Parallel Statistical Analysis Using STAPL

Abby Malkey, Adam Fidel, Timmie Smith, Nancy M. Amato
stapl-support@tamu.edu
Parasol Lab, Department of Computer Science and Engineering, Texas A&M University,
http://parasol.tamu.edu/

Parasol
Smarter computing.
Texas A&M University

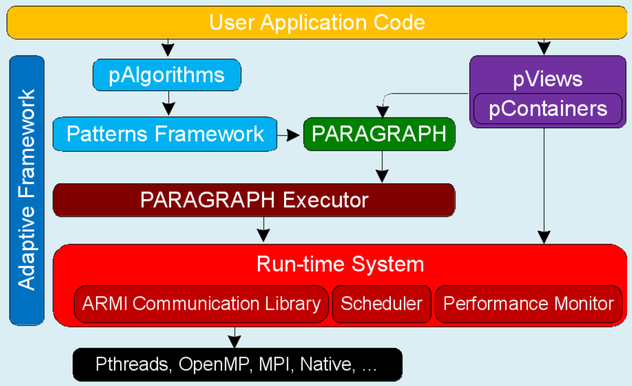
COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

Standard Template Adaptive Parallel Library (STAPL)

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 based on C++ STL constructs and extended, automatically tuned for parallel execution.
- **Portability**
  ARMI runtime system hides machine specific details and provides an efficient, uniform communication interface.



Purpose

The purpose of these experiments is to prove the ease and usability of STAPL by writing simple algorithms and showing the speedup of running the programs on different processor counts.

STAPL Components

pContainer **- a distributed collection of generic elements with methods to access and maintain the collection and provides shared-object view to the user.**

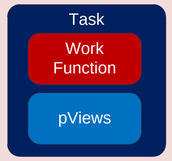
pView **– an abstract data type that separates a container interface from the underlying storage.**

pAlgorithms **–**

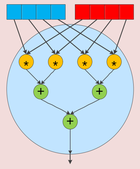
- **map reduce: applies multiple work functions to the elements of multiple views and reduces to a single answer to be assigned to a variable**
- **serial io: successively applies input operations to elements of the input view on location 0**
- **repeat view: repeats the mean value in the computation of variance**

Utilities **–**

- **do once: performs action on only one location**
- **proxy specialization: allows an object to be referred to across nodes**



Dependence Patterns for common computation patterns are provided.



Map-Reduce Task Graph

Algorithms Tested

Several algorithms were implemented in STAPL:

- **Mean, variance, and standard deviation – calculates the mean, variance, and standard deviation of a vector of randomly generated numbers**
- **Correlation coefficient – calculates the correlation coefficient for a vector of randomly generated numbers**
- **K nearest neighbors (single) – finds the *k* nearest points to a target point in a vector of points**
- **K nearest neighbors (all) – finds the *k* closest points for every point in a vector of points *currently in progress, no results***

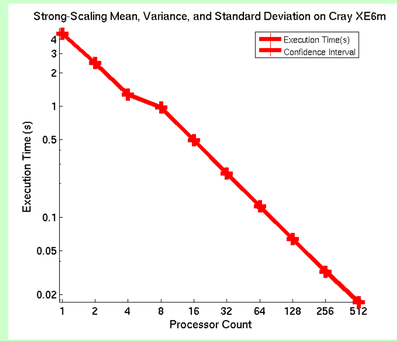
Sample Code

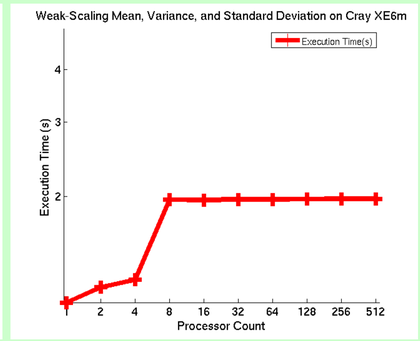
```
typedef
stapl::vector<double>
vec_type;
struct sample_wf{
   typedef void result_type;
   template <typename
Type>
   void operator()(Type t) {
     //implementation of t
   }
};
```

```
stapl_main:{
   vec_type pts;
   vec_view_type pts_view;
stapl::map_func(insert_wf(),pts_vi
ew);
   stapl::map_reduce
     (sample_wf(), reduce_wf(),
pts_view);
  stapl::do_once( //output results );
}
```

Algorithm Performance

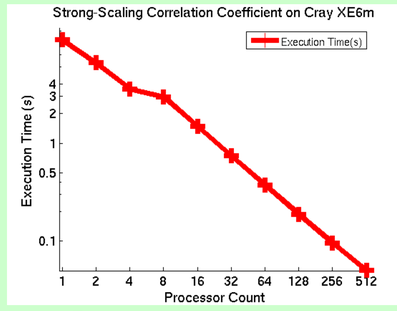
Mean, Variance, Standard Deviation

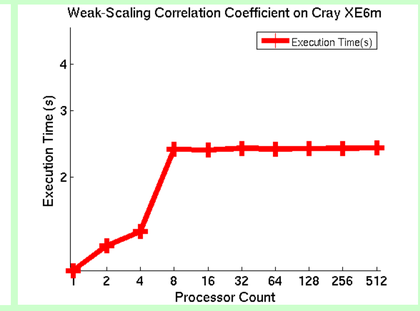




Strong-scaling times decreasing dramatically and weak-scaling times nearly linear.

