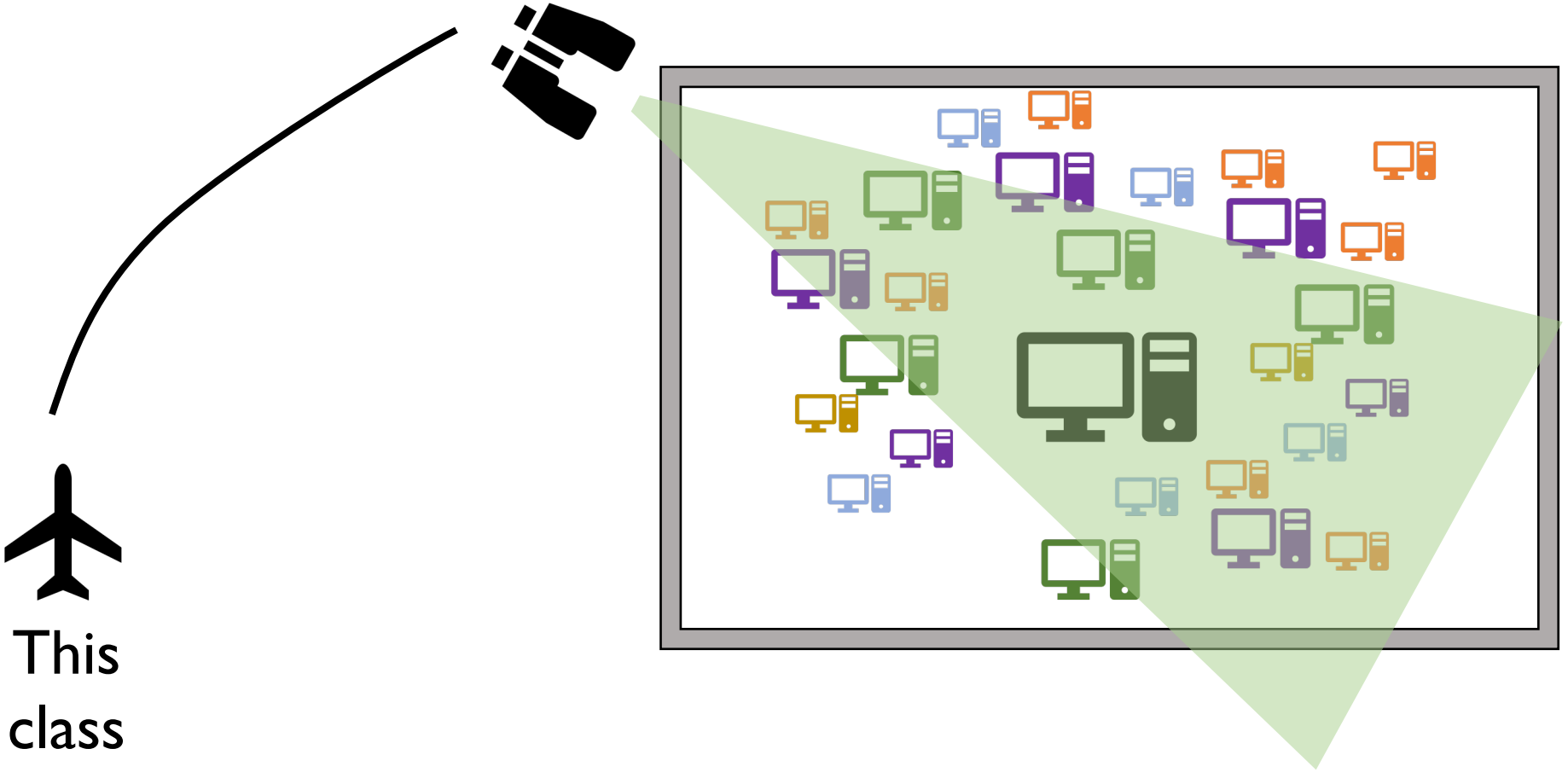
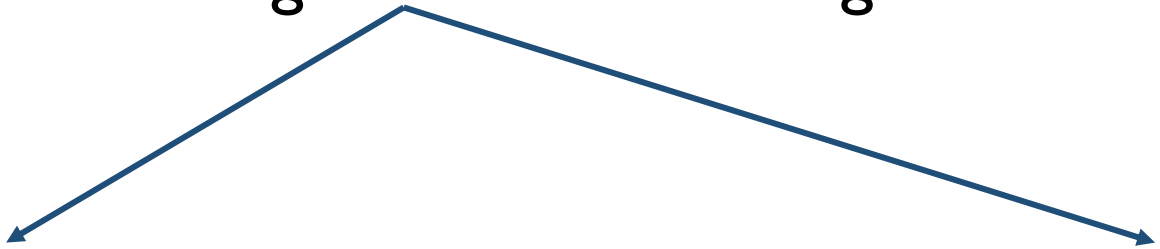
6) Wrap up


