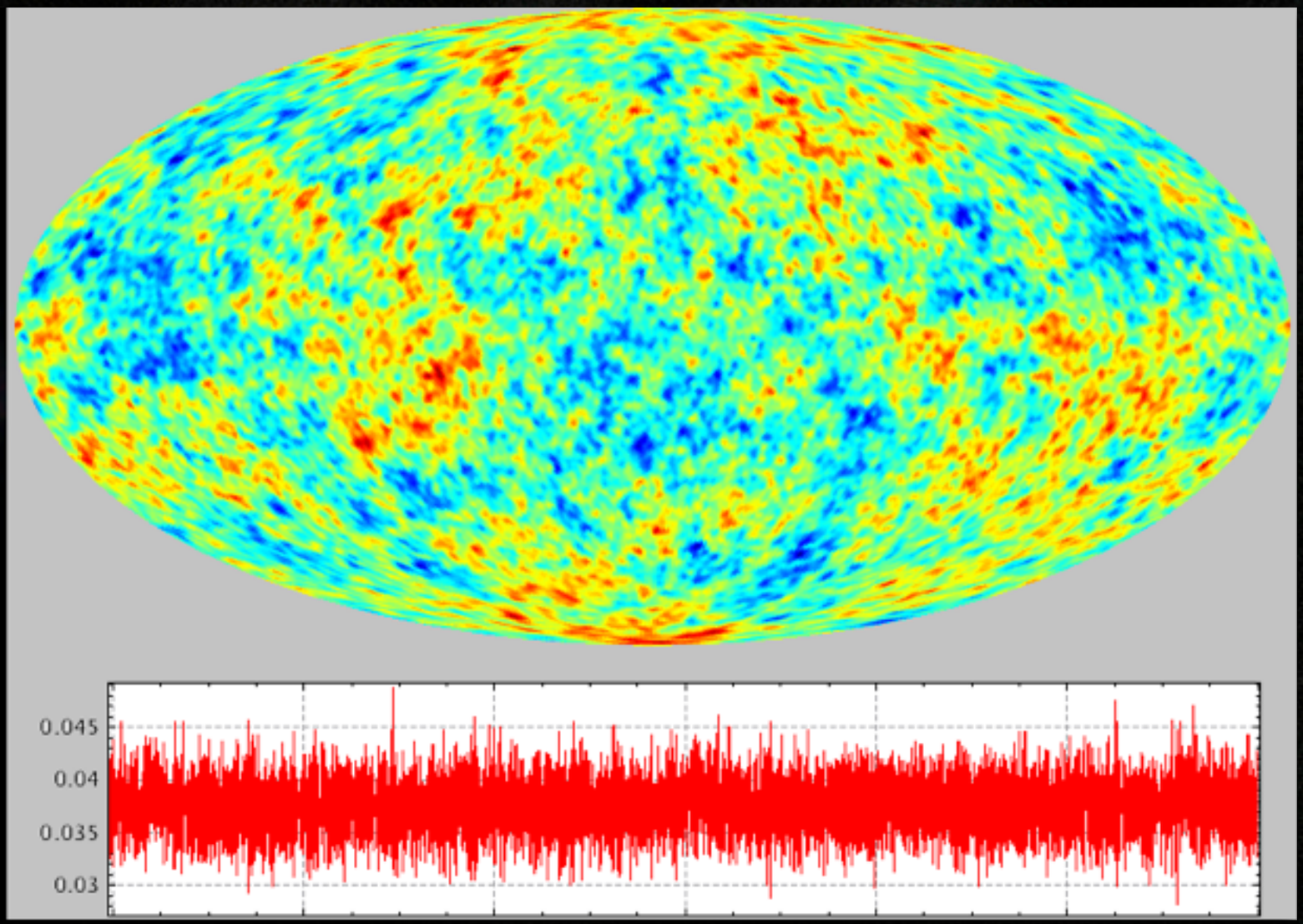


Astrophysical Data Processing on Heterogeneous Many-Core Systems

Theodore Kisner, LBNL

Astrophysical Data Processing

Data Acquisition

Data Manipulation / Calculations

Data Indexing / Storage

Data Selection / Retrieval

Data Manipulation / Calculations

High Level Results

Astrophysical Data Processing

Data Acquisition

Data Manipulation / Calculations

Data Indexing / Storage

Data Selection / Retrieval

Data Manipulation / Calculations

High Level Results

Astrophysical Data Processing

Data Manipulation / Calculations

- Image Operations: linear combination, filtering (2D FFT), sampling
- Timestream Operations: linear combination, filtering (1D FFT)
- Spherical Geometry: projection, pixelization, spherical harmonic transforms
- Monte Carlo Error Estimation: parallel random number generation

Why Many-Core Systems?

- (See Kathy's Talk)
- Electrical power is a finite resource - must increase "Flops / Watt"
