主讲简介

- 郭宁
- 数据处理，4 年左右（2012年初起）
- 涉及领域：
    - 数据挖掘算法
    - 数据平台架构
- 论文：
    - *Soft-CsGDT: soft cost-sensitive Gaussian decision tree for cost-sensitive classification of data streams*, In Proc. of BigMine Workshop of ACM SIGKDD, 11-14 August 2013, Chicago, IL, USA.

海量数据处理

——关键思想

郭宁@美团

2016/10/14

guoning.gn@gmail.com

目录

海量数据特征

- 背景：
    - 海量数据处理之前，就有数据处理
    - 海量数据处理，有什么独特之处?
- 海量数据，特点：
    - 数据量大：远超单节点存储能力
    - 高速到达：存量数据是海量，增量数据也是海量的
    - 数据多样化：
        - 结构化特征不明显
        - 冗余信息多
        - 容错数据多：举例：系统升级，交易日志格式变更；服务降级，产生容错数据；

海量数据特征

201.158.69.116 - - [03/Jan/2013:21:17:20 -0600] fwf[-] tip[-] 127.0.0.1:9000 0.007 0.007 MX pythontab.com GET /html/test.html HTTP/1.1 "200" 2426 "http://a.com" "es-ES,es;q=0.8" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11"

187.171.69.177 - - [03/Jan/2013:21:17:20 -0600] fwf[-] tip[-] 127.0.0.1:9000 0.006 0.006 MX pythontab.com GET /html/test2.html HTTP/1.1 "aaa" 2426 "http://a.com" "es-ES,es;q=0.8" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11"

容错数据：本来应该为 http status code，但出现非法数据。

目录

- 数据特征：海量数据处理 *vs.* 普通数据处理
- 基本原理：海量数据处理，典型场景和关键过程
- 关键思想：
  - 分发计算
  - 机架感知
- 本质分析

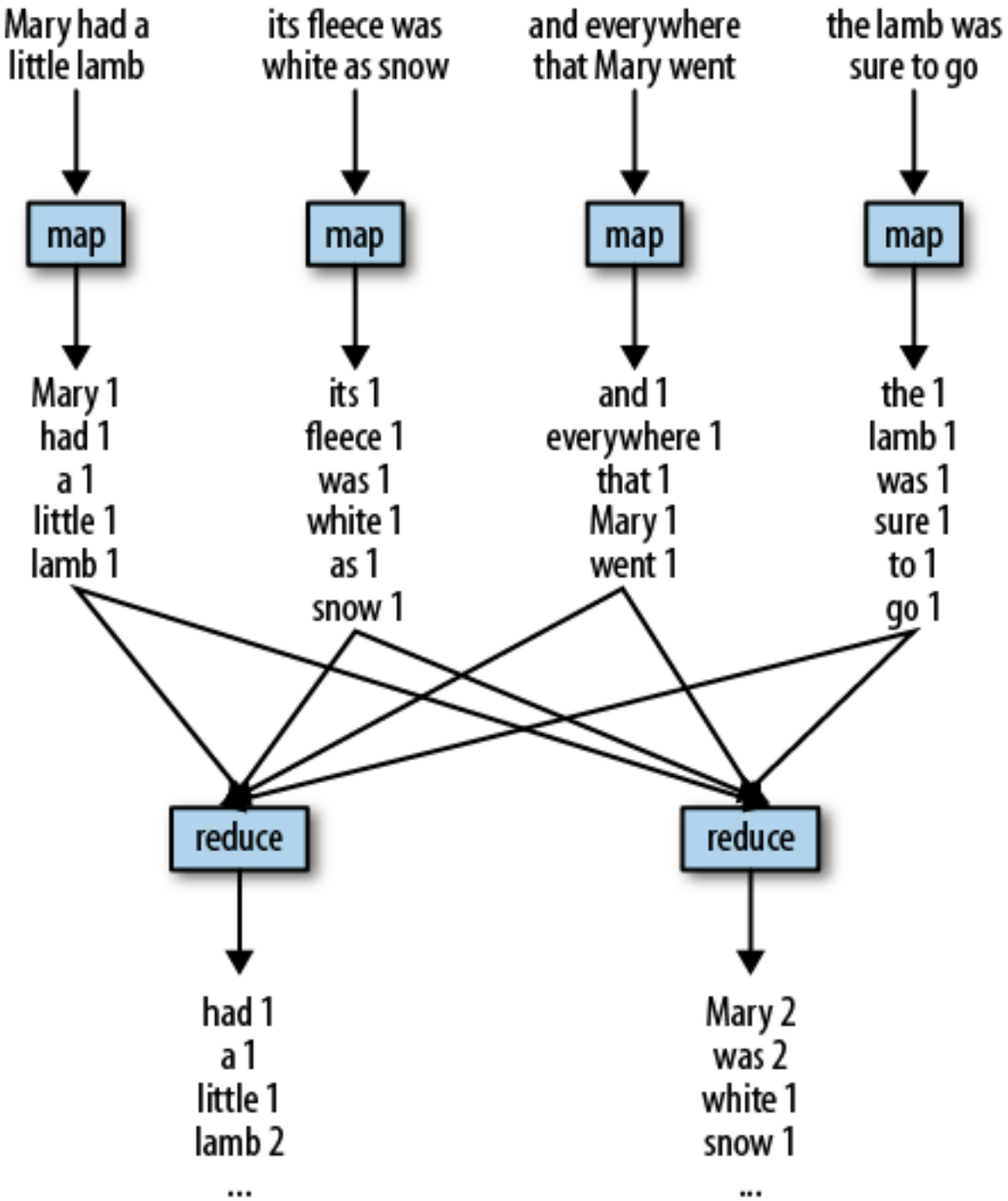
基本原理：典型场景

- 场景：
  - 词频统计
  - 统计右侧文本中，所有单词出现的频率。

Mary had a little lamb
its fleece was white as snow
and everywhere that Mary went
the lamb was sure to go.

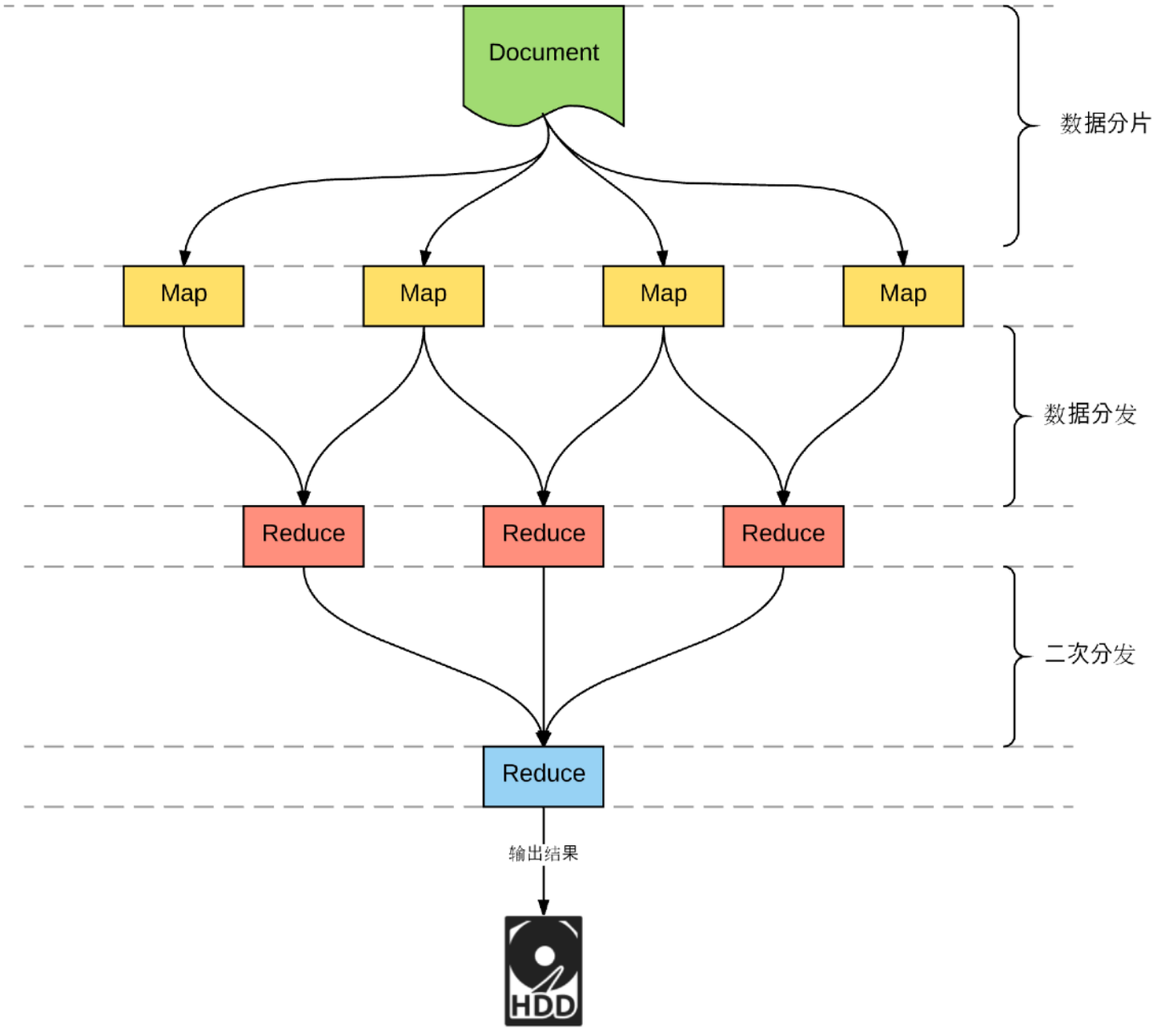
基本原理：典型场景

- 解决方式：常说的 MR
  - Map：
    - 输入？
    - 输出？
    - 作用？数据处理，快速的数据处理，通常处理逻辑简单
  - Reduce：
    - 输入？
    - 输出？输出的数据单元 <= 输入数据单元数量
    - 作用？对 map 结果进行聚合、收敛



