Distributed Suffix Trees on Apache Hama

Reid Horton and Joohyun Kim
Suffix Trees

Suffix trees are valuable for speeding up string processing algorithms, such as exact and inexact substring matching.

However, suffix trees require much more memory than other alternatives, such as suffix arrays, which provide similar speedups for the same set of algorithms.
Project Goals

We have implemented a similar data structure that can be spread out across many computers, allowing one to work with larger data sets than can be handled by a single machine.

Our long term goal is to develop a transcriptome assembler that can run in the cloud by combining this distributed data structure with existing genome assembly software.
How Suffix Trees Work

Suffix trees store every suffix of a string in a tree, representing each suffix with a path from the root to a leaf node.
Distributed Suffix Trees

When main memory is exhausted, computations on data become unacceptably slow. Regular suffix trees are relatively space inefficient, and will quickly run out of memory.

By distributing the data across computers, we can increase available memory by adding more compute nodes.
Distributed Suffix Trees

In a distributed suffix tree, several computers work together to store all the suffixes of the original string.

Each computer is assigned a different subset (or subsets) of suffixes, and they build sparse suffix trees containing only those subsets.
Sparse Suffix Trees

Several algorithms already exist to construct regular suffix trees. However, sparse suffix trees only contain a subset of all the suffixes of a string, so their construction is somewhat different.

There are a few papers that describe rules for building sparse suffix trees, but turning that into a working program is less straightforward.
Apache Hama

Hama is a distributed computing framework that runs on top of Hadoop, and it allows us to manage efficient communication between compute nodes.

While sparse suffix trees can be constructed separately without communication, we hope to leverage more of Hama’s power while developing the assembly program.