- Traditional CPUs: optimize serial performance, increase clock speed, instruction-level parallelism, hardware cache management.
- The New Reality: use more transistors for calculation, pack them into many simpler cores, clock speeds ~1GHz or less, cache partially managed by software driver / application.

Many-Core Systems Today

- Most systems are heterogeneous- some traditional CPU cores for running OS, serial bottlenecks, coordination of lightweight cores, etc
- Practical performance is constrained by ***data movement*** across multi-level memory hierarchy

- Examples- multi-core CPU, plus one or more cards:

Goal : Scale Relevant Calculations

- Would like a cross-platform library of tools that “just works” for the operations important to astrophysical data processing
- Compilers are not magical, and there are no existing “mid-level” libraries that are cross-platform...

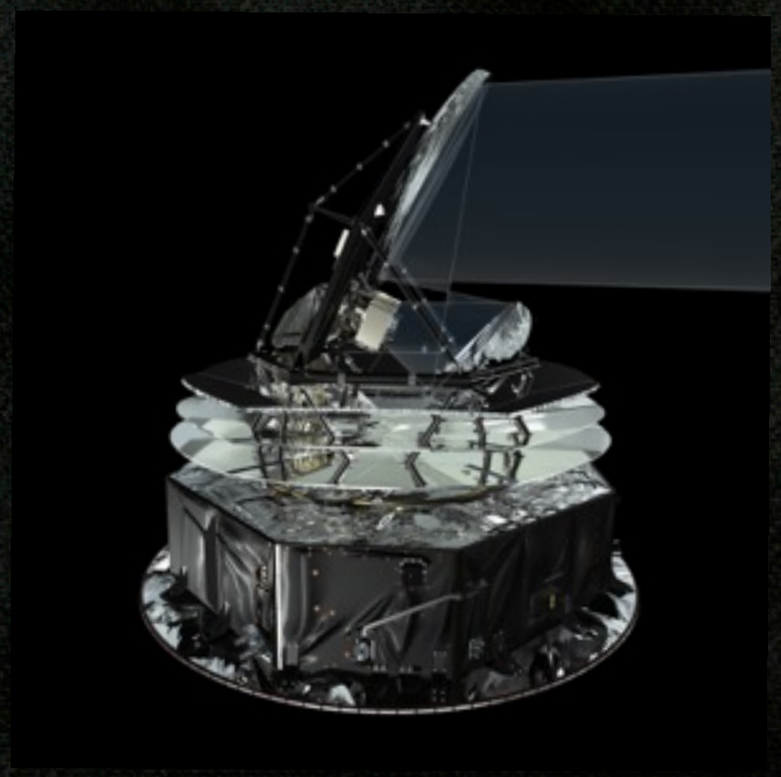
- What tools do exist?

