Alpaca

Cactus Tools for Application Level Profiling and Correctness Analysis

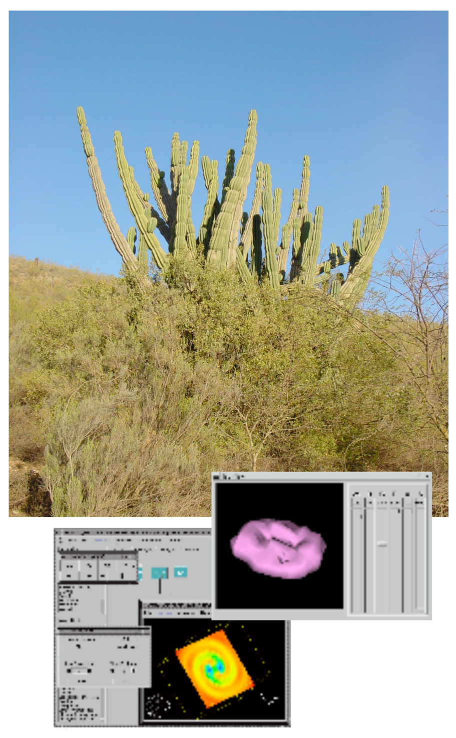
E. Schnetter (PI), G. Allen, T. Goodale, M. Tyagi
Center for Computation & Technology, LSU

High performance computing is at a critical crossroads in at least three areas:

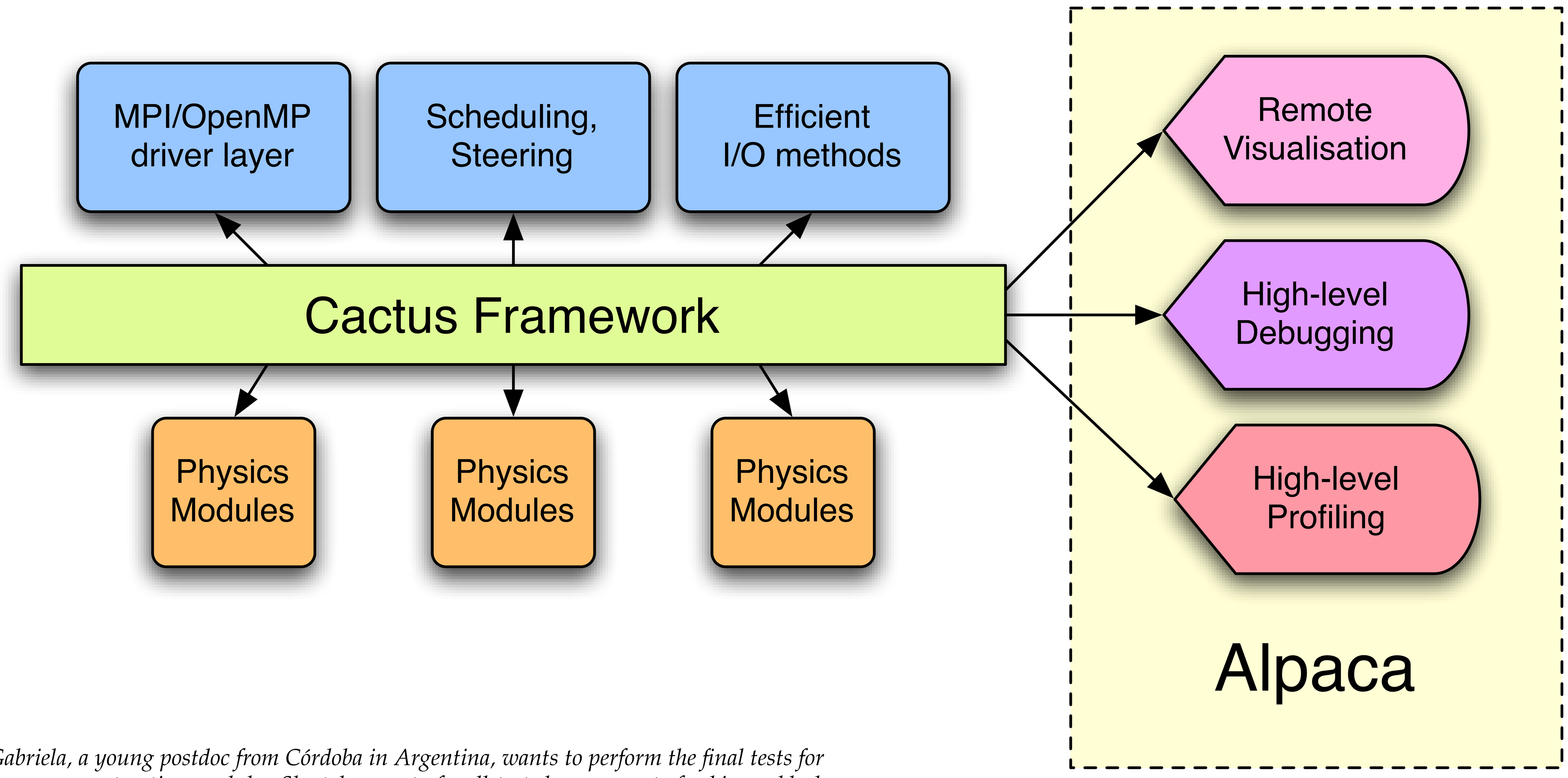
(i) *Hardware:* Radically new petascale architectures exceeding a million processors are being designed for deployment;