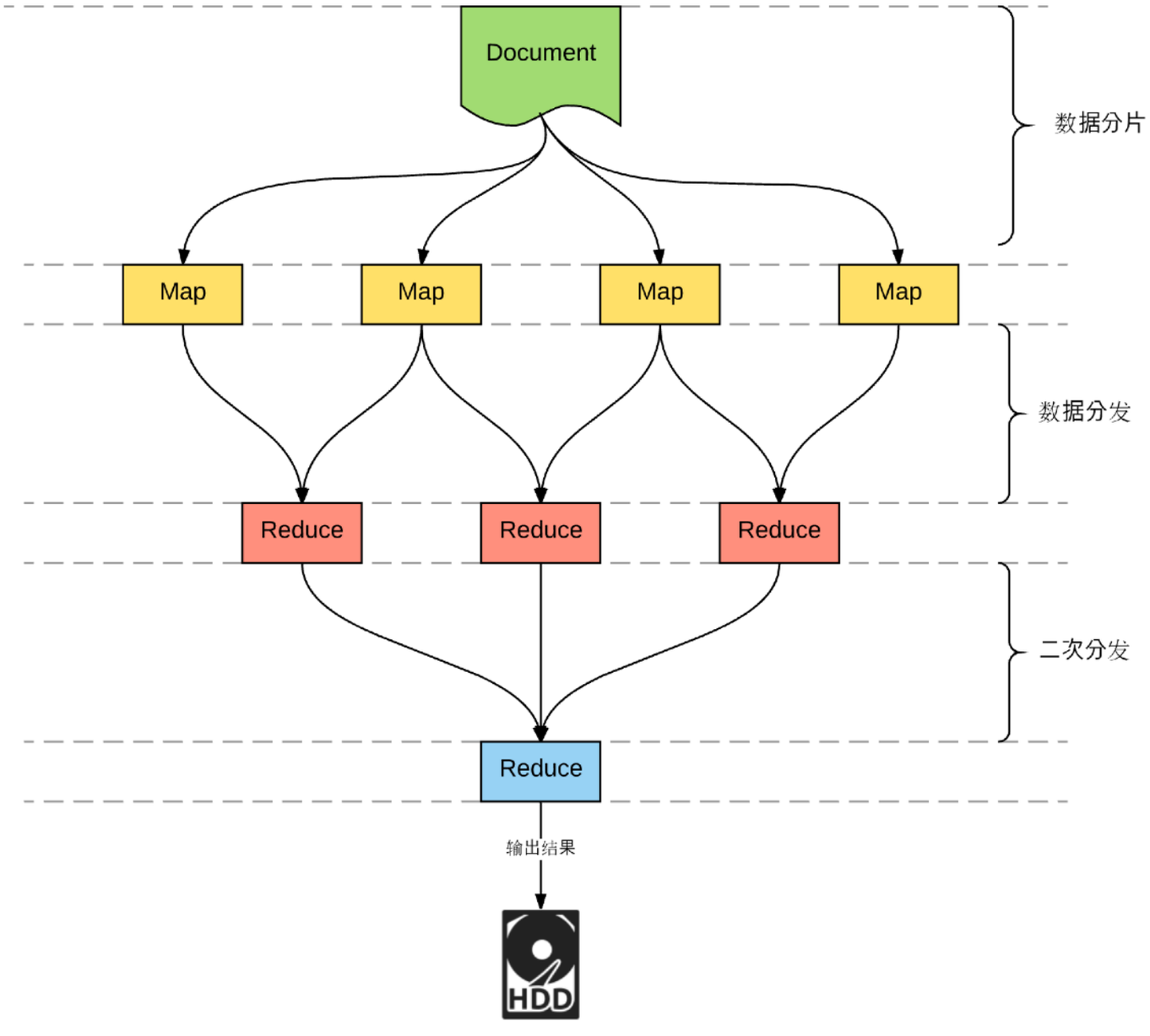
基本原理：关键过程

- 实际上，上述场景包含更丰富的内容：
  - 分片（Partition）：
    - Map 之前的数据切割、分片
  - 分发（Shuffle）：
    - Map 之后，将 Map 结果 shuffle 到特定的 Reduce 进行聚合
  - Multi-MapReduce：
    - 一次 MR 之后，通常还没有结束，会再次进行 Map 或 Reduce



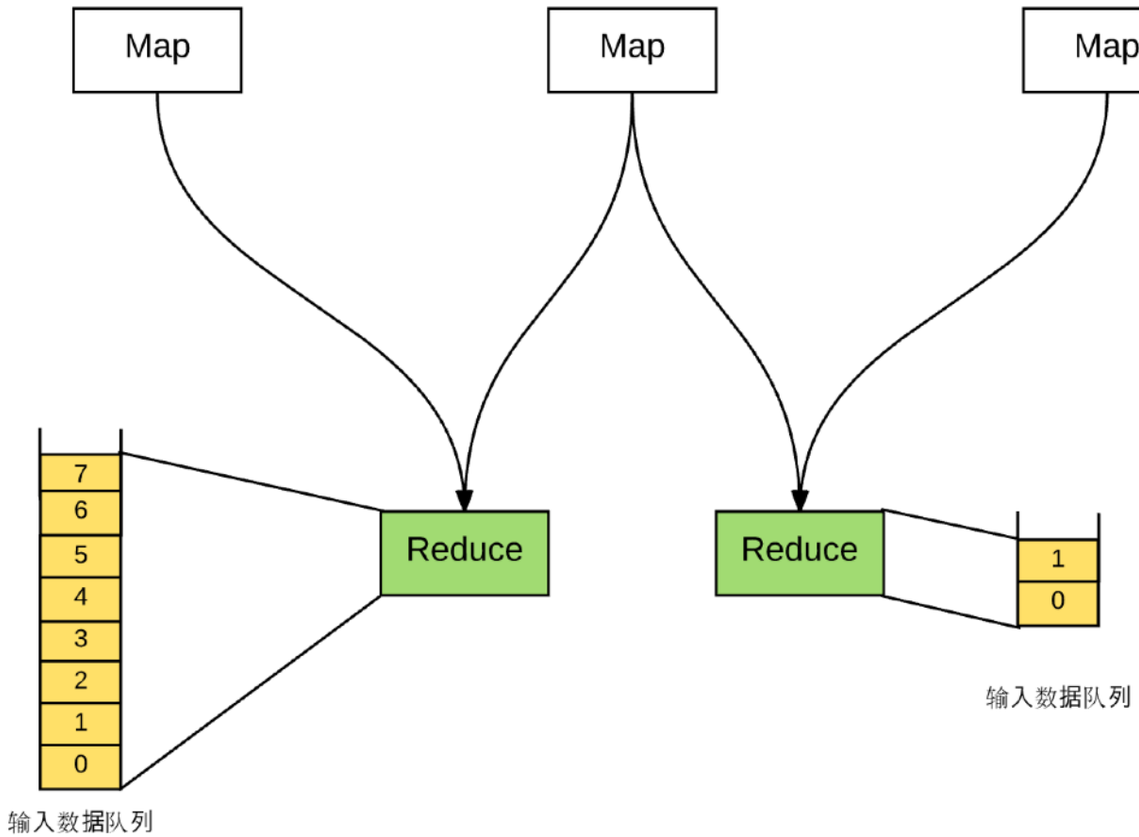
基本原理：关键过程

- 数据分片、数据分发之间的区别：
  - **数据分片**（partition）：
    - 第一次进行数据处理之前，并不知道数据的具体内容，因此，是简单的数据切割
  - **数据分发**（shuffle）：
    - 已经知道数据的具体内容，按照既定的业务逻辑，将特定的数据，送入特定的节点，数据分发中，要处理的典型问题：数据倾斜



