Hadoop vs Distributed Systems

- Hadoop is a large-scale distributed batch processing infrastructure.
- While it can be used on a single machine, its true power lies in its ability to scale to hundreds or thousands of computers, each with several processor cores.
- Hadoop is also designed to efficiently distribute large amounts of work across a set of machines.

•Actually Hadoop is built to process "web-scale" data on the order of hundreds of gigabytes to terabytes or petabytes.

•Hadoop includes a distributed file system which breaks up input data and sends fractions of the original data to several machines in your cluster to hold.

•This results in the problem being processed in parallel using all of the machines in the cluster and computes output results as efficiently as possible.

CHALLENGES AT LARGE SCALE

- Performing large-scale computation is difficult.
- Whenever multiple machines are used in cooperation with one another, the probability of failures rises.
- In a distributed environment, however, partial failures are an expected and common occurrence.
- Networks can experience partial or total failure if switches and routers break down.

- Data may not arrive at a particular point in time due to unexpected network congestion.
- Individual compute nodes may overheat, crash, experience hard drive failures, or run out of memory or disk space.
- Data may be corrupted, or maliciously or improperly transmitted.
- Multiple implementations or versions of client software may speak slightly different protocols from one another.

- Clocks may become desynchronized, lock files may not be released, parties involved in distributed atomic transactions may lose their network connections part-way through
  - In each of these cases, the rest of the distributed system should be able to recover from the component failure or transient error condition and continue to make progress.
  - Providing such resilience is a major software engineering challenge.

Distributed System

- Different distributed systems specifically address certain modes of failure, while worrying less about others.
- Other distributed systems make different trade-offs, as they intend to be used for problems with other requirements (e.g., high security).

Hadoop

- Hadoop provides no security model, nor safeguards against maliciously inserted data
- it is designed to handle hardware failure and data congestion issues very robustly

Other Factors

- Management of compute resources. The compute hardware has finite resources available to it. The major resources include:
  - –Processor time
  - –Memory
  - –Hard drive space
  - –Network bandwidth
  - •Synchronization
- To be successful, a large-scale distributed system must be able to manage the above mentioned resources efficiently.