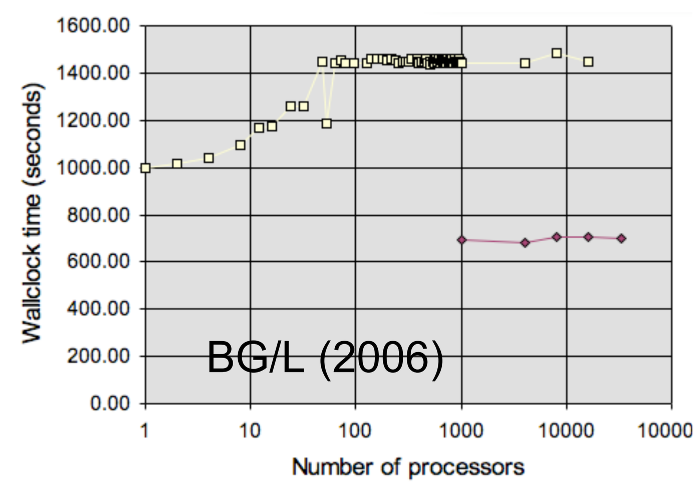
(ii) *Software:* Standard approaches to system software are outdated;

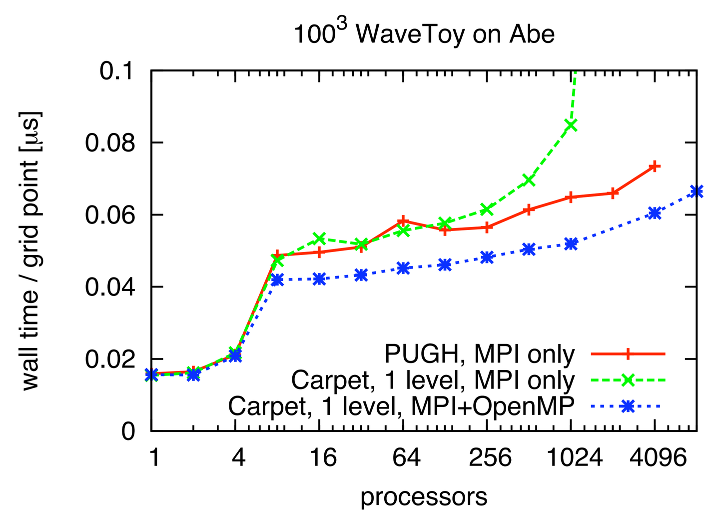
(iii) *Complex Applications:* Traditional, simplified, static applications, developed by single groups, are evolving towards highly complex codes that require teams of researchers and computer scientists to develop and use.