基本原理：关键过程

- 数据分片、数据分发之间的区别：
  - **数据分片**（partition）：
    - 第一次进行数据处理之前，并不知道数据的具体内容，因此，是简单的数据切割
  - **数据分发**（shuffle）：
    - 已经知道数据的具体内容，按照既定的业务逻辑，将特定的数据，送入特定的节点，数据分发中，要处理的典型问题：数据倾斜



小结：基本原理

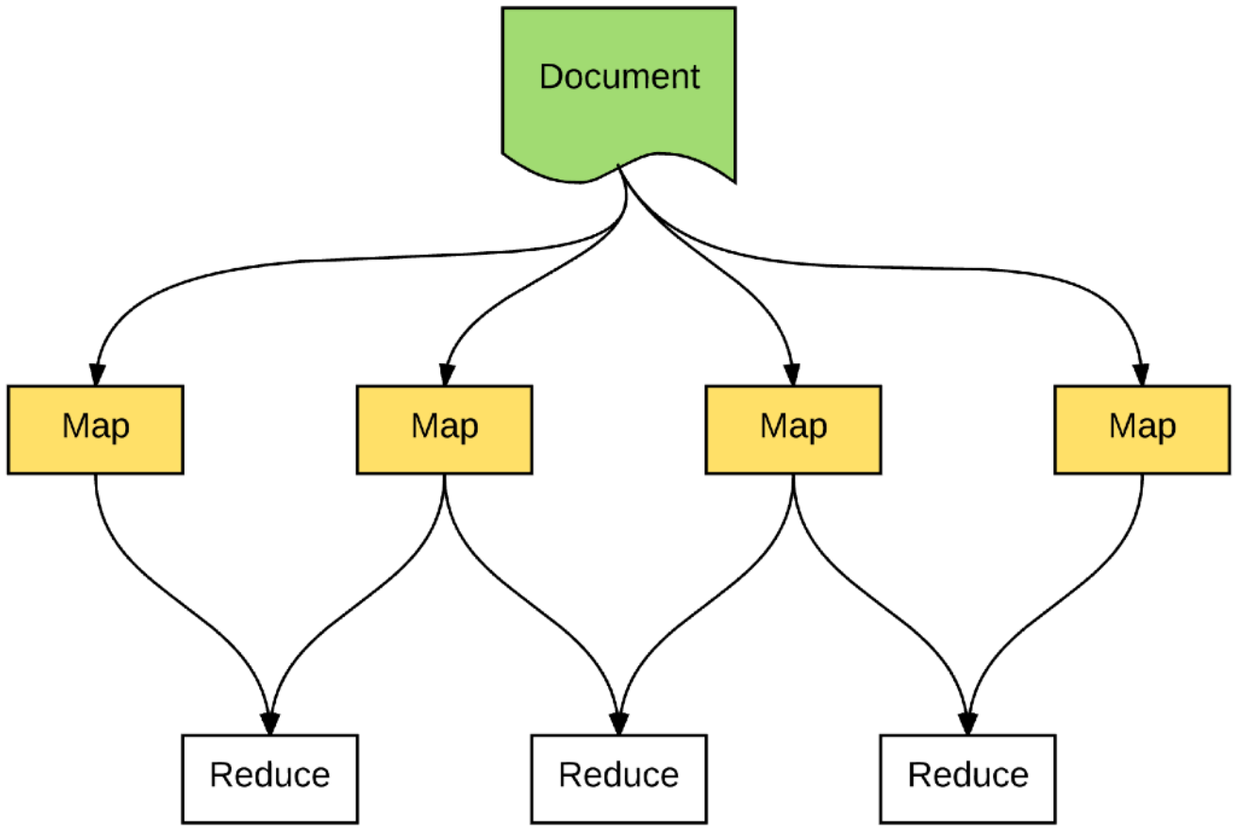
- 海量数据处理，基本原理：
    - 海量数据处理，涉及 4 个典型过程：
        - 数据分片
        - Map
        - Shuffle
        - Reduce
    - 海量数据处理，调优时，根据具体的业务场景、数据特征，减弱「数据倾斜」现象
    - Map-Reduce 是一种编程模型，实现方式可以有多种，代表性的实现，就是 Hadoop

- 疑问：
    - 上面的场景，潜在瓶颈在哪？海量数据，各种瓶颈

目录

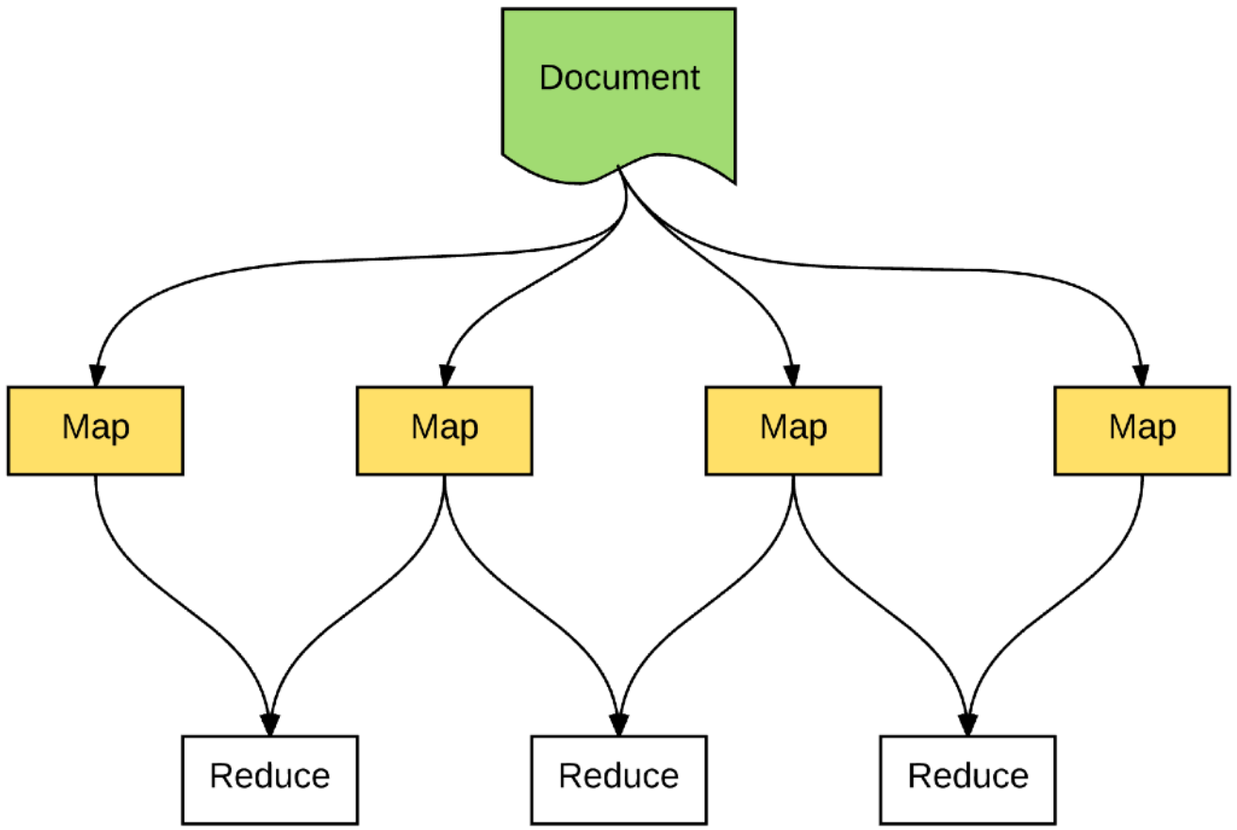
- 数据特征：海量数据处理 *vs.* 普通数据处理
- 基本原理：海量数据处理，典型场景和关键过程
- 关键思想：
  - 分发计算
  - 机架感知
- 本质分析