Hadoop

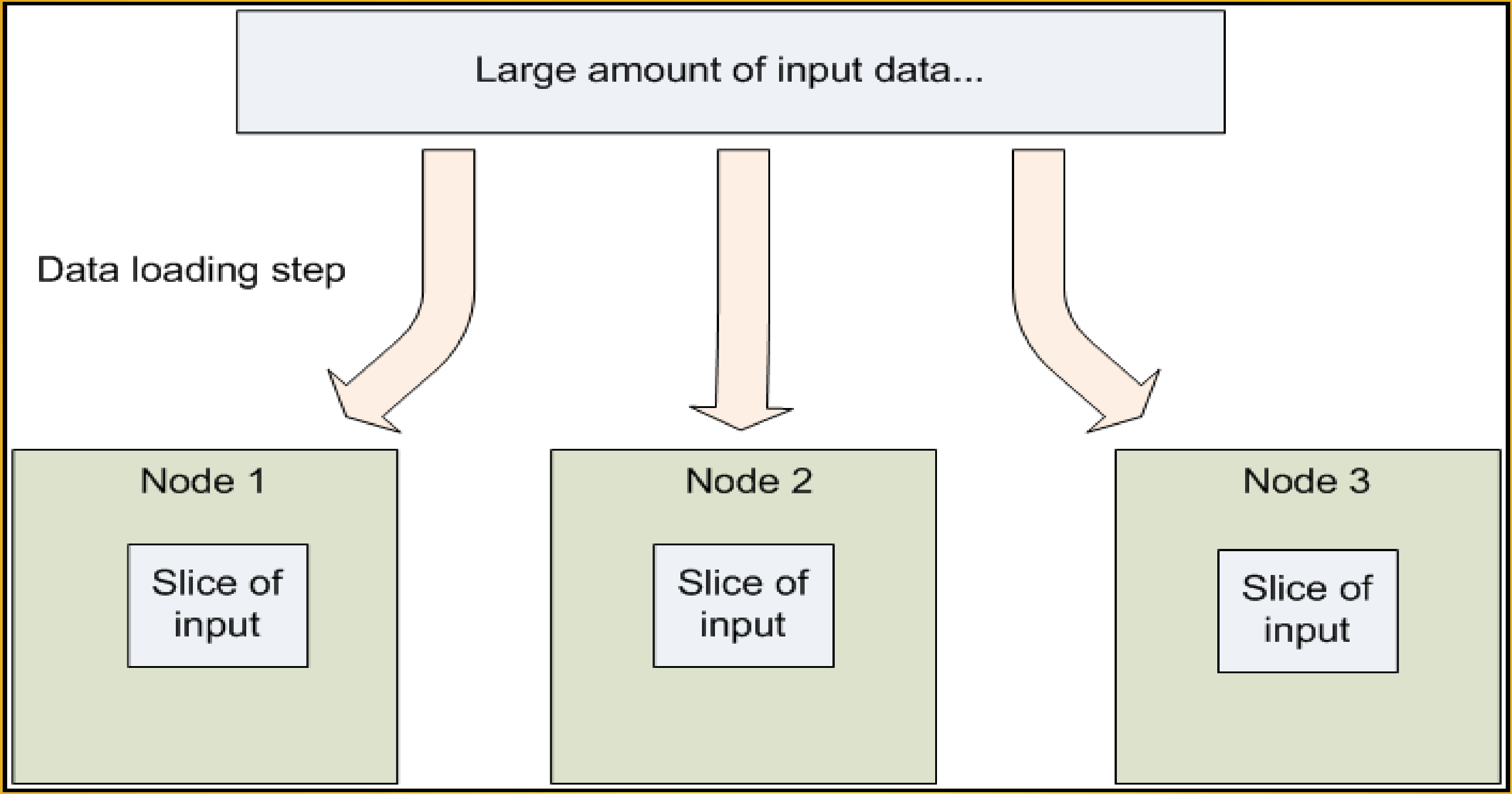
In Short:

- Hadoop unique is its **simplified programming model** which allows the user to quickly write and test distributed systems, and its **efficient, automatic distribution of data and work across machines** and in turn utilizing the underlying parallelism of the CPU cores.

More Factors Distinguishing- Hadoop from other systems - *Data Distribution*

- Hadoop Distributed File System (HDFS) will split large data files into chunks which are managed by different nodes in the cluster.
- In addition to this each chunk is replicated across several machines.
- An active monitoring system then re-replicates the data in response to system failures

- Data is conceptually **record-oriented** in the Hadoop programming framework.
- Hadoop framework then schedules these processes in proximity to the location of data/records using knowledge from the distributed file system.
- As most data is read from the local disk straight into the CPU, alleviating strain on network bandwidth and preventing unnecessary network



