

Keynote Talk

Experiences with MapReduce, an Abstraction for Large-Scale Computation

Jeffrey Dean
Google, Inc.
Mountain View, California, USA
jeff@google.com

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a Map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a Reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The MapReduce run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: thousands of MapReduce programs have been implemented and several thousand thousand MapReduce jobs are executed on Google's clusters every day.

In this talk I'll describe the basic programming model, discuss our experience using it in a variety of domains, and talk about the implications of programming models like MapReduce as one paradigm to simplify development of parallel software for multi-core microprocessors.

Keywords: Algorithms, Design, Performance, Reliability

Bio

Jeff joined Google in 1999 and is currently a Google Fellow in Google's Systems Infrastructure Group. While at Google he has worked on Google's crawling, indexing, query serving, and advertising systems, implemented several search quality improvements, and built various pieces of Google's distributed computing infrastructure. Prior to joining Google, he was at DEC/Compaq's Western Research Laboratory, where he worked on profiling tools, microprocessor architecture, and information retrieval. He received a Ph.D. from the University of Washington in 1996 working with Craig Chambers on compiler optimization techniques for object-oriented languages. Prior to graduate school, he worked at the World Health Organization's Global Programme on AIDS.