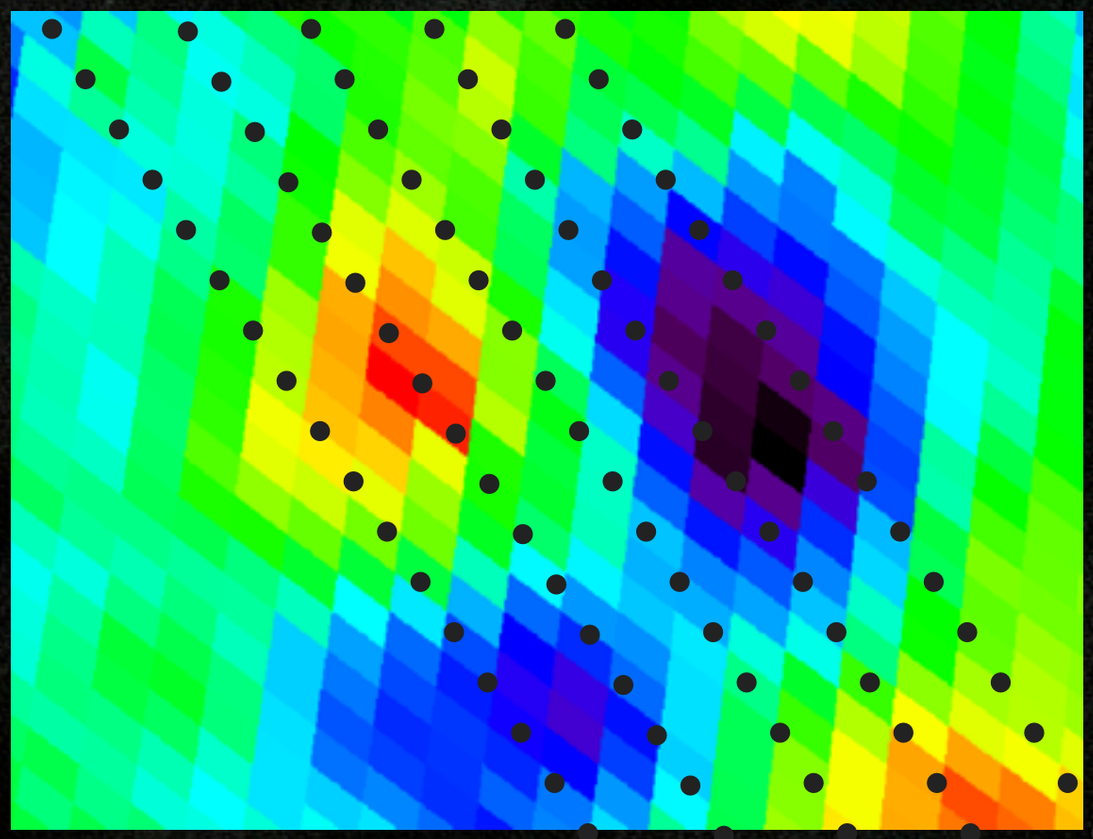
Name	Notes	Open Source?
NVIDIA CuFFT, CuBLAS	Cuda Only; subset of needed tools	NO
PGI Accelerator	Cuda Only; OpenMP style syntax	NO
AccelerEyes LibJacket	Cuda Only; Wide range of tools!	NO
BrownDeer LibStdCL	OpenCL helper tools	GPLv3
GPU Systems Libra	Cuda/OpenCL; range of math ops	NO
Intel MKL	Intel only; math ops	NO

One Path Forward - Middle Layer Tools

- Use OpenCL for cross-platform support
- Build a collection of Kernels for common data processing operations
- Do simple tuning based on detected hardware properties + simple parameter space search
- Provide a high-level interface to access these tools
- Work has already begun...

