Data Intensive Computing

Adam Carter
EPCC, The University of Edinburgh
EUDAT Training Coordinator

APARSEN Advanced Practitioners Course
Glasgow, July 2013

EUDAT

Introduction

- Data Intensive Computing and its Relationship to Data Curation/Preservation
- The talk is slightly tangential…
  - *but* there are many overlaps in the subjects, technologies and aims of the data preservation and data intensive computing/research
- Data is being preserved *so that it can be re-used*

EUDAT

Data Preservation Lifecycles

- Most data preservation lifecycles include that idea that data is “unarchived” or “awakened”
- In the future data is likely to be more active more of the time
  - A good thing:
    - Active curation
    - Annotation, tagging and linking
  - A challenge:
    - Can best practices of archiving be maintained on a “live” dataset?
- You could certainly still look on this as a different use-case, but in this case it’s still important to understand what is going on elsewhere

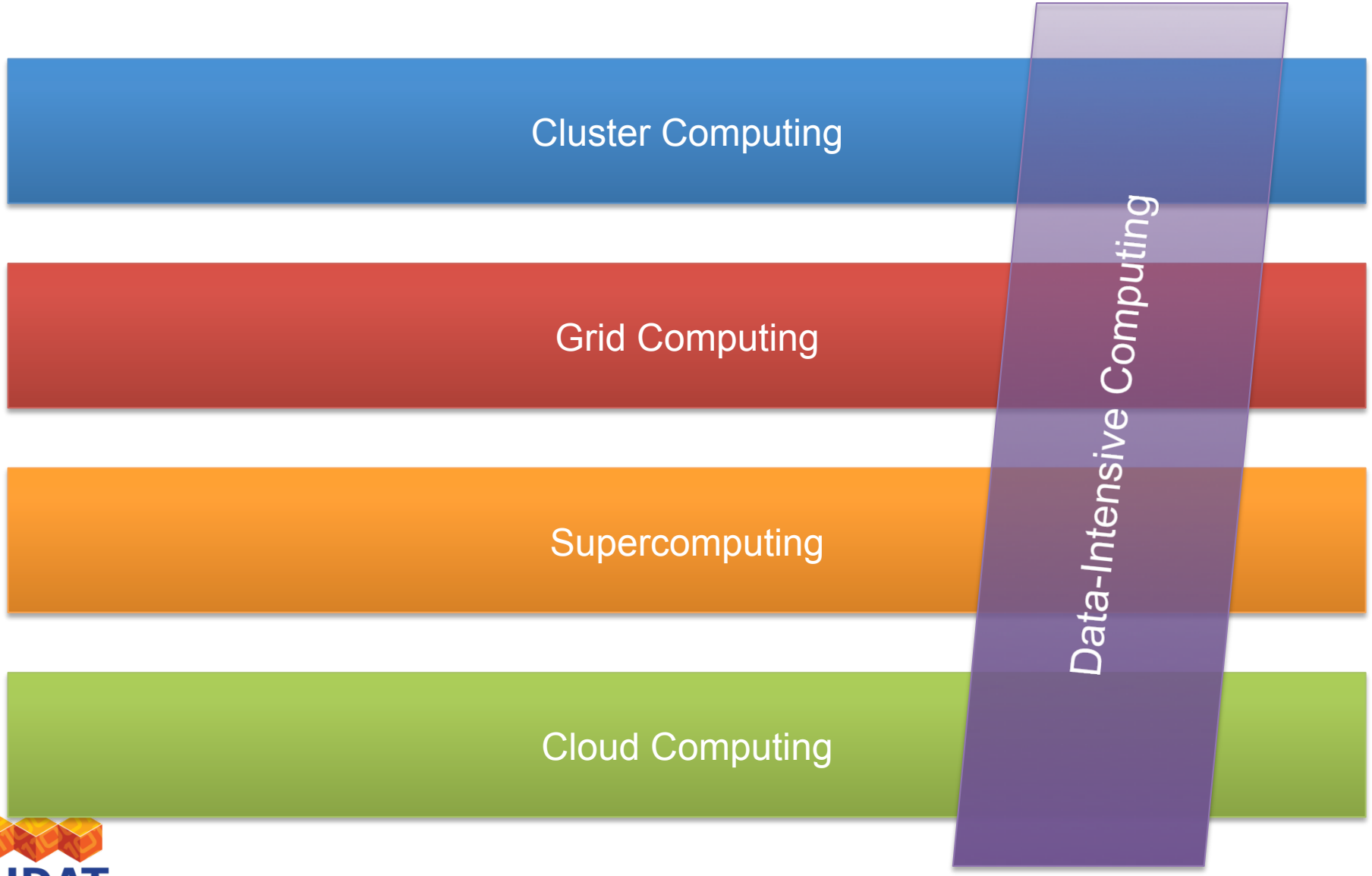
EUDAT

Data Intensive Computing

Computing applications which devote most of their execution time to computational requirements are deemed compute-intensive and typically require small volumes of data, whereas computing applications which require large volumes of data and devote most of their processing time to I/O and manipulation of data are deemed data-intensive. – **Wikipedia**

- My working definition:
  - *I/O-bound computations*
- Data is (generally) too big to fit in memory
  - Efficient disk access is required to get the data to the CPU on time
  - Having the data in the right place at the right time is *vital*

EUDAT


Cluster Computing