This
class

What is the proportion of:

Recap

✓ Parallelism = Large data/tasks + Large number of resources



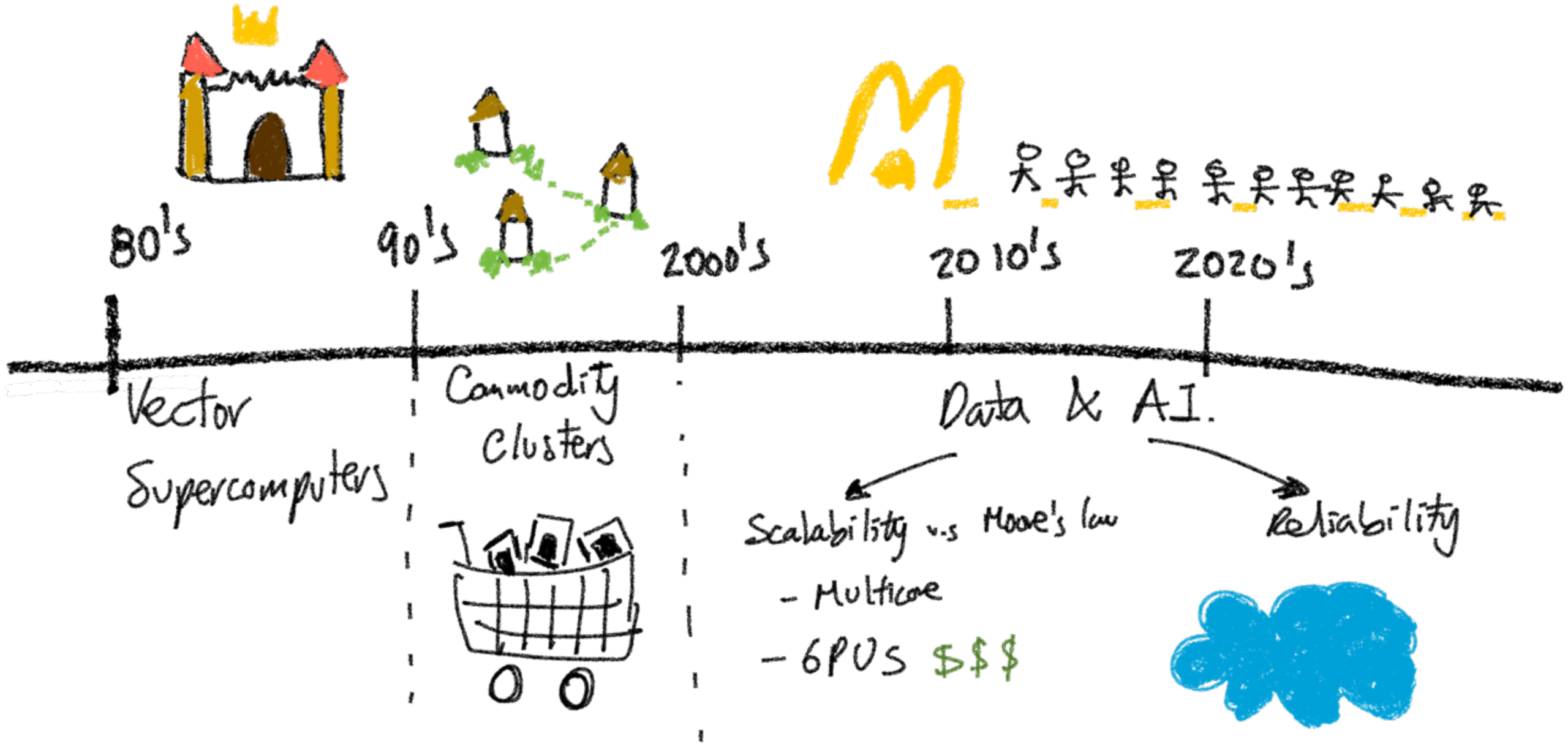
Data Parallelism

How to distribute, transform, and aggregate data?

Task Parallelism

How to distribute and execute tasks given data dependencies?

✓ Constrains: Problem based + Resources based: Memory & Communication


80's
90's
2000's
2010's
2020's
Vector
