Chapter 4
Parallel Applications

▼ Part I. Preliminaries
  Chapter 1. What Is Parallel Computing?
  Chapter 2. Parallel Hardware
  Chapter 3. Parallel Software
  Chapter 4. Parallel Applications
  Chapter 5. Supercomputers
► Part II. Tightly Coupled Multicore
► Part III. Loosely Coupled Cluster
► Part IV. GPU Acceleration
► Part V. Map-Reduce
► Appendices
Parallel computing applications are characterized along two orthogonal dimensions (Figure 4.1): little CPU—big CPU, and little data—big data.

Unlike the parallel hardware and software dimensions, most of which were discrete, the application dimensions are continuous. Any parallel application might fall anywhere along the CPU and data dimensions: an application might involve lots and lots of computation (big CPU) but only a modicum of data (medium-small data); or an application might involve a fair amount of data (medium-big data) but only an intermediate amount of computation (medium CPU); and so on. The next few sections explore the extremes of the dimensions.

**Little CPU Little Data Application**

A little CPU little data application works with only a small amount of data, and does only a few calculations (CPU cycles) with each data item. Still, the calculations are or can be done in parallel. A spreadsheet is an example. Compared to a supercomputer program, a spreadsheet works with very little data (the cell values) and does very little computation (the cell formulas). However, the cells can be calculated in parallel, as long as any data dependencies between cells are obeyed.

**Big CPU Little Data Application**

A big CPU little data application also works with only a small amount of data, but it spends a large amount of CPU time doing calculations with that data. Doing the calculations in parallel can speed up the application. Cryptographic applications are often of this kind. Bitcoin mining is one example. A bitcoin “block” is a piece of digital cash. It occupies just a few kilobytes of data. But determining the value of a certain field of the block—“mining” the bitcoin, as it is called—requires calculating the SHA-256 cryptographic hash function many, many times. These calculations can be, and usually are, performed in parallel. Also, bitcoin miners can mine multiple blocks in parallel.

**Little CPU Big Data Application**

A little CPU big data application devotes only a little bit of CPU time to each data item, but works with an enormous number of data items. Consequently the application can take a long time to run, and processing the data in parallel can speed it up. Map-reduce, pioneered by Google and implemented in the popular Apache Hadoop, is a widely used paradigm for parallel big-data applications. Apache’s “Powered by Hadoop” web page* shows that many major Internet players—Amazon, eBay, Facebook, Hulu, LinkedIn, Spotify, Twitter, Yahoo, and others—use Hadoop running on multicore clusters for their big data analytics. Google also does big data analytics with their own map-reduce software. The Parallel Java 2 Library includes Parallel

---

* [https://wiki.apache.org/hadoop/PoweredBy](https://wiki.apache.org/hadoop/PoweredBy)
Java Map Reduce (PJMR), a lightweight map-reduce framework built on top of the Library’s cluster parallel programming capability.

**Medium CPU Big Data Application**

Sometimes, an application described as “big data” refers, not so much to the quantity of data, as to the dimensionality of the data. Machine learning applications are often of this kind. For example, consider the KDD Cup 1999 data set, a network intrusion data set. The full data set, consisting of 4.9 million records, occupies 743 megabytes of storage. Each record consists of a series of features describing one network intrusion incident, such as the duration of the incident, the network service that was attacked, whether the attacker was able to log into a root shell, and so on; 41 features in all. In terms of data quantity, this is not a very big data set; Google and Facebook store much, much more data. But the data set is high-dimensional (each feature is one dimension), and algorithms that try to find patterns in the data set must spend considerable CPU time to deal with the large number of dimensions. Such applications might be characterized as medium CPU big data.

**Big CPU Big Data Application**

Finally, a big CPU big data application works with lots and lots of data and does lots and lots of calculations with each data item. Scientific and engineering calculations on supercomputers are of this kind. As an example of the extreme scale of these applications, consider the LAMMPS molecular dynamics program, which simulates the motion of atoms from first principles of physics, running on the Keeneland supercomputer at the Oak Ridge National Laboratory. Keeneland is a medium-size cluster of 120 nodes, with two CPU cores and three GPU accelerators per node. LAMMPS running a one billion atom benchmark for 100 time steps requires about half a terabyte ($5 \times 10^{11}$ bytes) of data and would take the better part of an hour (2350 seconds) on one core of Keeneland. Running on the entire cluster, the same benchmark takes just 17.7 seconds.†

Points to Remember

- Parallel computing applications can be characterized along two dimensions: little CPU—big CPU, and little data—big data.
- “Big data” can refer to the quantity of the data or to the dimensionality of the data.