Grid Computing
Supercomputing
Cloud Computing
Data-Intensive Computing

EUDAT

The Role of Data Infrastructures in Data Intensive Computing

- Traditionally, we bring the data to the compute
- In the future, we'll want to bring the compute to the data
  - So where is the data?
  - More than likely it's in a repository…
  - Maybe an "archive"…

EUDAT

What will the data in the archive look like?

- Files?
- Rows & Tables in a Relational Database?
- Tuples in a Triple Store?

EUDAT

How might you bring in compute?

- As (relational) database queries (SQL)
- As queries against an RDF store (SPARQL)
- As VMs which can mount local disks
- As scripts or executables that you allow a user to run
- As services that you as a data service offer with some kind of API (e.g. as a web service)

EUDAT

- These are the approaches that will need to be offered by “repositories” holding large amounts of “live” data.
  - Many will probably also be relevant to archives
- How can a user get the information back out of the archive?
  - As complete files?
  - Over the Internet (and your network!)?

EUDAT

Back to the Compute…

- Need to understand the performance of your computations *and your data transfers*
- Do you know how fast your program runs?
- Do you know if it's spending all of its time on compute or if it's spending its time waiting for data?
- Where is your data bottleneck?

- *Benchmarking is key*

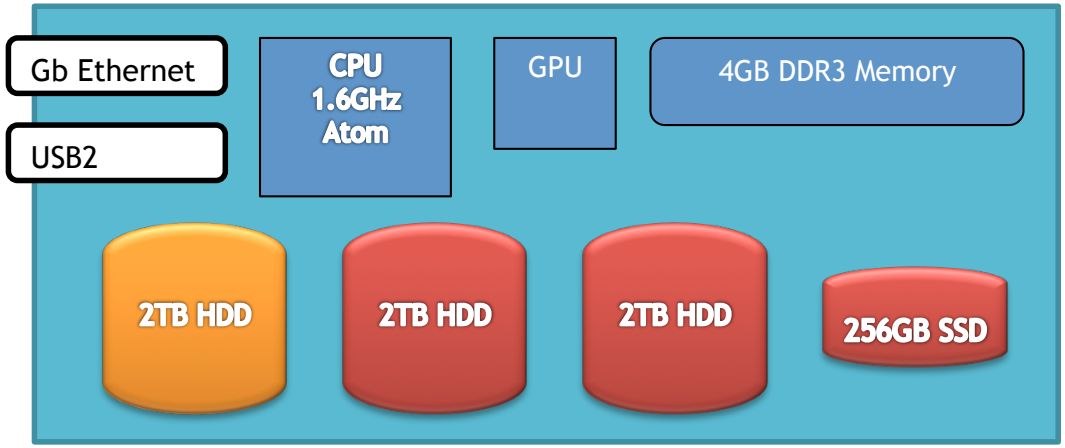
EUDAT

Amdahl's *Other Laws*

- Gene Amdahl's quantification of the balance required for data-intensive applications:
  - One bit of I/O per compute cycle
  - Memory Size (bytes) / instructions per second = 1
  - One IOop per 50,000 instructions