Supercomputers
Commodity
Clusters
Data & AI.
Scalability v.s Moore's law
- Multicore
- GPUs $$$
Reliability

Reality of Scientific Computation today

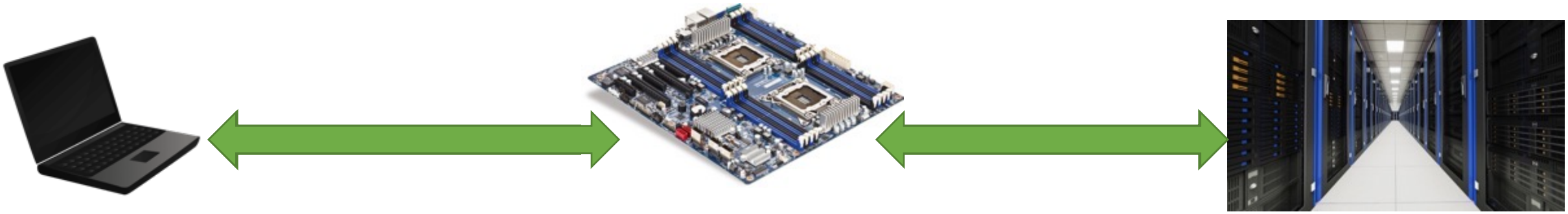
Heterogenous

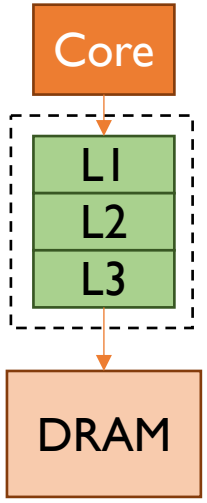
Continuous change

Local & In demand

Portability & Reliability

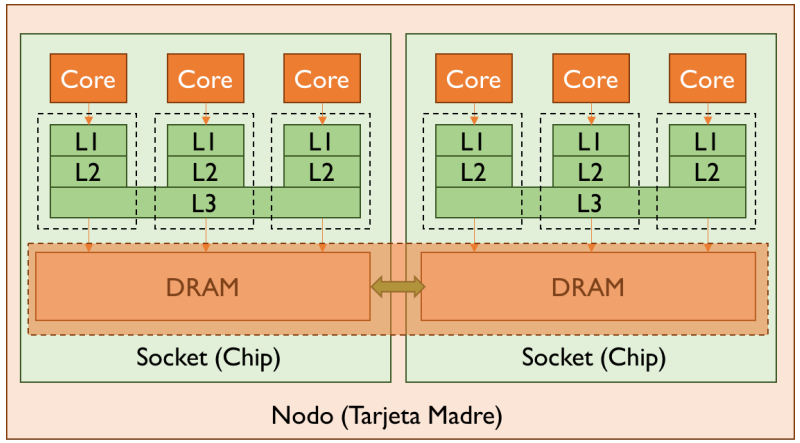
Balanced communication is more challenging than ever!