关键思想：分发计算



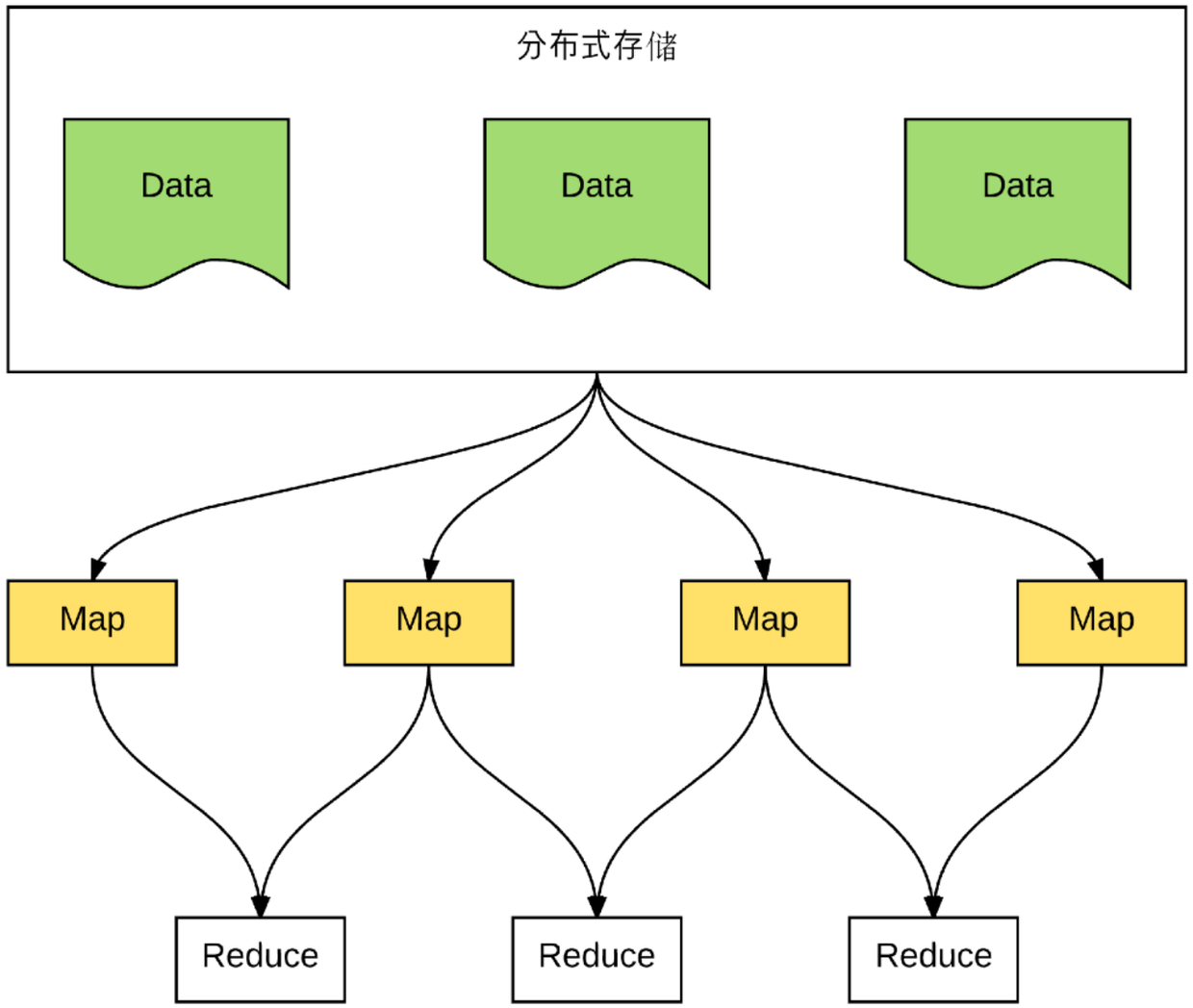
关键思想：分发计算

- 分布式存储：
  - 面对海量数据，单个存储节点，无法完成所有数据的存储，因此，采用分布式存储。



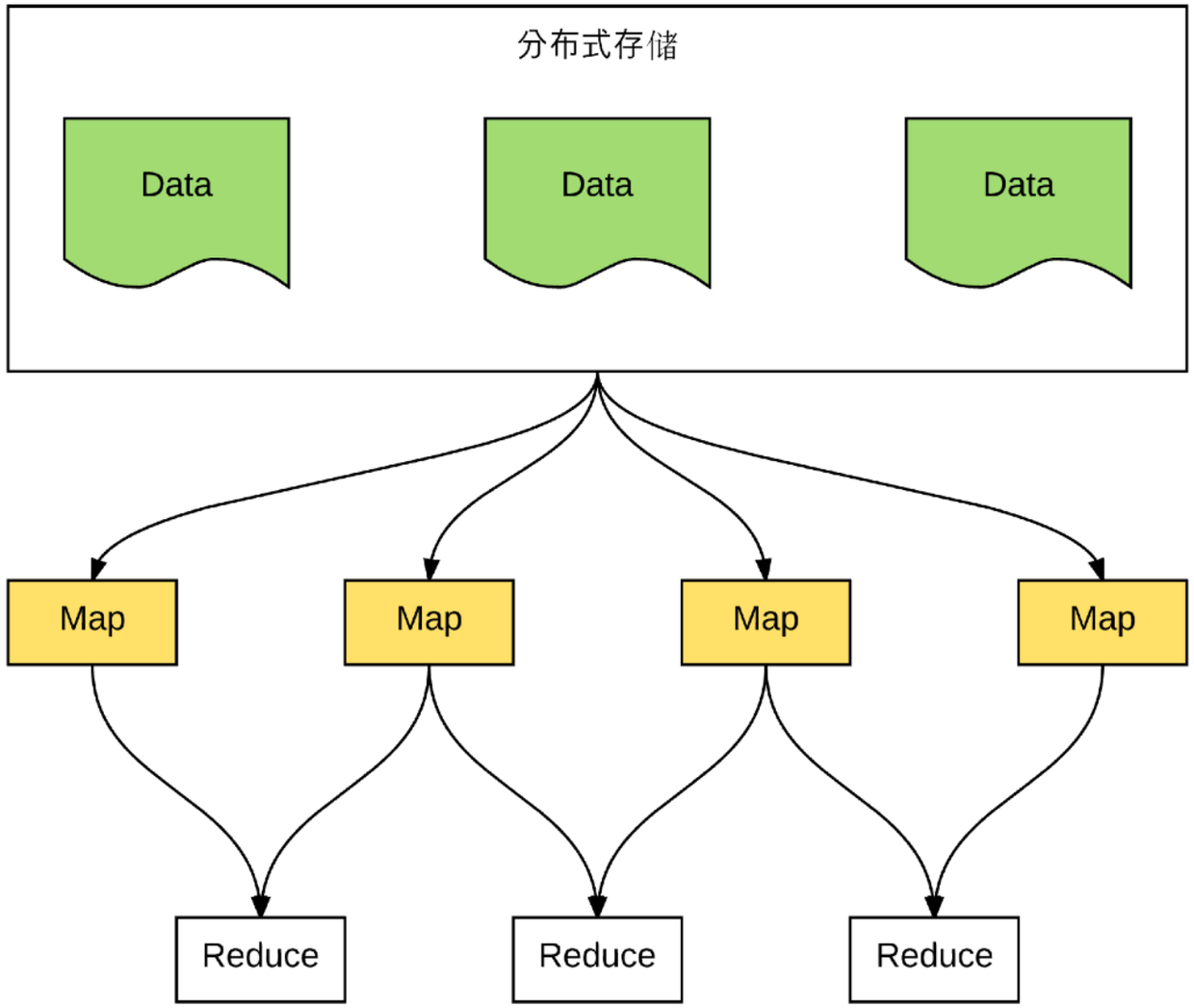
关键思想：分发计算

- 分布式存储：
  - 面对海量数据，单个存储节点，无法完成所有数据的存储，因此，采用分布式存储。



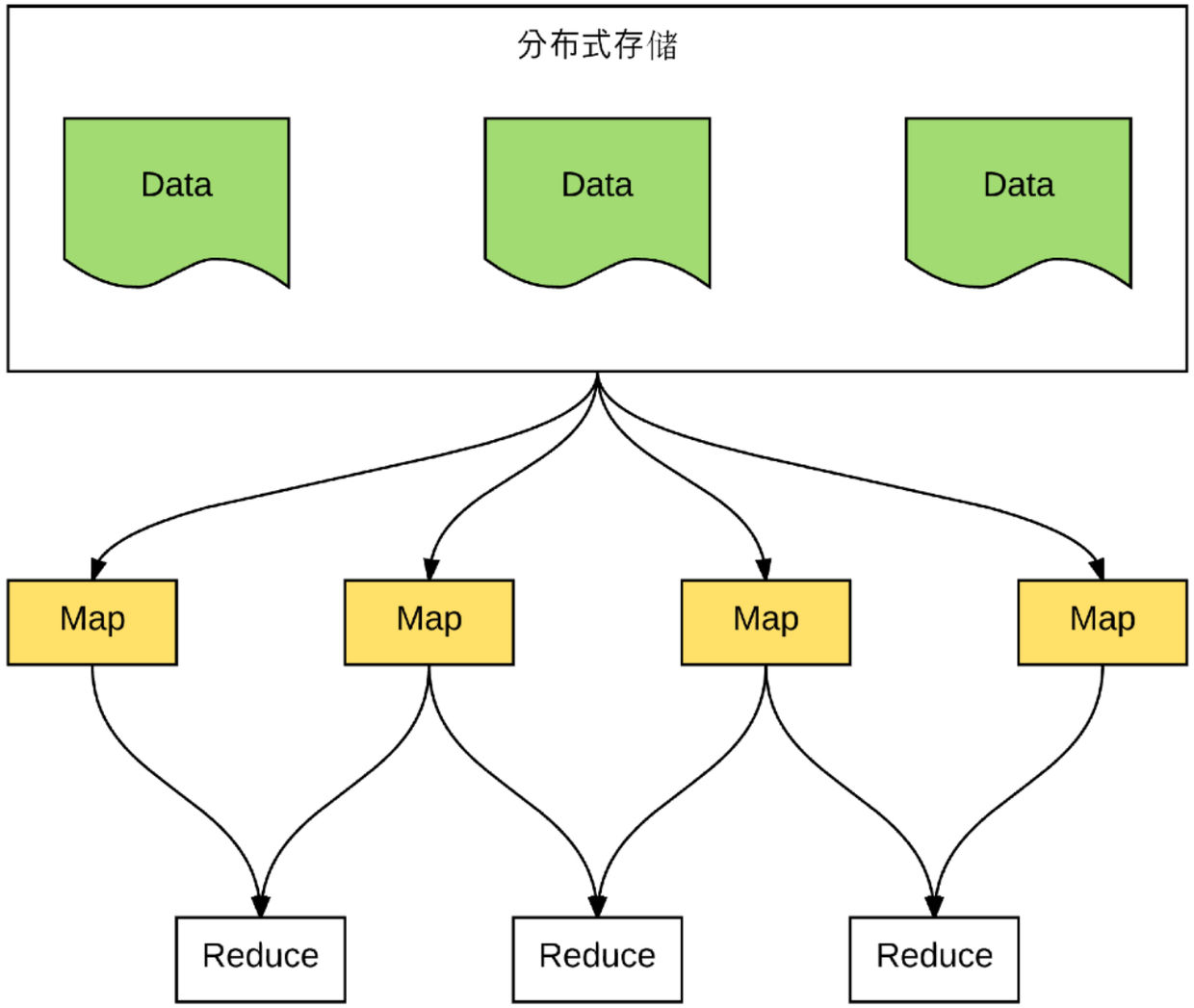
关键思想：分发计算

- 分布式存储：
  - 面对海量数据，单个存储节点，无法完成所有数据的存储，因此，采用分布式存储。
- 存在的问题：
  - 小数据量时，都没有问题，一旦是真的海量数据，就会导致：**数据分片过程**中，Map 节点读取 Data 时，**耗费大量的网络带宽**，网络被打满，成为新的**瓶颈**，随着数据量的剧增，达到网络带宽的瓶颈。

