



YIHAN SUN

Join-based Parallel Balanced Binary Trees

Friday, October 11, 2019 – 10:30 a.m. – NSH 4305

As one of the most fundamental data structure in algorithm design and programming, trees play an important role in almost every area in computer science. Advantageous and effective design of trees must consider new challenges in today's applications. First, the large scale of data requires trees to exploit parallelism and theoretical efficiency. Secondly, the comprehensiveness in applications necessitates a full-featured interface for trees. Finally, the particularity of each individual application calls forth generic frameworks for trees to unify multiple settings.

To tackle with these challenges, this thesis focuses on the balanced binary trees in the parallel setting, and proposes novel algorithms, frameworks, and implementations, that are simple and efficient for a variety of large-scale applications. The core of this thesis is an algorithmic framework called the join-based algorithms, which bases all tree algorithms on a primitive join. The join function captures the essence for rebalancing, augmentation and persistence (multi-versioning). Based on join, a variety of simple and efficient tree algorithms work generically across multiple balancing schemes, across multiple augmentations, and for both in-place and persistent updates.

Theoretically, this thesis proposes a wide range of parallel tree algorithms. They all have optimal work and poly-logarithmic span. By placing conditions on the join function, we abstract out the preferred properties of a balancing scheme that ensures the optimal cost bounds of the join-based algorithms. This also leads to theoretically efficient parallel solutions to some applications, such as the geometry data structures.

In practice, the thesis work leads to an implementation of the join-based parallel trees, called P-Trees, in a C++ library called PAM. PAM supports a complete interface for sequences, ordered sets, ordered maps and augmented maps (formally defined in this thesis). Applying the library leads to high-performance implementation to a variety of real-world applications, ranging from 2D range-based searches to hybrid transactional and analytical processing (HTAP) database management systems.

P-Trees enable both concise implementations and high performance for all tested applications. Experiments show that P-Trees achieve speedups 40-100 across different applications on 72 cores with hyperthreading. P-Trees' performance can be up to magnitudes better than existing solutions specific to these applications.

Thesis Committee:
Guy E. Blelloch, Chair
Andrew Pavlo
Daniel D. K. Sleator
Michael T. Goodrich, University of California, Irvine