Roadmap

Alex Aiken

December 7, 2015

http://legion.stanford.edu

We Want You!

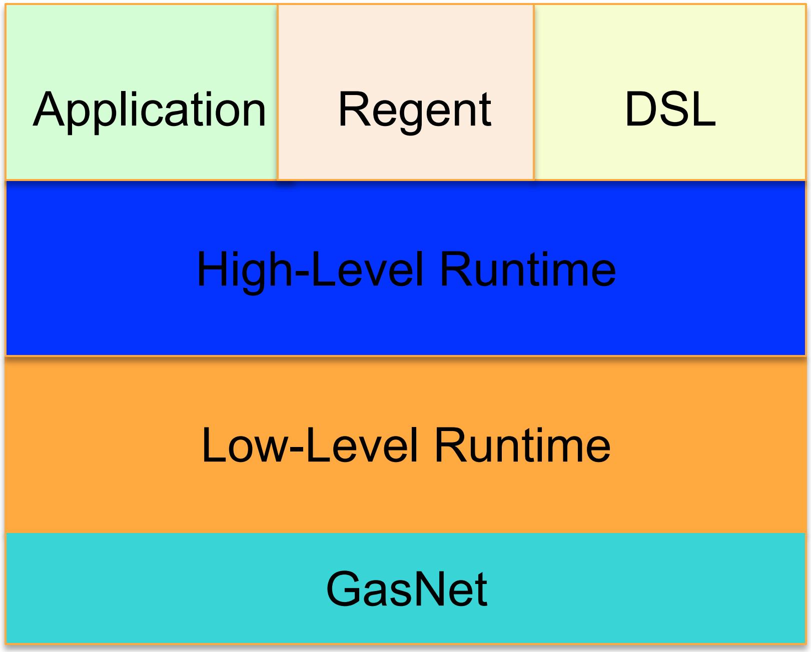
- To provide feedback.
- Much of what you’ve seen today is based on feedback/experience from the past year.
- We appreciate critical input
  - What isn’t working for you and why
  - If you know why …

In Progress (Partial List)

- Resilience
- Extensibility
- Visualization
- Disk support
- New DMA subsystem
- JIT support
- Improved index space launches

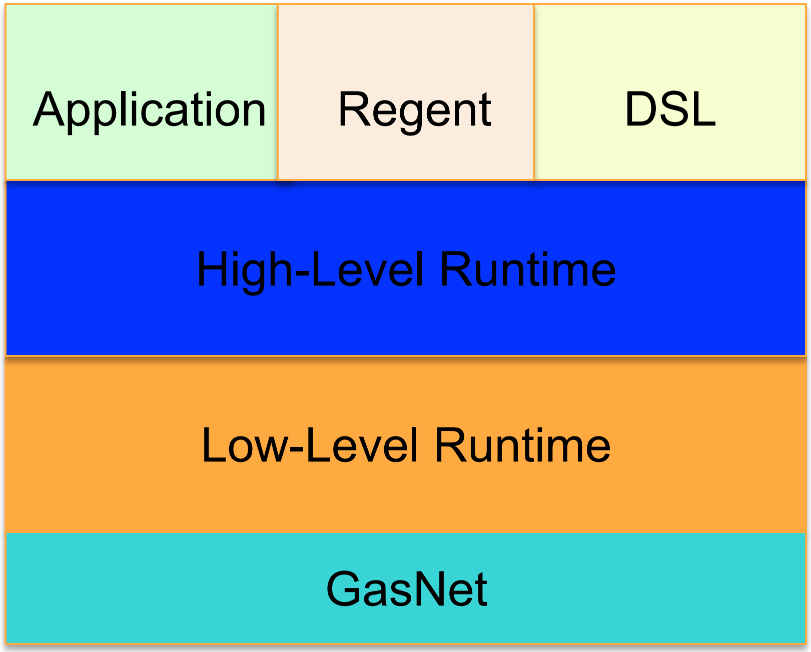
Resilience

- If a task fails, Legion has enough information in the task graph to restart it.
- Step 1: New *versioning* high-level runtime
  - done
- Other benefits
  - Faster
  - More robust design
  - Will also support speculation



Extensibility

- Modularize the low-level runtime
  - Done
- Makes it much easier to add support for new kinds of systems
  - Implement a standard interface
- E.g., Knights Landing, OpenMP



Visualization

- Pieces
  - In-situ analysis
  - External data formats
  - Integrating with visualization tools
- Done examples of in-situ analysis
  - S3D
- Plan to add OpenGL support
  - Through module interface in low-level runtime

Disk Support

- Use disk as another kind of memory
  - i.e., large regions held on disk

- Allows out-of-core algorithms with no code changes
  - A mapping decision

- Done
  - Part of the system to support external resources

New DMA Subsystem

- Data movement is complicated and important
- Custom DMA strategies generally much faster
- Consider moving data from memory A to memory B
  - Layout of data in A and B
    - AoS, SoA, dense, sparse?
  - Path from A to B
    - E.g., GPU FB -> Node RAM -> Node RAM -> GPU FB
- In progress

JIT Support

- JIT is useful when
  - Special cases can be much faster than general case
  - Special cases will be reused many times
- Example
  - DSL clients, where some runtime information influences desired code
  - DMAs: If particular path/layout combination comes up, likely to come up again
- Plan: JIT-compile and register new runtime ops
  - LLVM IR will be first supported input language

Index Space Launches

- Index space launch = launch a task for every point in an index space
- Efficiency of this operation is critical
  - Overhead proportional to # of tasks launched
  - Inversely proportion to length of tasks
- We have one solution now
  - Long-running tasks using simultaneous coherence, programmer-specified synchronization
- Investment in more automatic solutions
  - Bonus: Probably higher performance, too

The Rest

- Regent and Legion
  - Will support all features in both
  - Currently some discrepancies
- Interoperation
- Backfill expected/needed items
  - Testing
  - Documentation
  - Tools (debugging, profiling)
- Support users!
  - Help existing efforts and new ones

Tomorrow: The Exercise

- Bring a laptop and a power supply
- Just need a browser
  - Safari, Chrome, Firefox reasonably well tested
  - IE not so much ….
- Exercise will be done on Amazon’s EC2
  - Login credentials will be sent later tonight
- Coding will be in Regent

Dinner @ 6:00

MacArthur Park
27 University Ave
Palo Alto

Down Palm Drive
Between El Camino and the train tracks underpass
On left