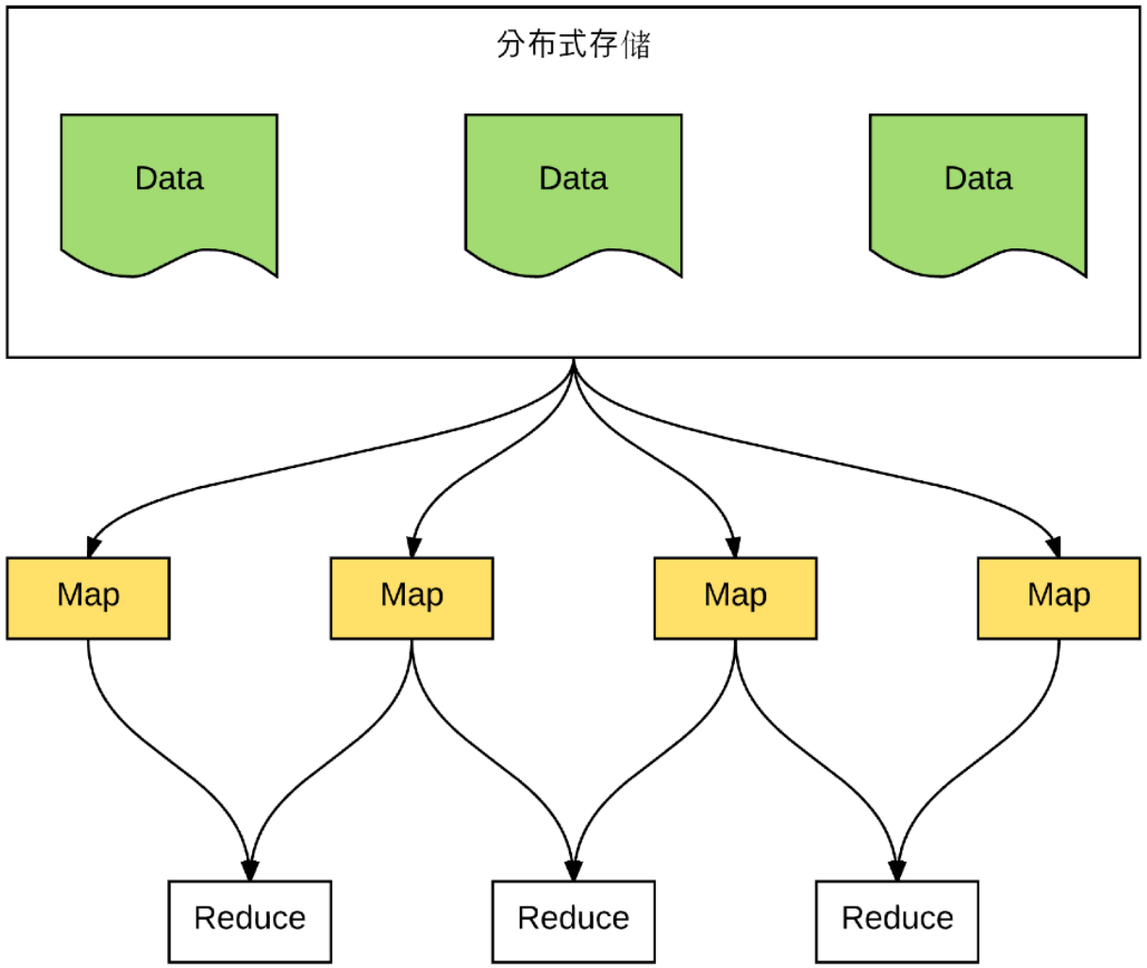


关键思想：分发计算

- 分布式存储：
  - 面对海量数据，单个存储节点，无法完成所有数据的存储，因此，采用分布式存储。
- 存在的问题：
  - 小数据量时，都没有问题，一旦是真的海量数据，就会导致：**数据分片过程**中，Map 节点读取 Data 时，**耗费大量的网络带宽**，网络被打满，成为新的**瓶颈**，随着数据量的剧增，达到网络带宽的瓶颈。
- 几个瓶颈及解决思路：
  - **磁盘容量**的瓶颈：采用**分布式存储**解决
  - 单机**计算能力**的瓶颈：采用**分布式计算**解决
  - **网络带宽**的瓶颈：海量数据，Map 节点，读取海量数据，达到网络，怎么办?