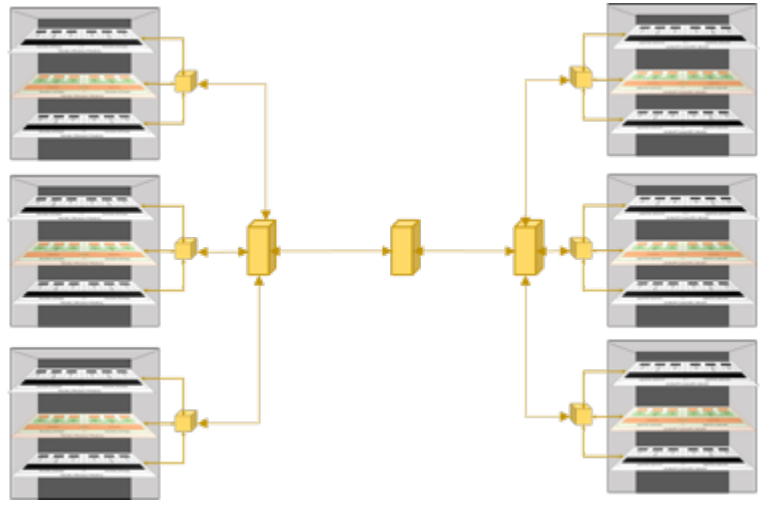
Single Processor

Compiler



Shared Memory

OpenMP



Distributed Memory

MPI

Recap

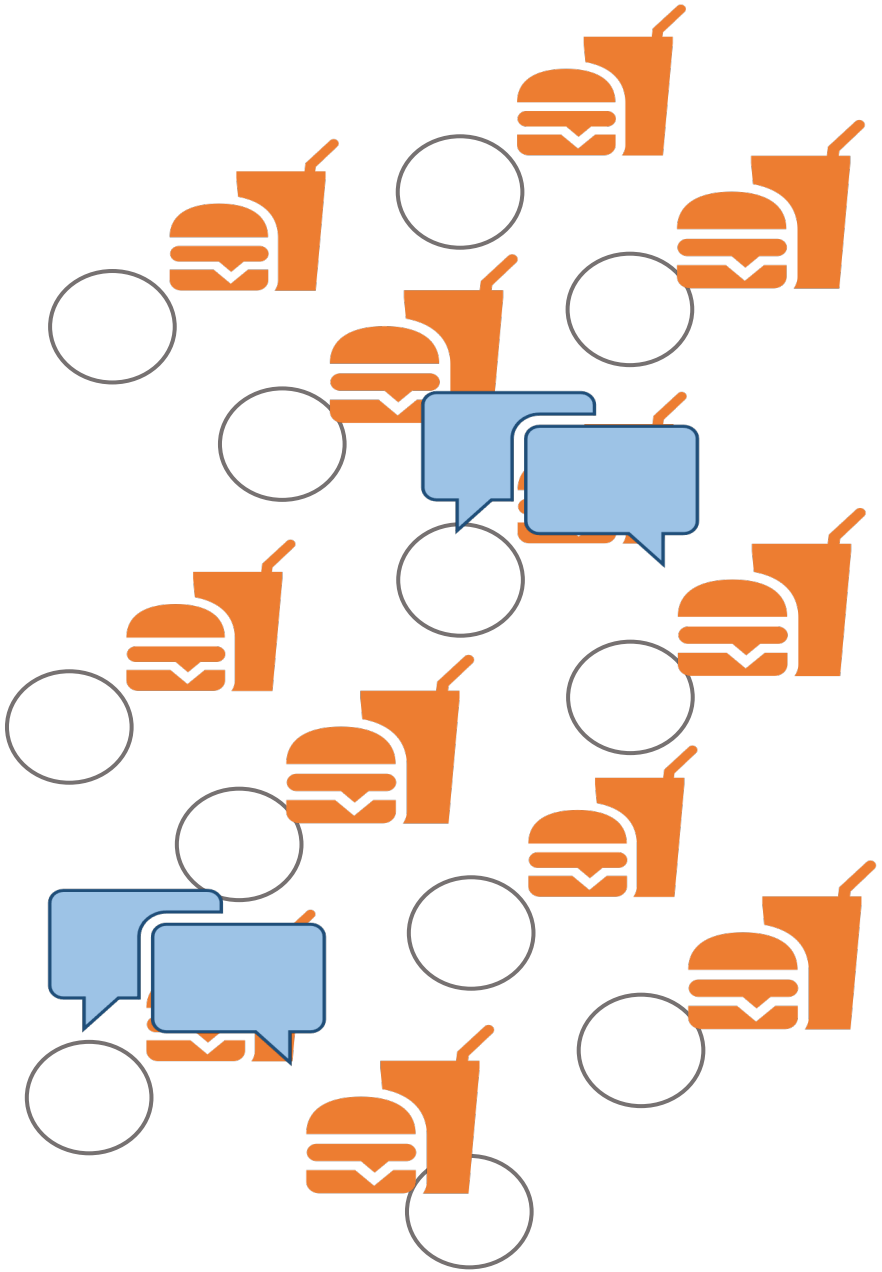
Communication is significantly more expensive than computation