EUDAT

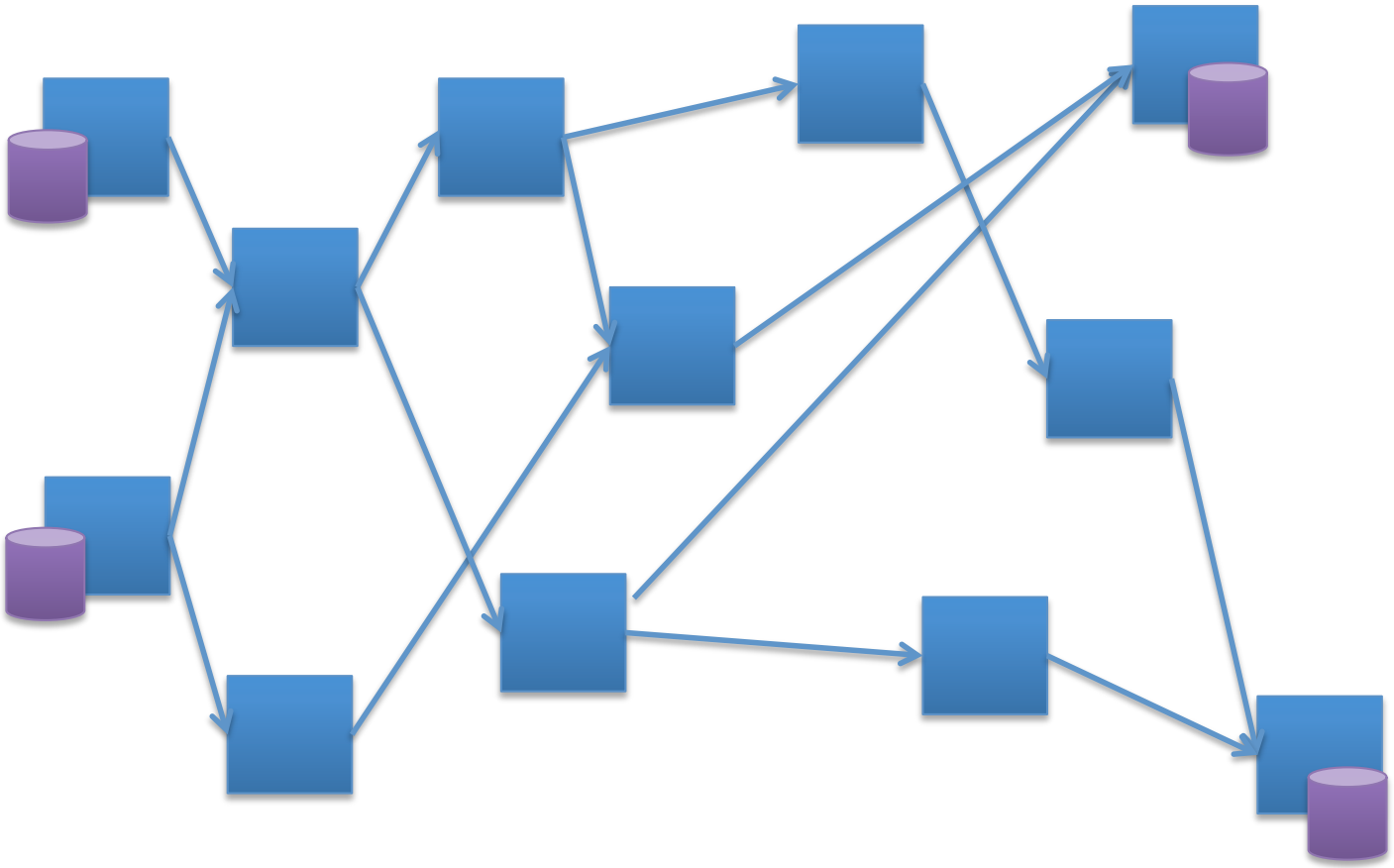
A Data Intensive Computer



× 120

EUDAT

…or use whole datacentre(s)



EUDAT

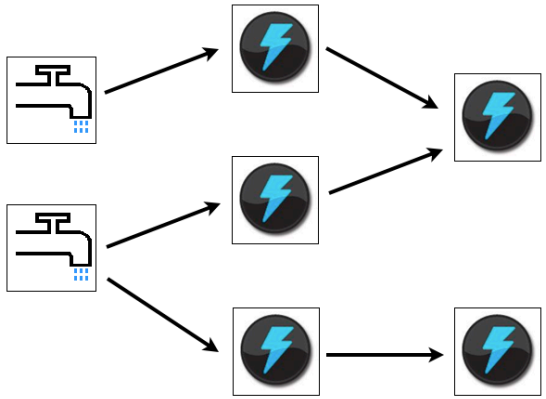
Making best use of a machine designed for data-intensive computing