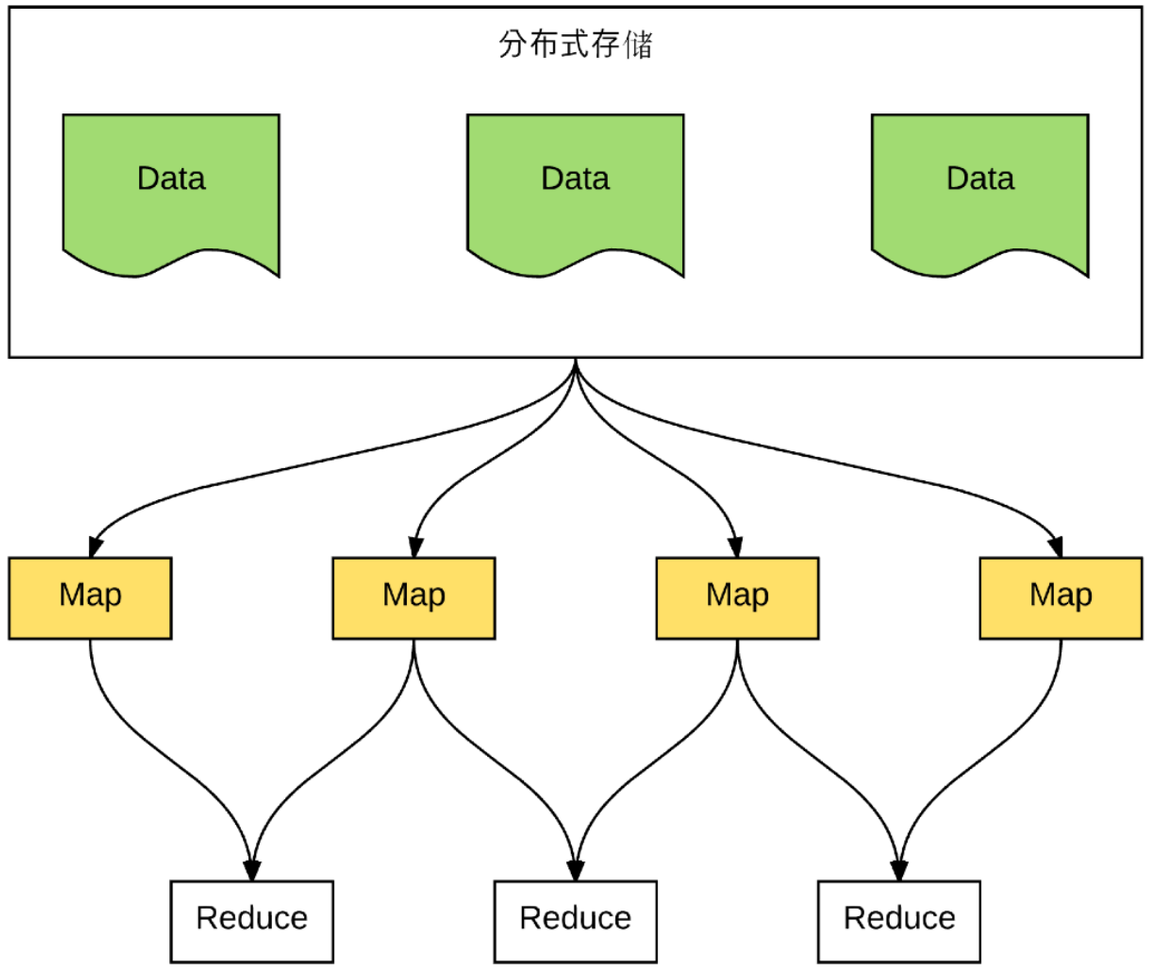


关键思想：分发计算



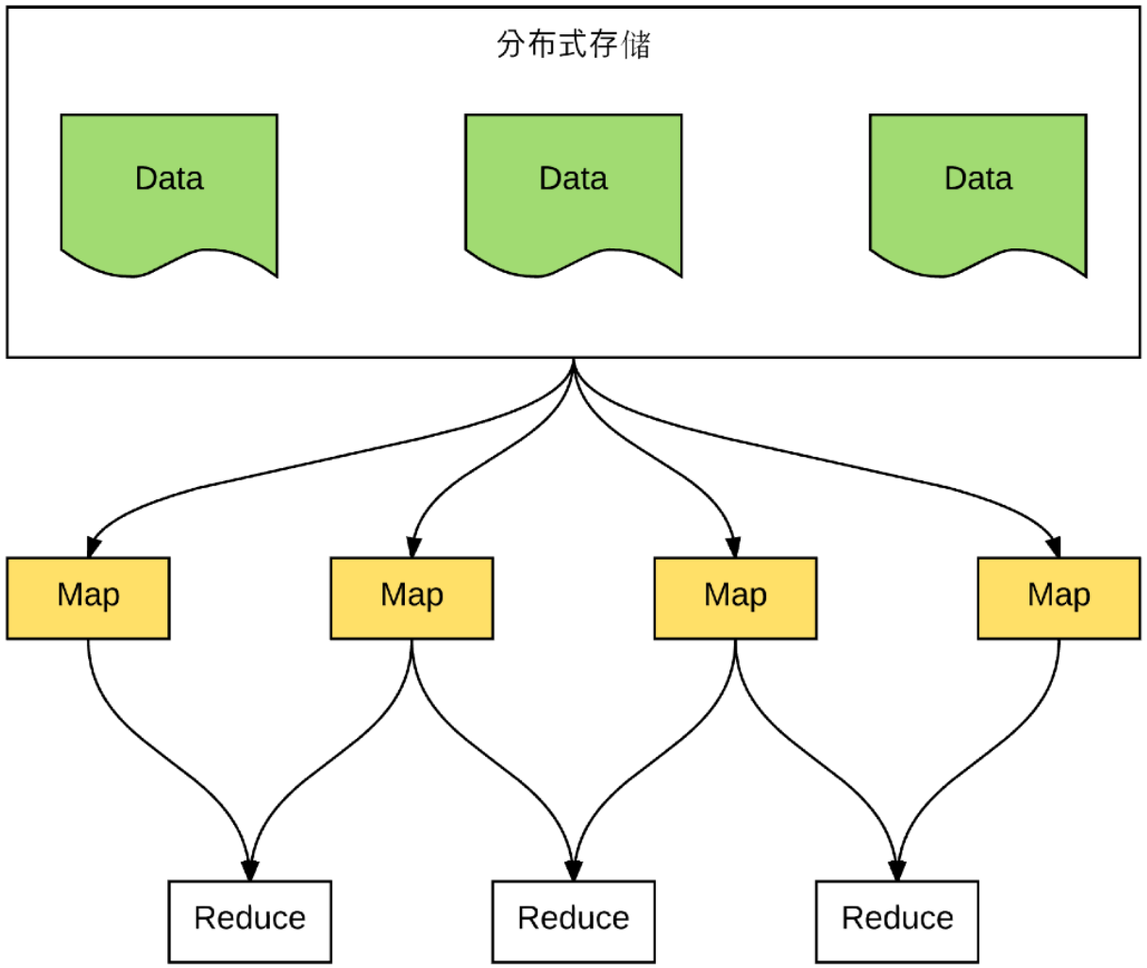
关键思想：分发计算

- 为什么会出现：网络带宽瓶颈?