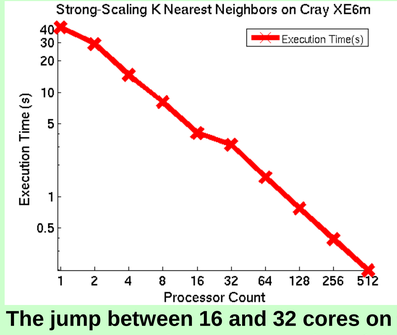
Correlation Coefficient

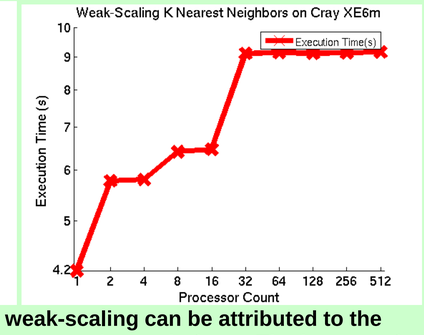




Results consistent with expectations; decreasing strong-scaling times and steady weak-scaling times.

K Nearest Neighbors





The jump between 16 and 32 cores on weak-scaling can be attributed to the processes running on two nodes of the system instead of one. The increase from one to two cores is a bit higher than expected, but it is because the program is now running on two cores and communication is occurring, whereas no communication occurs when running on only one core. Otherwise, good results.

Acknowledgment & References

This research supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11, by DOE awards DE-AC02-06CH11357, B575363, by Samsung, Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

This research used resources of the National 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.

- **"The STAPL Parallel Graph Library,"** Harshvardhan, Adam Fidel, Nancy M. Amato, Lawrence Rauchwerger, In Wkshp. on Lang. and Comp. for Par. Comp. (LCPC), Tokyo, Japan, Sep 2012.
- **"The STAPL Parallel Container Framework,"** Gabriel Tanase, Antal Buss, Adam Fidel, Harshvardhan, Ioannis Papadopoulos, Olga Pearce, Timmie Smith, Nathan Thomas, Xiabing Xu, Nedhal Mourad, Jeremy Vu, Mauro Bianco, Nancy M. Amato, Lawrence Rauchwerger, In Proc. ACM SIGPLAN Symp. Prin. Prac. Par. Prog. (PPOPP), Feb 2011.
- **"The STAPL pView,"** Antal Buss, Adam Fidel, Harshvardhan, Timmie Smith, Gabriel Tanase, Nathan Thomas, Xiabing Xu, Mauro Bianco, Nancy M. Amato, Lawrence Rauchwerger, In Wkshp. on Lang. and Comp. for Par. Comp. (LCPC), Oct 2010.
- **"STAPL: Standard Template Adaptive Parallel Library,"** Antal Buss, Harshvardhan, Ioannis Papadopoulos, Olga Tkachyshyn, Timmie Smith, Gabriel Tanase, Nathan Thomas, Xiabing Xu, Mauro Bianco, Nancy M. Amato, Lawrence Rauchwerger, In Haifa Experimental Systems Conference, Haifa, Israel, May 2010.
- **"Design for Interoperability in STAPL: pMatrices and Linear Algebra Algorithms,"** A. Buss, T. Smith, G. Tanase, N. Thomas, M. Bianco, N. M. Amato, L. Rauchwerger, In Wkshp. on Lang. and Comp. for Par. Comp. (LCPC), Jul 2008.
- **"ARMI: A high level communication library for STAPL,"** Nathan Thomas, Steven Saunders, Tim Smith, Gabriel Tanase, Lawrence Rauchwerger, Parallel Processing Letters, 16(2):261-280, Jun 2006.
- **"A Framework for Adaptive Algorithm Selection in STAPL,"** N. Thomas, G. Tanase, O. Tkachyshyn, J. Perdue, N. M. Amato, L. Rauchwerger, In Proc. ACM SIGPLAN Symp. Prin. Prac. Par. Prog. (PPOPP), pp. 277-288, Chicago, Illinois, Jun 2005.
- **"Using Load Balancing to Scalably Parallelize Sampling-Based Motion Planning Algorithms,"** A. Fidel, S. A. Jacobs, S. Sharma, N. M. Amato, L. Rauchwerger, In Proc. Int. Par. and Dist. Proc. Symp. (IPDPS), Phoenix, Arizona, USA, May 2014.
- **"Efficient Massively Parallel Transport Sweeps,"** W. Daryl Hawkins, Timmie Smith, Michael Adams, Lawrence Rauchwerger, Nancy M. Amato, Marvin Adams, Trans. Amer. Nucl. Soc., 107(1):477-481, Nov 2012.
- **"Parallel Protein Folding with STAPL,"** Shawna Thomas, Gabriel Tanase, Lucia K. Dale, Jose M. Moreira, Lawrence Rauchwerger, Nancy M. Amato, Concurrency and Computation: Practice and Experience, 17(14):1643-1656, Dec 2005.
- **"Modeling and velocity analysis with a wavefront-construction algorithm for anisotropic media**," Richard. L. Gibson Jr., Vincent. Durussel, and Kyoung-Jin Lee. Geophysics, vol. 70, issue 4, July 2005.