Application	ALPACA Users	Abbreviated List of Requirements
Numerical Relativity	Erik Schnetter (LSU, PI), Pablo Laguna[a] (Penn State), Edward Seidel (LSU, SI), Luciano Rezzolla[a] (AEI), Manuela Campanelli[a] (U. Texas)	Large scale simulations. Interest and involvement in new architectures and paradigms. Adaptive mesh refinement introduces new performance issues which need ALPACA tools.
Computational Fluid Dynamics	Mayank Tyagi (LSU, co-PI), Yaakoub El-Khamra (LSU), Kum Won Cho (KISTI)	Multi-block simulations and unstructured meshes lead to difficulties in load balancing. Many existing packages need to be integrated. Using the Cactus CFD Toolkit as educational HPC tool.
Reservoir Simulations	Christopher White (LSU, SI), Mayank Tyagi (LSU, co-PI)	High-throughput simulations. Complex geometries, elaborate physical models.
Coastal Modeling	Jim Chen (LSU, SI), Mayank Tyagi (LSU, co-PI)	Simulations require robustness & reliability. Long-term simulations (many time steps) on massively parallel computers.
Quantum Gravity	David Rideout[a] (Imperial College, UK)	Young field, requires experimenting with a wide variety of algorithms, not necessarily PDE based. Performance crucial.
Astrophysics	Joel Tohline (LSU)	Large scale simulations. Interest and involvement in new architectures and paradigms. Efficient solvers for elliptic equations need ALPACA tools.



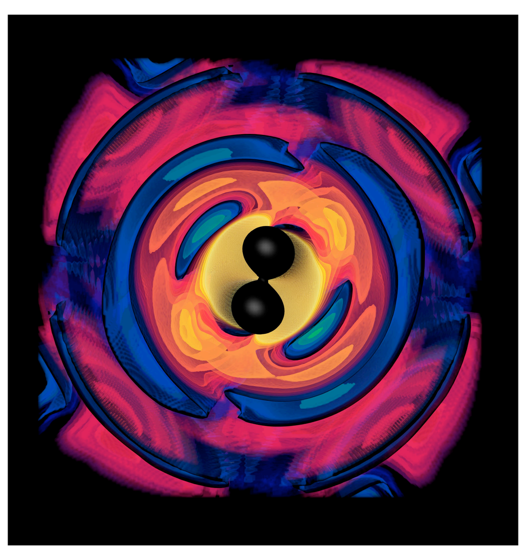




Gabriela, a young postdoc from Córdoba in Argentina, wants to perform the final tests for her new wave extraction module. She takes a set of well-tested components for binary black hole initial data, time evolution, boundary conditions, etc., and adds her new module to it. After building the application on the Tezpur supercomputer at LSU, she submits a job using a new parameter file she created.

Using the ALPACA debugger user interface, she watches the signal as the gravitational waves are detected by her module. She notices that the waveform amplitude increases with radius, which is unphysical. Still using the debugger, and still from within the same job, she single-steps through the individual algorithmic steps of her wave extraction module. She notices that the problem is caused by the lapse function, which has unexpectedly small values at small radii. Correspondingly, she switches to a different gauge condition, and after a few iterations the lapse starts to grow. Since this effect is only visible in binary black hole systems, she could not have detected it on a single-processor machine. After correcting this problem, she moves on to setting up a simulation with a high resolution to reproduce a known published result.

While waiting for the results of this simulation, she notices that the simulation makes only slow progress. Using the same ALPACA user interface, she activates some interactive performance monitoring tools for this run. These tools profile the ongoing simulation, and then access a server with "performance experience" from earlier runs without her new module, showing her that her new simulation runs only half as fast as "it should". Having this background knowledge, she is able to pinpoint the problem to a recent change in the horizon finder – not in her own code, as she first assumed. She then writes an email to the horizon finder developer asking for advice.



Alpaca will develop, *at the application level:*

(i) New fault tolerant capabilities that will be needed for increasingly large scale machines

(ii) New performance monitoring capabilities which will make it much easier to determine how the more complex application codes perform on current and future hardware

(iii) New interactive debugging capabilities, critical to locate and cure software or algorithmic errors

(iv) Integration with Eclipse, the increasingly popular code development environment.

Alpaca will be developed with full involvement from application developers across a broad range of areas.

NSF

LSU
CENTER FOR COMPUTATION & TECHNOLOGY