Today, the largest publicly available graph dataset is the Hyperlink Web graph, with over 3.5 billion vertices and 128 billion edges. Despite being publicly available for several years now, most experimental work in the literature reports results on much smaller graphs, or resorts to external or distributed memory to process the Hyperlink graph. A natural question then, is to ask whether we can efficiently solve a broad class of problems on this graph in memory. Given that existing systems capable of processing this graph only support static processing, another interesting question is whether we can efficiently represent and update this graph in memory.

In the first part of this talk I will present theoretically efficient implementations of parallel graph algorithms for a set of 20 important graph problems. Our codes, which we have publicly open-sourced as part of the Graph Based Benchmark Suite (GBBS) solve these problems on the Hyperlink Web graph using a single commodity multicore machine with a terabyte of RAM, with most of the codes running in under a minute on this graph. The running times of our implementations outperform existing state-of-the-art implementations on the largest real-world graphs, and for many of the problems that we consider, this is the first time they have been solved on graphs at this scale.

In the second part of this talk I will present Aspen, a framework for processing streaming graphs, which are graphs that evolve over time via a stream of updates (batches of edge insertions/deletions). The key idea is to represent the adjacency information using a purely-functional compressed tree data structure called a C-tree, which can be efficiently updated. Aspen achieves millisecond latency for processing small batches, and achieves throughputs of hundreds of millions of updates per second for very large batches, while providing strict serializability for both queries and updates. Using Aspen, we are able to concurrently update and run analytics on the Hyperlink Web graph using a single commodity multicore server with a terabyte of RAM.

Based on joint work with Guy Blelloch and Julian Shun.