Massivizing = Scalable, High Performance, Reliable, Efficient Graph Processing Systems

dr. ir. Alexandru Iosup
Parallel and Distributed Systems Group
Societal Challenges

The quadruple helix: prosperous society & blooming economy & inventive academia & wise governance depend on datacenters

- Enable data access & processing as a fundamental right in Europe
- Enable big science and engineering (2020: €100 bn., 1 mil. jobs)
- “To out-compute is to out-compete”, but with energy footprint <5%
- Keep Internet-services affordable yet high quality in Europe
- The Schiphol of computation: Netherlands as a world-wide ICT hub
Societal Challenges, Concretely: Graph Processing for Everyone

Oracle 1.2M followers, 132k employees
company/day: 40-60 posts, 500-700 comments

270M MAU
200+ avg followers
>54B edges

1.2B MAU 0.8B DAU
200+ avg followers
>240B edges

* SMEs in EU/NL = 60% gross value added, little to no ICT expertise
Scientific Challenges

How to massivize graph processing?

- Super-scalable, super-flexible, yet efficient graph-processing infrastructure
- End-to-end automation of large-scale graph processing
- Dynamic, compute- and data-intensive graph-processing workloads
- Evolving, heterogeneous hardware and software
- Strict performance, cost, energy, reliability, and fairness requirements
Massivizing Graph-Processing Systems

5’ — Pitch on Massivizing Graph-Processing Systems

5’ — Two Exemplary Steps Forward
- Benchmarking distributed or heterogeneous graph-processing systems
- Designing distributed and heterogeneous graph-processing systems

5’ — Towards a Taskforce on Data Science as a Service
What does a benchmark consist of?

• Four main elements:
  • data schema: defines the structure of the data
  • workloads: defines the set of operations to perform
  • performance metrics: used to measure (quantitatively) the performance of the systems
  • execution rules: defined to assure that the results from different executions of the benchmark are valid and comparable

• Software as Open Source (GitHub)
  • data generator, query drivers, validation tools, ...
Graphalytics, in a nutshell

• An LDBC benchmark
• Advanced benchmarking harness
• Diverse real and synthetic datasets
• Many classes of algorithms
• Granula for manual choke-point analysis
• Modern software engineering practices
• Supports many distributed/heterogeneous platforms

http://graphalytics.ewi.tudelft.nl/
https://github.com/tudelft-atlarge/graphalytics/
What is the performance of

**General Challenges**

- Performance Metrics
- Graph Diversity
- Algorithm Diversity

**Challenges for evaluating GPU-enabled systems**

- In-memory graph formats
- Optimization techniques
- GPU generations

---

Sample Result: BFS Algo on Amazon Data for all systems

**Initialization time** dominates total execution time

- **T-G**: 800 ms
- **T-H**: 800 ms
- **MG**: 1500 ms
- **M**: 2000 ms

Massivizing Graph-Processing Systems

5’ — Pitch on Massivizing Graph-Processing Systems

5’ — Two Exemplary Steps Forward
- Benchmarking distributed or heterogeneous graph-processing systems
- Designing distributed and heterogeneous graph-processing systems

5’ — Towards a Taskforce on Data Science as a Service
Existing Graph-Processing Systems: *Either Distributed or Heterogeneous*

- **Distributed CPU-based** systems cannot use additional computational power of accelerators
  - Oracle Labs
  - PGX

- **GPU-enabled** systems are (mostly) single-machine systems, cannot handle large-scale graphs
  - medusa-gpu
  - mapgraph
  - Gunrock
  - TOTEM
  - VertexAPI2

---

Our approach: 3 Families of Distributed and Heterogeneous (CPU+GPU) Graph-Processing Systems

• Can use both the CPUs and the GPUs of multiple machines.
• Explore the design space with a focus on partitioning.
• Design three families of systems with different partitioning architectures.
• Select and design promising policies for each family.
• Calculate the workload fraction for the CPU(s) and the GPU(s) based on profiling.

3 Families Explored: 2 Lessons Learned

1. There is no overall winner, but C-R is in general the worst.
2. Our new PG policy for Combined systems shows good performance.

- PageRank, 4 machines
- Also tried BFS and WCC

Promising Results for Distributed *and* Heterogeneous Graph-Processing Systems

Massivizing Graph-Processing Systems

5’ — Pitch on Massivizing Graph-Processing Systems

5’ — Two Exemplary Steps Forward

• Benchmarking distributed or heterogeneous graph-processing systems
• Designing distributed and heterogeneous graph-processing systems

5’ — Towards a Taskforce on Data Science as a Service
Take-Home Message

(Cloud computing +) Big Data

Graph processing as example

Important New Challenges
1. Benchmarking
2. Distributed Heterogeneous Systems
3. …
Next? A Taskforce on Data Science as a Service

Identify industry needs in the Netherlands
• Stakeholders: datacenter operators, ICT designers, ICT analysts, ICT researchers, governance, ICT media

Establish a joint research agenda, between fundamental and applied research
• Groundbreaking ideas for important challenges
• Prototypes and Proof-of-Concepts, not only ideas

Build a solid, pragmatic collaboration
• Relevant recommendations for relevant problems
• Embedding of human resources, joint networking, etc.
Contact Our Team

Collaboration or discussion about:

• Leveraging open-source / open-access cloud computing and big data systems
• Distributed and heterogeneous graph-processing

A.iosup@tudelft.nl
+31-15-2784433
@Alosup
http://pds.twi.tudelft.nl/~iosup/
https://www.linkedin.com/in/aiosup

PDS Group, Faculty EEMCS, TU Delft
Room HB07.050, Mekelweg 4, 2628CD Delft
Recommended Reading

Elastic Big Data and Computing

Time-Based Analytics

Graph Processing / Benchmarking
Disclaimer: images used in this presentation obtained via Google Images.

• Images used in this lecture courtesy to many anonymous contributors to Google Images, and to Google Image Search.

• Many thanks!