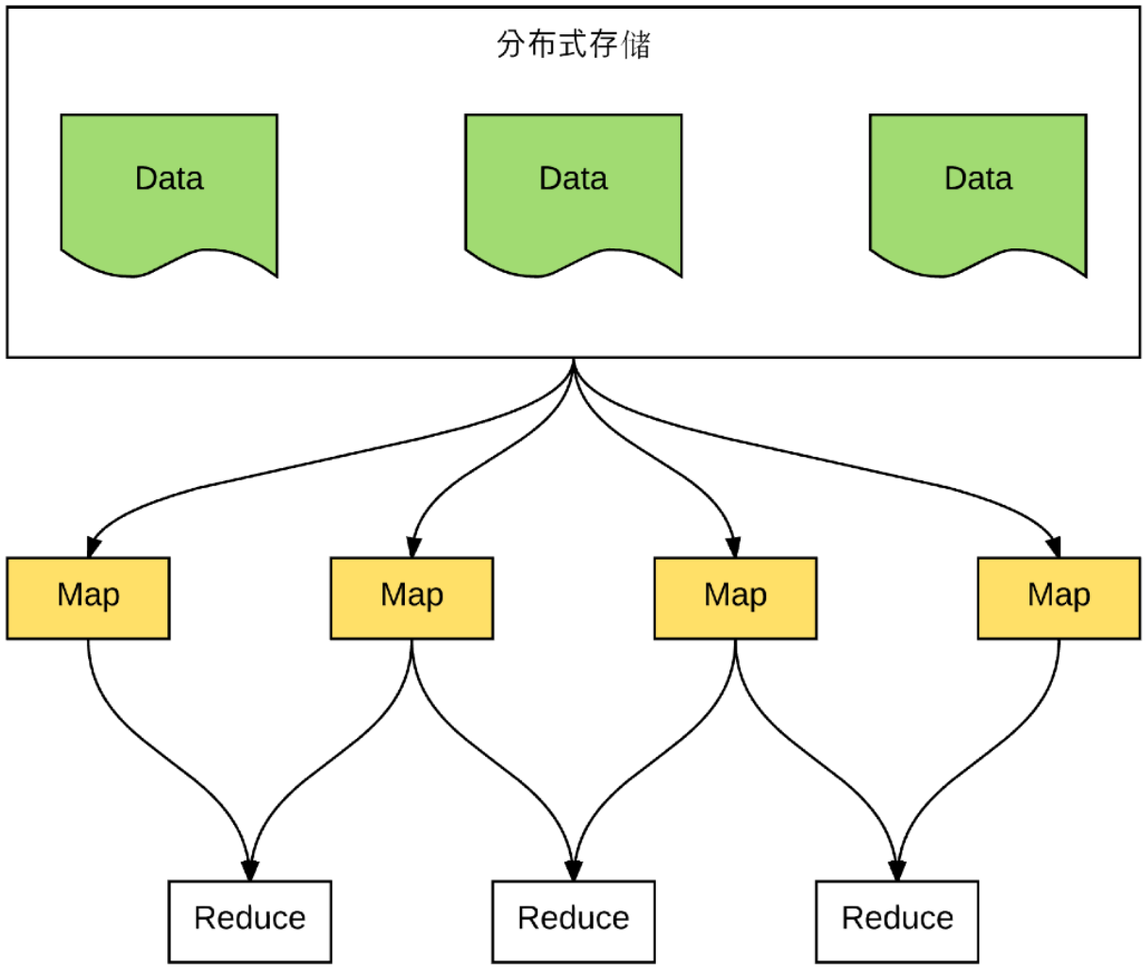
关键思想：分发计算

- 为什么会出现：网络带宽瓶颈?
- 本质原因：
  - Map 节点，计算过程中，需要输入，需要数据，就从其他地方读取数据，耗费网络带宽。



关键思想：分发计算

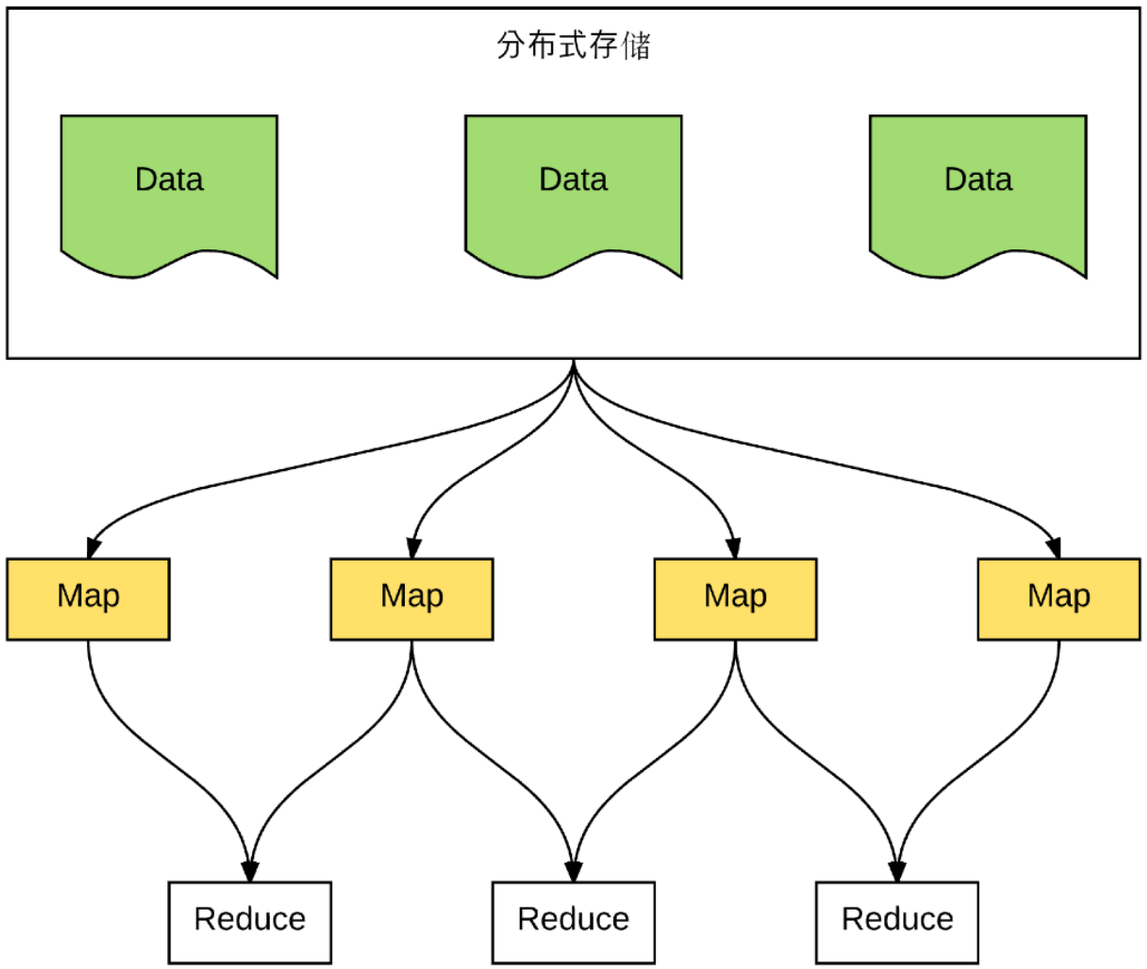
- 为什么会出现：网络带宽瓶颈?
- 本质原因：
  - Map 节点，计算过程中，需要输入，需要数据，就从其他地方读取数据，耗费网络带宽。



- 换个思路，就能解决上述问题，上述本质原因是：
  - 分发数据，即，计算节点把数据，读取到本地，耗费计算。

关键思想：分发计算

- 为什么会出现：网络带宽瓶颈?
- 本质原因：
  - Map 节点，计算过程中，需要输入，需要数据，就从其他地方读取数据，耗费网络带宽。

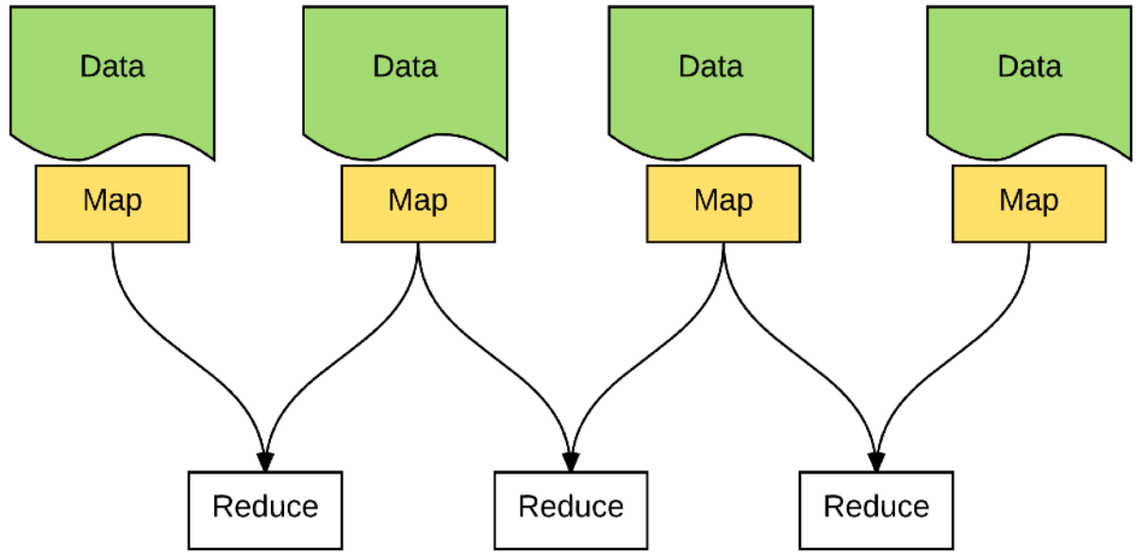


- 换个思路，就能解决上述问题，上述本质原因是：
  - 分发数据，即，计算节点把数据，读取到本地，耗费计算。

- 解决思路：
  - 分发计算（Deliver Compation），把计算逻辑（计算代码）分发到 Data 所在的节点上，就能极大降低网络消耗，简直是天才。

关键思想：分发计算

- 为什么会出现：网络带宽瓶颈?
- 本质原因：
  - Map 节点，计算过程中，需要输入，需要数据，就从其他地方读取数据，耗费网络带宽。



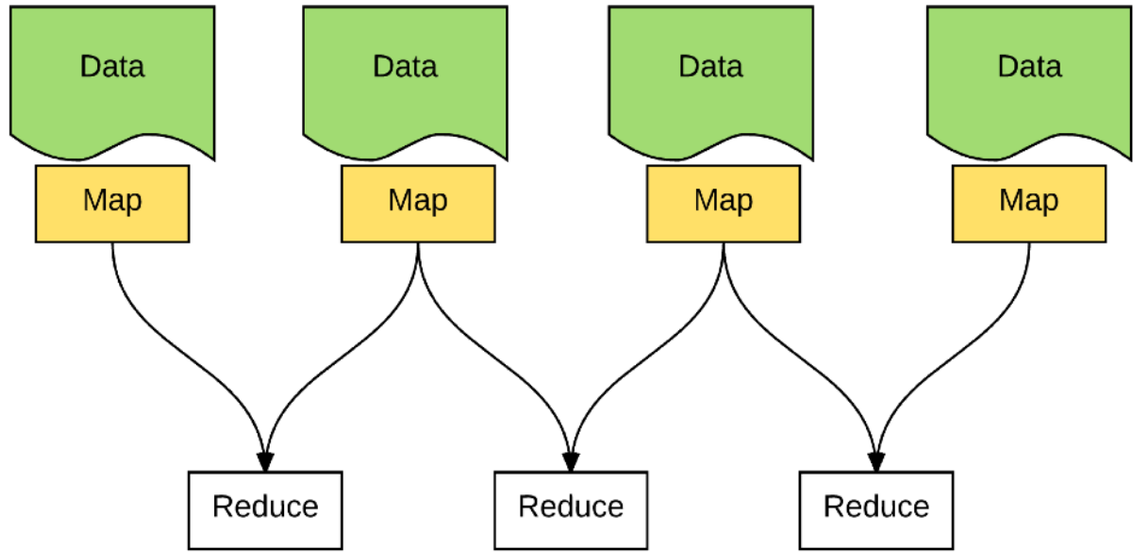
- 换个思路，就能解决上述问题，上述本质原因是：
  - 分发数据，即，计算节点把数据，读取到本地，耗费计算。