- Work on streams of data, not files
  - Not (so easily) searchable
  - Not (so easily) sortable
  - Not all programs can benefit from this approach
    - and those that can, might require work
- Use multiple threads and asynchronous I/O
- If you're using files, use a library that does some of the hard work for you, e.g. MPI-IO

EUDAT

Some Data Intensive Technologies

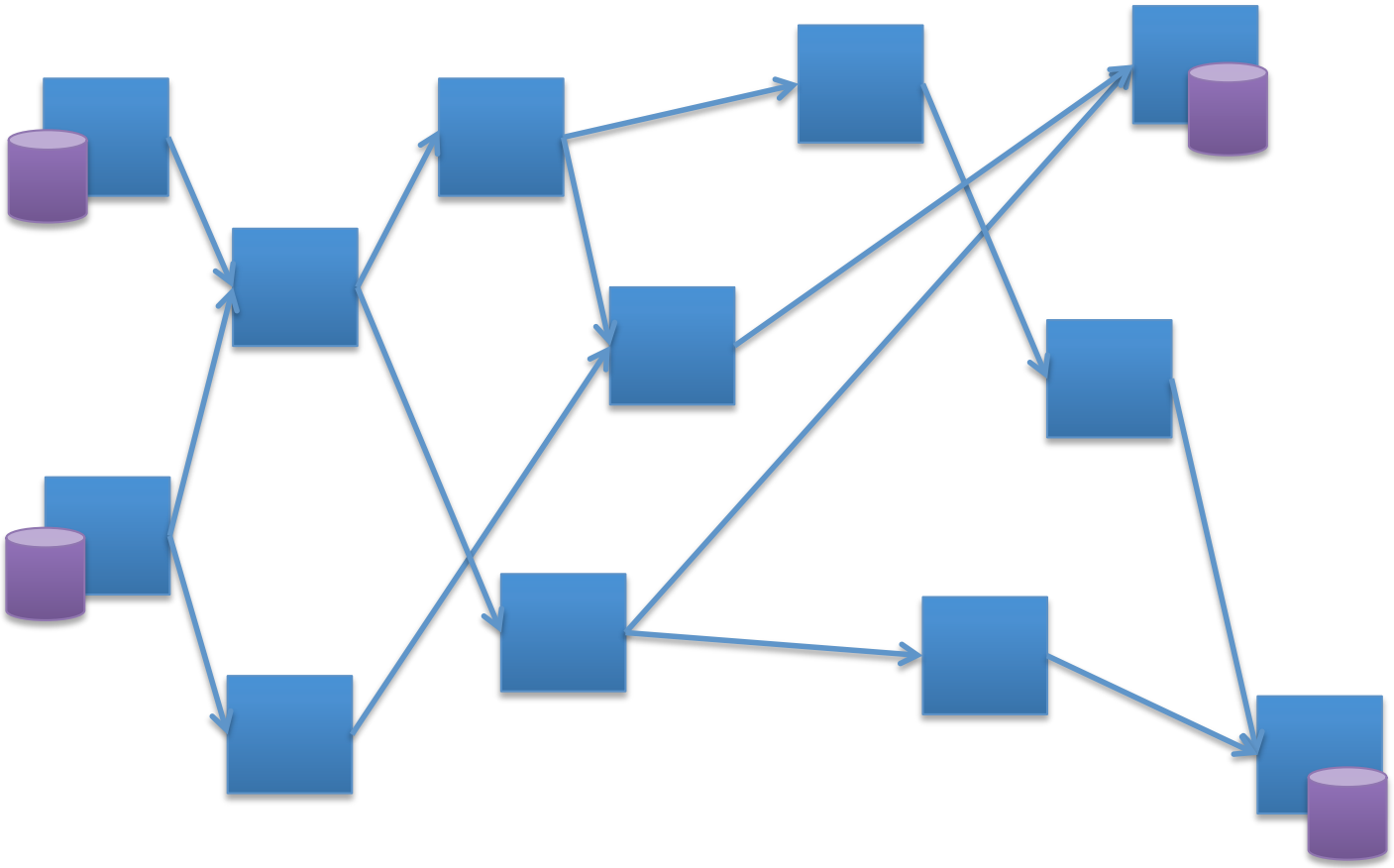
- MapReduce/Hadoop (described earlier in the week)
- Storm (http://storm-project.net)



  - Low latency
  - Real-time
  - No writes to disk at intermediate stages
  - Reportedly not quite such good scalability in terms of throughput

