STAPL: An Adaptive, Generic, Parallel C++ Library

Antiu Buss, Sam Jacob, Tarun Kumar Jain, Ioannis Papadopoulos, Olga Pearce, Timme Smith, Gabriel Tanase, Nathan Thomas, Xiabing Xu, Shuai Yu, Mauro Blanco, Nancy M. Amato, Lawrence Rauchwerger

STAPL Support at TAMU: stapl-support@tamu.edu
Parasol Lab, Department of Computer Science and Engineering, Texas A&M University, http://parasol.tamu.edu/

STAPL Overview

- Particle Transport Computation
  Efficient Massively Parallel Implementation of Discrete Ordinates Particle Transport Calculations

- Protein & RNA Folding
  Probabilistic Roadmap Methods from motion planning adapted to protein and RNA folding

- Seismic Ray Tracing
  Simulation of propagation of seismic rays in earth's crust.

Applications using STAPL

- User Application Code
- Views
- pContainers
- pAlgorithms
- pRange
- pGraph
- pAccumulate
- pRanges
- pContainers
- pAlgorithms
- pGraph
- pAccumulate
- pRanges

STAPL is a framework for developing parallel C++ code. Its core is a library of C++ components with interfaces similar to the (sequential) C++ Standard Template Library (STL).

Project Goals

- Ease of use
  Shared Object Programming Model provides consistent interface across shared or distributed memory systems.

- Efficiency
  Application building blocks are based on C++ STL constructs and extended and automatically tuned for parallel execution.

- Portability
  ARM runtime system hides machine specific details and provides an efficient, uniform communication interface.

pContainers and Views

- pContainer - a distributed collection of generic elements with methods to access and maintain the collection that provides a shared-view object to the user.
- The shared-view object provides uniform access to data independent of the physical location where it is stored.
- pContainer interfaces are equivalent to their STL counterparts (e.g., pvector and STL vector).

- View - an abstract data type that allows decoupling a container interface from the underlying storage.
- Views allow a data set to be filtered and traversed in multiple ways - row- or column-wise traversal of a pMatrix for example.

pRange

- Dynamic, composable task graph of a parallel computation.
- A graph vertex is a work function and the data (represented by a partition of a view) to be processed.
- Tasks that need to access the same data in a particular order have a graph edge between them to enforce the execution order when the graph is processed by the Executor in the runtime.

pAlgorithms

- Build and execute pRanges to perform parallel computation.
- The work functions used are easy to write. They look like sequential code, and can call functions from other libraries, including other pAlgorithms.
- The Factories use each encode a pattern of computation. Several pAlgorithms use the same Factory. For example, p_find and p_accumulate use the map-reduce factory.

STAPL provides parallel equivalents for all algorithms in the C++ STL. STAPL also provides pAlgorithms for its pContainers. These algorithms include traversals for pGraph and numeric algorithms for pMatrix.

References

- "Parallel Protein Folding with STAPL," G. Tanase, T. Smith, N. Thomas, L. Rauchwerger, Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.