Google's MapReduce Programming Model — Revisited

Ralf Lämmel
Data Programmability Team
Microsoft Corp.
Redmond, WA, USA

Abstract

Google's MapReduce programming model serves for processing and generating large data sets in a massively parallel manner (subject to a suitable implementation of the model). We deliver the first rigorous description of the model. To this end, we reverse-engineer the seminal MapReduce paper and we capture our observations, assumptions and recommendations as an executable specification. We also identify and resolve some obscurities of the informal presentation in the seminal MapReduce paper. We use typed functional programming (specifically Haskell) as a framework for design recovery and executable specification. Our development comprises three components: (i) the basic program skeleton that underlies MapReduce computations; (ii) the opportunities for parallelism in executing MapReduce computations; (iii) a higher-level approach to MapReduce-like computations, inspired by the aggregators in Google's domain-specific language Sawzall. Our development does not formalize the more implementational aspects of an actual, distributed execution of MapReduce computations.

Keywords: Data processing; Parallel programming; Distributed programming; Software design; Executable specification; Typed functional programming; MapReduce; Sawzall; Map; Reduce; List homomorphism; Fold; Unfold; Bananas; Haskell.