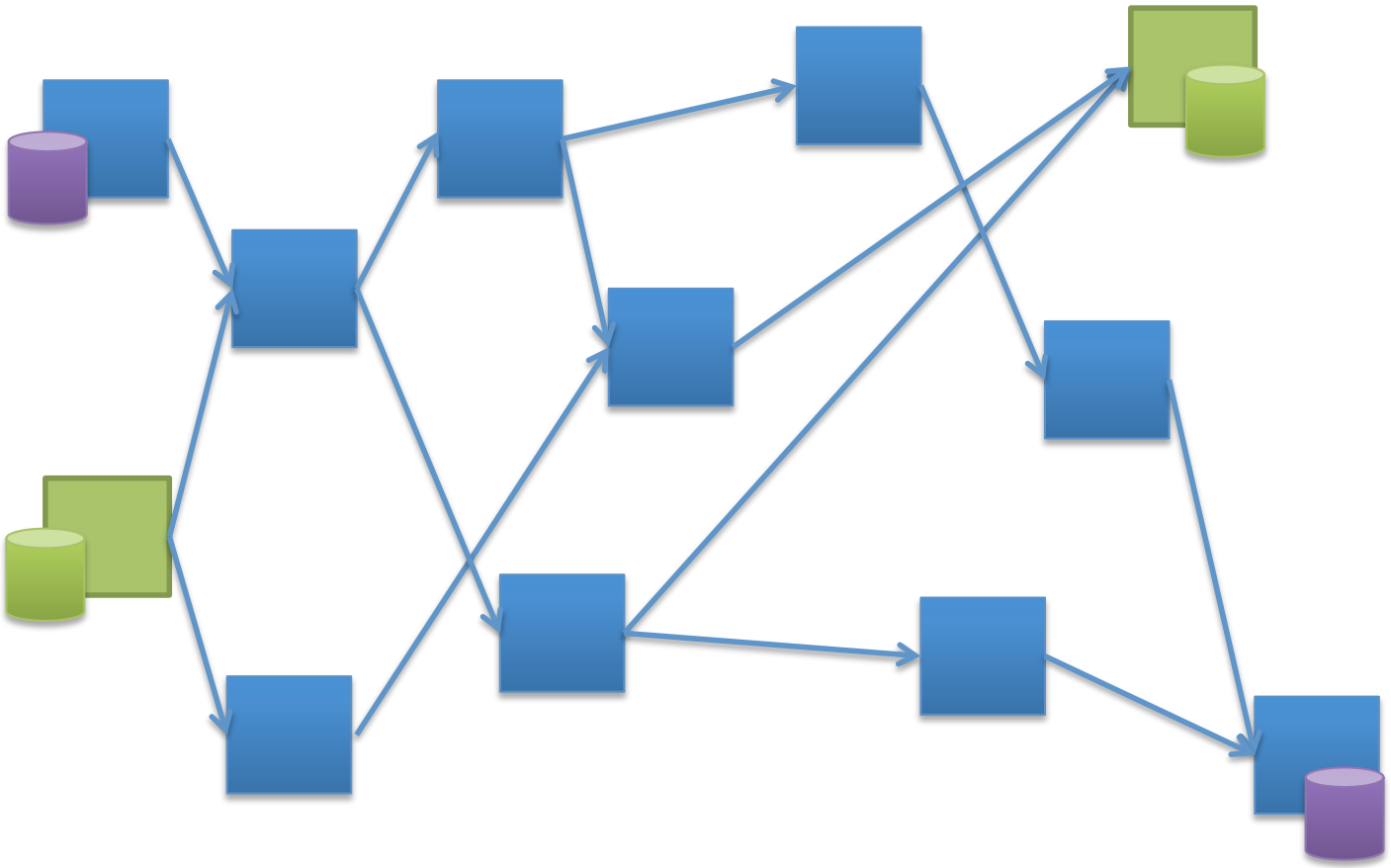
EUDAT

…or use whole datacentre(s)



EUDAT

…or the internet?



EUDAT

Conclusions

- Data-Intensive is a new(ish) kind of computing
  - necessitated by the huge amounts of data
  - and offering new opportunities
- Need to think about new ways of doing computing
  - It's usually parallel computing, but not "traditional HPC"
- Matters for data preservation. Either:
  - you're preserving huge amounts of data that need to be easily reused
  - you need to process large amounts of data to do a meaningful reduction so that the stored data retains its value

EUDAT