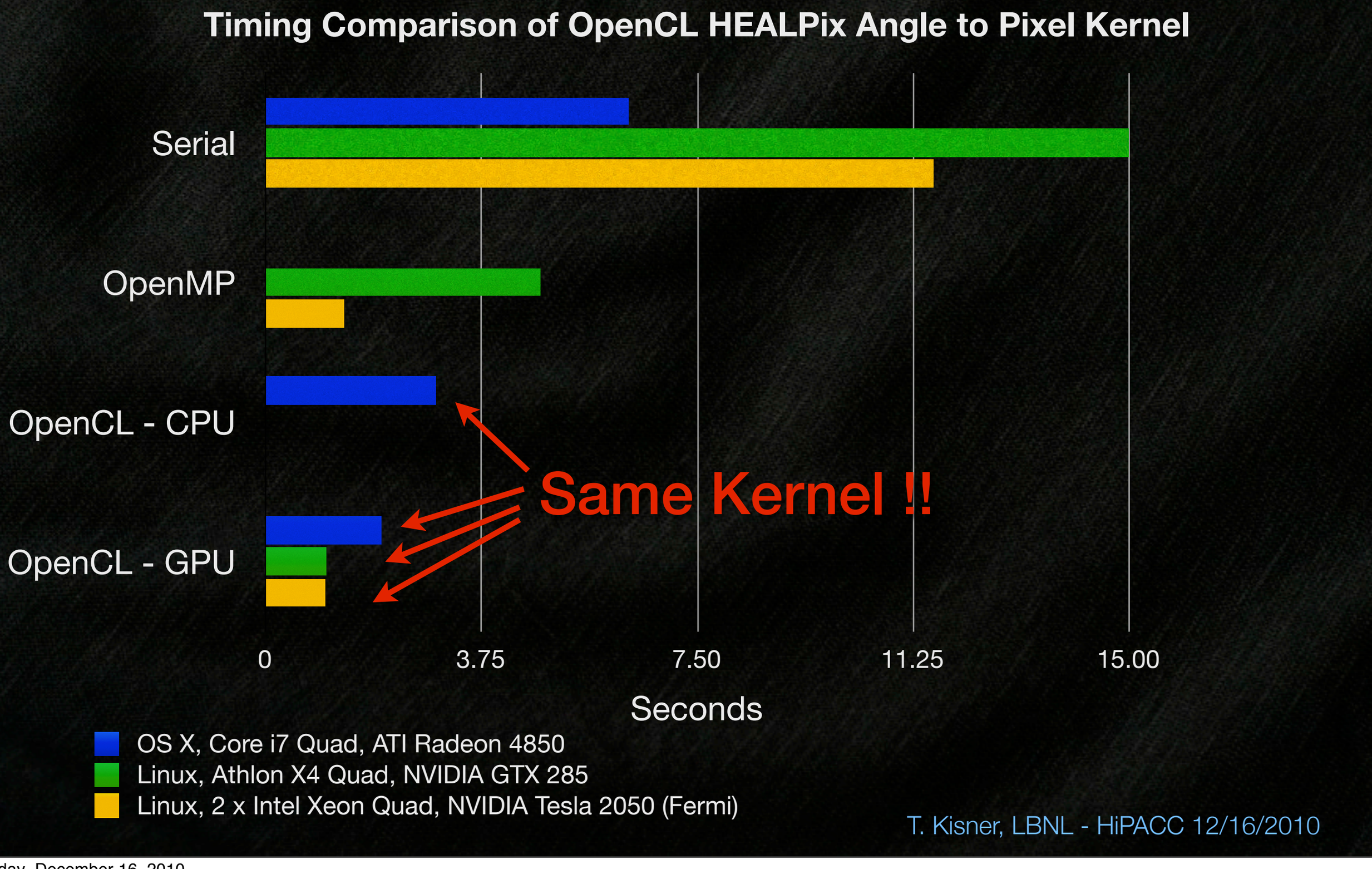
Example: Pixelization of Detector Pointing





- Traditionally done with the HEALPix software library
- Instead, implement as an OpenCL Kernel: contains one conditional, cos, sqrt, etc

Example: Pixelization of Detector Pointing



Conclusions

- OpenCL is a promising foundation for astrophysical calculations on heterogeneous many-core systems
- For real-world use, we need high-level tools
- Constructing a portable library of compute kernels for astrophysics seems both a tractable and useful path into the many-core future
- Much work to be done!