Data is distributed across nodes at load time

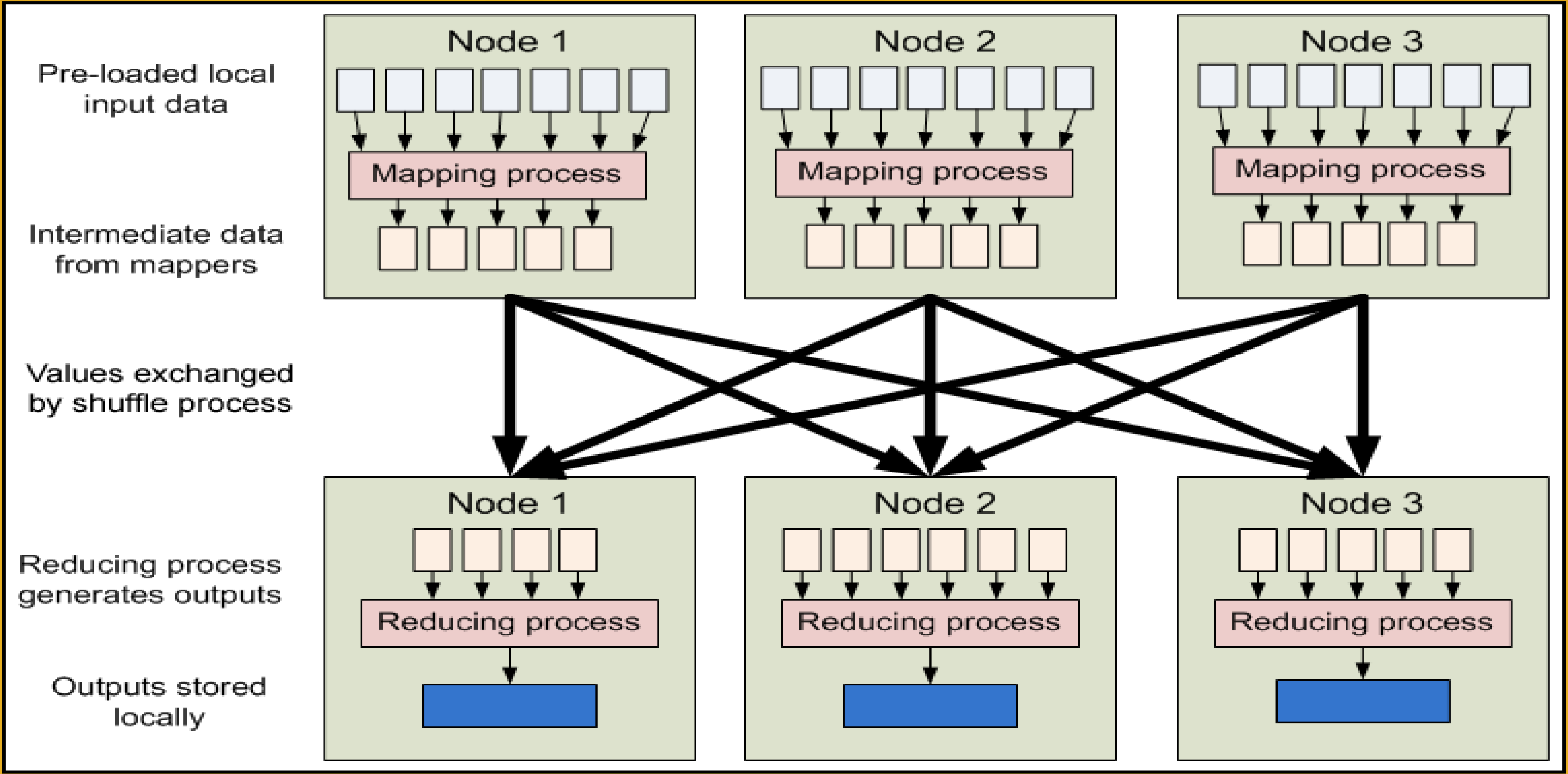
MAPREDUCE: ISOLATED PROCESSES

•Hadoop limits the amount of communication which can be performed by the processes, as each individual record is processed by a task in isolation from one another.

•Hadoop will not run just any program and distribute it across a cluster.

•Programs must be written to conform to a particular programming model, named "MapReduce.“

•In MapReduce, records are processed in isolation by tasks called *Mappers*. The output from the Mappers is then brought together into a second set of tasks called *Reducers*



Mapping and reducing tasks run on nodes where individual records of data are already present.

- Communication in Hadoop is performed *implicitly,* however in contrast to more conventional distributed systems where application developers explicitly marshal byte streams from node to node over sockets or through MPI buffers.
- By restricting the communication between nodes, Hadoop makes the distributed system much more reliable

•Individual node failures can be worked around by restarting tasks on other machines

FLAT SCALABILITY

- Hadoop in contrast to other distributed systems is its flat scalability curve.
- As the effort of coordinating work among a small number of machines may be better-performed by distributed systems, the price paid in performance and engineering effort (when adding more hardware as a result of increasing data volumes) increases non-linearly.

- A program written in distributed frameworks other than Hadoop may require large amounts of refactoring when scaling from ten to one hundred or one thousand machines.
- After a Hadoop program is written and functioning on ten nodes, very little--if any--work is required for that same program to run on a much larger amount of hardware.