“Now That’s What I Call Multi-Tasking”: Evaluating a Parallel Programming Framework
Brandon Caffie, Adam Fidel, Harshvardhan, Timmie Smith, Robert Metzger, Nancy M. Amato, Lawrence Rauchwerger
Paraso Lab, Department of Computer Science and Engineering, Texas A&M University

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: Consistent interface across shared or distributed memory systems.
• Efficiency: Based on C++ STL constructs, automatically tuned for parallel execution.
• Portability: ARM runtime system hide machine specific details and provides an efficient, uniform communication interface.

How STAPL Works
• pContainer: A distributed parallel container of generic elements that provides a shared object view to the user.
• pView: An abstract data type that decouples a container from its underlying storage.

Calling a STAPL algorithm allows many cores to execute divided tasks independently and in parallel.

STAPL with 90,000 Elements
Project Euler
Project Euler is an online website hosting a collection of challenging mathematical and computational problems. Problems implemented in parallel:

• Finding the sum of even Fibonacci numbers under a given number.
• Finding the largest prime factor of a number.
• Finding the smallest positive number that is evenly divisible by 1 to n.

Mathematical expressions:
• \( a^2 + b^2 = (s - a - b)^2 \)

STAPL Implementation

STAPL with 900,000,000 Elements

Parallelism vs. Efficiency:
We implemented a parallel program to find the smallest multiple of numbers from 1 to N using brute force, which uses as many elements as needed, which can increase to very large N.

It was concluded a more efficient algorithm exists that uses N elements.

Challenges
• Load imbalance: While running a program to find the largest prime factor, we later realized that higher the number, the more prime factors occurs.
• To eliminate this, a partitioning strategy was used to evenly split the data to different cores.

Parallelism vs. Efficiency:
We implemented a parallel program to find the smallest multiple of numbers from 1 to N using brute force, which uses as many elements as needed, which can increase to very large N.

It was concluded a more efficient algorithm exists that uses N elements.

Conclusion
• Analysis shows that STAPL displays good usability and performance when implementing different Project Euler problems.
• STAPL facilitates the design of parallel programs so that novice developers can construct parallel programs for their domain.

Acknowledgment & References
This research was supported in part by NSF awards 0551586, CCF-0838916, CCF-0903703, OCI-0904168, IIS-1131720, IIS-1131721, by IBM India R&D CaRios001013, by LEIDS DE-KO20-6CHIC107, ISUT563, by Samsung, Omnova, Intel, Diamond/Sun and by Award KUS-6502-13-001, 0701 from the King Abdullah University of Science and Technology (KAUST).

This research was supported in part by NSF awards CCF-051586, CCF-0838916, CCF-0903703, OCI-0904168, IIS-1131720, IIS-1131721, by IBM India R&D CaRios001013, by LEIDS DE-KO20-6CHIC107, ISUT563, by Samsung, Omnova, Intel, Diamond/Sun and by Award KUS-6502-13-001, 0701 from the 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 DE-AC02-05CH11231.