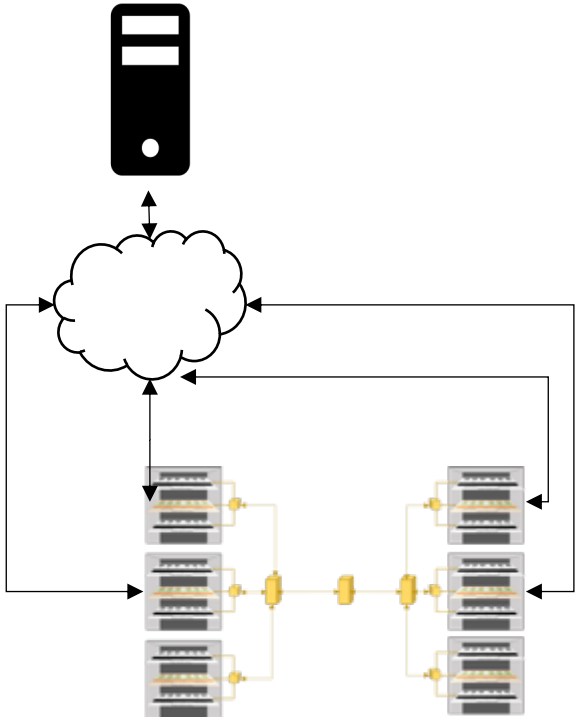
Memory access

Shared Memory

Distributed Memory

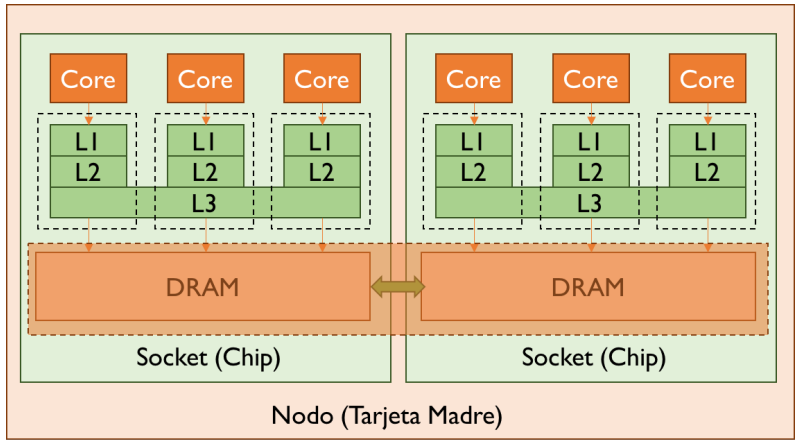
2000-10's Data Centers





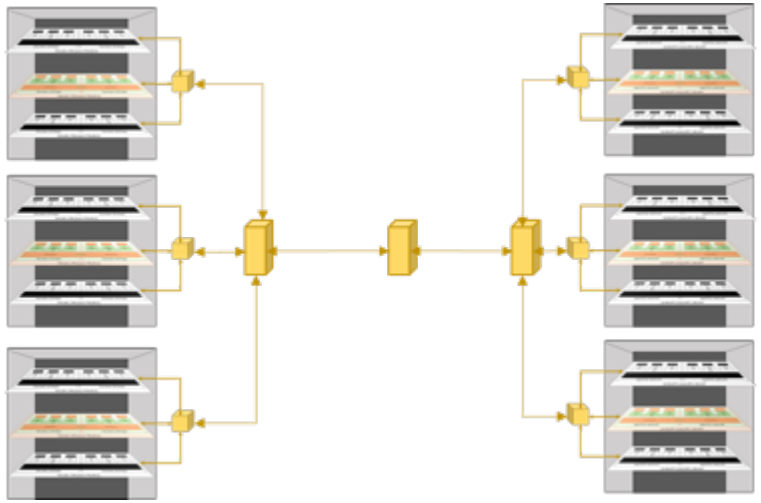
Global Name Space

Spark



Shared Memory

OpenMP



Distributed Memory

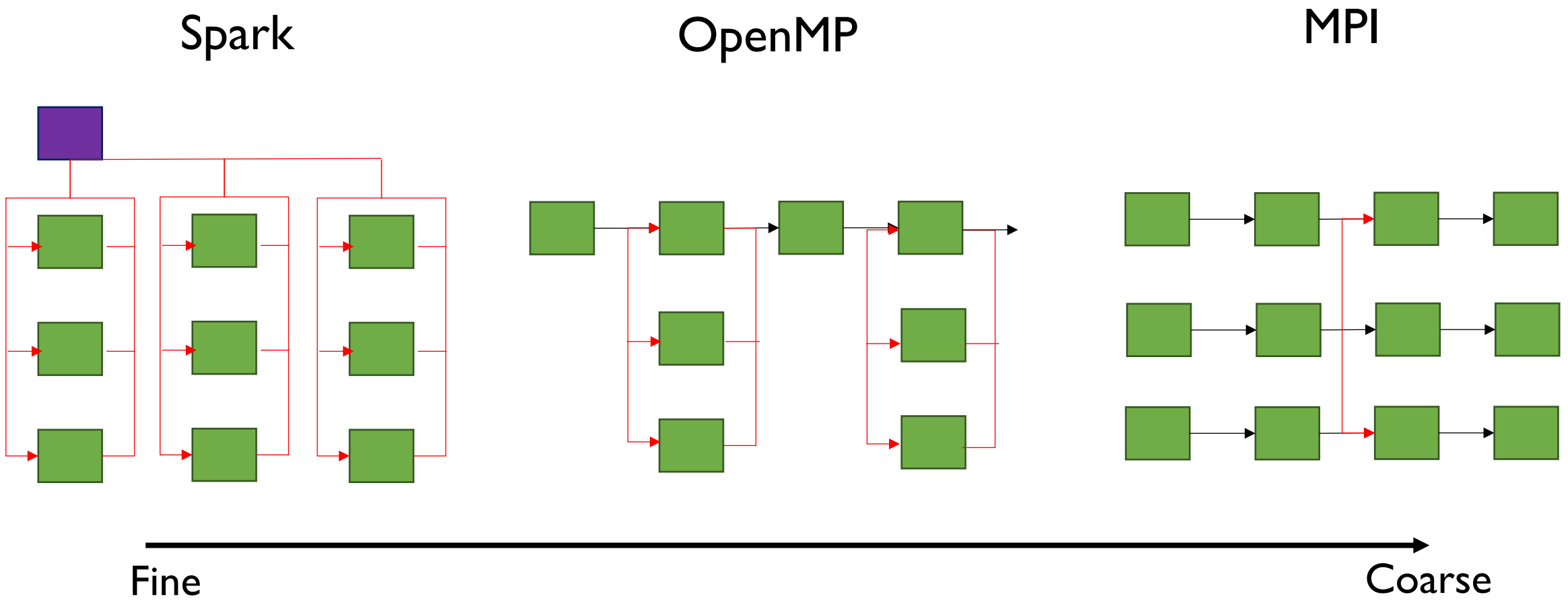
MPI

Spark

OpenMP

MPI

Granularity



	Spark	OpenMP	MPI
Goal	Manage large data sets in clusters with failures	Compiler directives to manage threads and shared memory	Standard for message-passing library interface
Principle	Functional Programming	Fork and Join	Distributed Memory
Tools	Transformations/Actions HDFS/RDDs	Parallel region: for/sections	Communication: 1-1, global, one sided
External	Cluster Manager like YARN		Scheduler: SLURM