NuMRF: A Numerical MapReduce Framework

W. Larson,
Mathematical Sciences Institute, The Australian National University

MapReduce is a functional programming pattern that has revolutionised commercial data analytics. A MapReduce application comprises two user-defined processing stages, whose composition implements end-to-end processing. These stages are: a Map() stage, which transforms its input into a stream of `<key, value\>` pairs; and a Reduce() stage that takes as its input `<key, value>` pairs grouped by key and computes a single final value for each key. A MapReduce framework performs a global sort to connect Map() outputs to Reduce() inputs. This approach endows the application with parallelism with little-to-no user effort, and enable it to run successfully in the face of faults. MapReduces use of sorting is a potent strategy for data with unknown structure. But what if the data under analysis are highly structured and this structure is known a priori? My work, which is supported by the Open Petascale Library project, aims to create a scientist-friendly, high performance computing-capable MapReduce solver execution framework called NuMRF. NuMRF is currently under development, but when completed will offer a python-based calling framework supporting multiple levels of parallelism and fault-tolerance, with the added attraction of compatibility with legacy MPI-based parallel codes and support for iterative MapReduce processing. I will begin with a brief overview of the emerging leadership-class computing landscape and MapReduce. I will describe how MapReduce can be used directly in scientific computing problems (e.g., climate data analysis), and what changes to the pattern would support better numerical analysis use cases. I will describe the NuMRF design, elucidating its elements and how this strategy may support implementing multilevel parallelism with relative ease. I will make a link between the requirements for a scientific computing MapReduce system and the coupling infrastructure found in Earth system models and other multi-physics/multiscale simulators. I will the present the NuMRF data model–the python Grids and Fields Toolkit (pyGraFT)–and describe how this toolset may simplify the construction of new user-defined parallel Map() functions. I will conclude with a discussion of possible use cases in the area of high-dimensional PDE solvers and climate modelling and analysis.