Data-intensive HPC: opportunities and challenges

Patrick Valduriez

▶ **To cite this version:**

Patrick Valduriez. Data-intensive HPC: opportunities and challenges. BDEC: Big Data and Extreme-scale Computing, Barcelona Supercomputing Center, Jan 2015, Barcelone, Spain. lirmm-01184018

HAL Id: lirmm-01184018
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01184018v1

Submitted on 12 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Data-intensive HPC: opportunities and challenges

Patrick Valduriez
Inria, Montpellier
http://www-sop.inria.fr/members/Patrick.Valduriez/

Data-intensive computing and HPC are two different computing paradigms (data-centric versus compute-centric), targeting different application domains (mainly business data processing versus computational science). But with the advent of big data, the blending of these areas into data-intensive HPC offers new opportunities to solve bigger problems, as well as new research challenges.

Big data has become a buzzword, referring to massive amounts of possibly complex data that are very hard to deal with traditional data management tools. In particular, the ability to produce high-value information and knowledge from big data makes it critical for many applications such as decision support, forecasting, business intelligence, research, and (data-intensive) science. Processing and analyzing big data (big analytics) is a major challenge since solutions must combine new data management techniques (to deal with new kinds of data) with large-scale parallel processing in computer clusters.

Parallel data processing has long been exploited in the context of database systems for highly structured data, i.e. represented using the relational data model. The fundamental principle behind database systems is data independence, which enables applications and users to share data at a high conceptual level while ignoring implementation details. It is this principle that yielded advanced capabilities such as high-level query languages such as SQL, schema and metadata management, access control, automatic query processing and optimization, transactions, efficient data structures and indexes, etc. In particular, it allows exploiting data parallelism through data partitioning to process large volumes of structured data, as for instance, in commercial data warehousing.

But big data encompasses different data formats (documents, sequences, graphs, arrays, …) that require significant extensions to traditional parallel database techniques. Data-intensive computing addresses this challenge in a way that is both scalable and fault-tolerant, by introducing new distributed file systems such as GFS and HDFS, new NoSQL database systems such as Bigtable and Hbase and new parallel programming frameworks such as MapReduce and Spark which provide powerful operators. Today, the major commercial DBMS companies (Oracle, IBM, Microsoft, …) have integrated these solutions into their offering to allow big data processing against both structured and unstructured data.

HPC applications are now also facing big data challenges, as they need to ingest more input data, e.g. from more powerful scientific instruments or sensor networks, and produce more output data for analysis, e.g. with smarter mathematical models and algorithms. Furthermore, analytics (i.e. the discovery of meaningful patterns and insights in data) becomes a newer, complementary big data market for HPC, with analytics methods applied to established HPC domains or high-end commercial

analytics pushing up into HPC.

While they share a number of similar challenges, data-intensive computing and HPC have major differences, particularly in terms of architectures (uniform storage versus hierarchical storage), file systems (few big files versus many smaller files), programming models (algebraic operators versus MPI), and programming languages (Java or Python versus C or C++).

Still, there are major opportunities for blending the two areas into data-intensive HPC, with the ability to solve bigger problems faster. In particular, applying the data independence principle to HPC could yield to smarter parallel file systems, with the ability to query metadata and use more efficient indexing techniques, and high-level parallel programming frameworks with efficient operators. On the other hand, introducing compute-centric techniques into data-intensive computing could yield better analytics, e.g. by supporting more complex machine learning models.

Data-intensive HPC brings also important challenges that will need cooperation from the data management and HPC communities. These challenges include data storage, data sharing, data movement, metadata management, parallel programming frameworks, tools for large-scale data integration and analysis, and data privacy.