- 解决思路：
  - 分发计算（Deliver Compation），把计算逻辑（计算代码）分发到 Data 所在的节点上，就能极大降低网络消耗，简直是天才。

目录

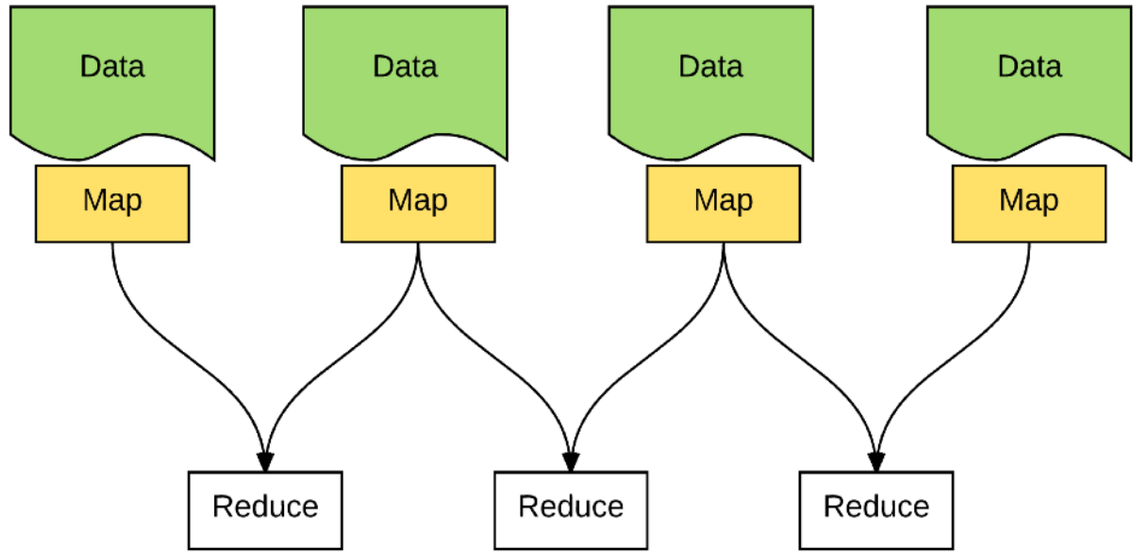
- 数据特征：海量数据处理 *vs.* 普通数据处理
- 基本原理：海量数据处理，典型场景和关键过程
- 关键思想：
  - 分发计算
  - 机架感知
- 本质分析

关键思想：机架感知



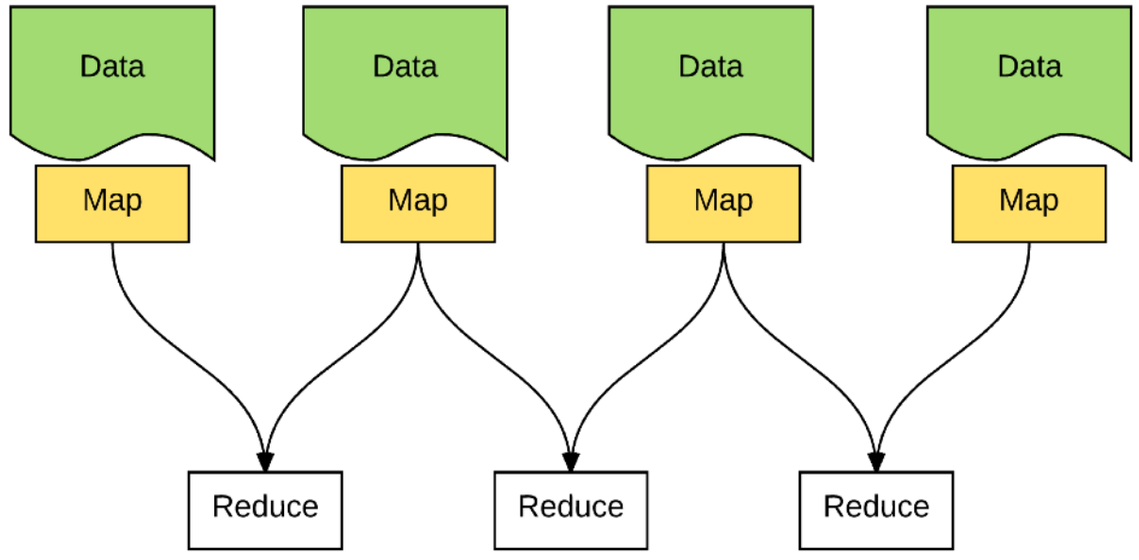
关键思想：机架感知

- 背景：分发计算存在的问题
  - 上面分发计算的基本思路是：把计算逻辑分发到数据节点上，就能避免大量的网络传输，解决海量数据处理场景下的带宽瓶颈。
  - 但实际上，分布式存储跟分布式计算，是要保持一定的隔离的：
    - 过度的耦合：会限制两个系统性能的发展和演进，无法实现计算引擎的可插拔
    - 计算能力受限：数据节点的数目，不应跟处理节点的数目绑定，即，不能因为节点数量，限制计算能力上限



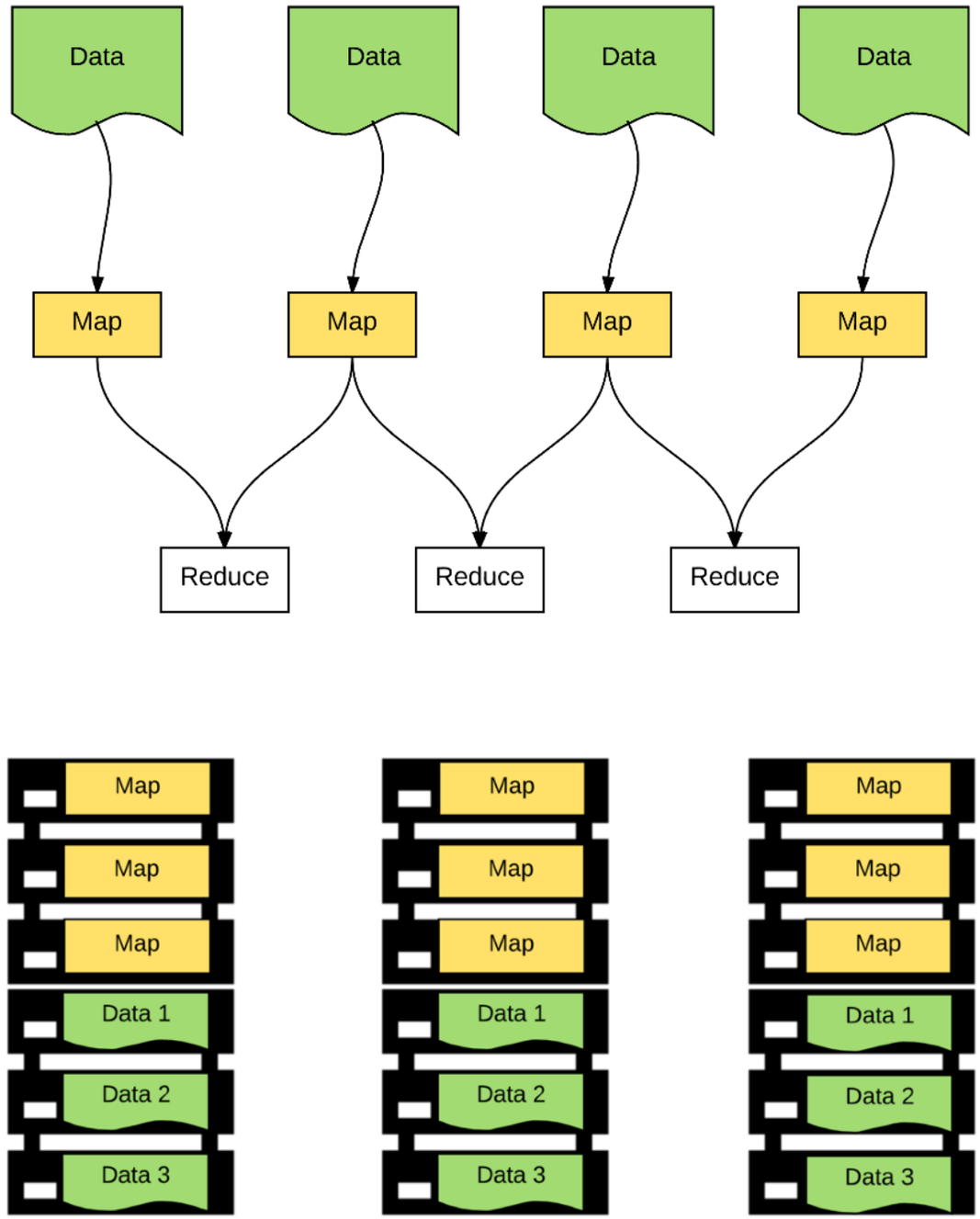
关键思想：机架感知

- 背景：分发计算存在的问题
  - 上面分发计算的基本思路是：把计算逻辑分发到数据节点上，就能避免大量的网络传输，解决海量数据处理场景下的带宽瓶颈。
  - 但实际上，分布式存储跟分布式计算，是要保持一定的隔离的：
    - 过度的耦合：会限制两个系统性能的发展和演进，无法实现计算引擎的可插拔
    - 计算能力受限：数据节点的数目，不应跟处理节点的数目绑定，即，不能因为节点数量，限制计算能力上限



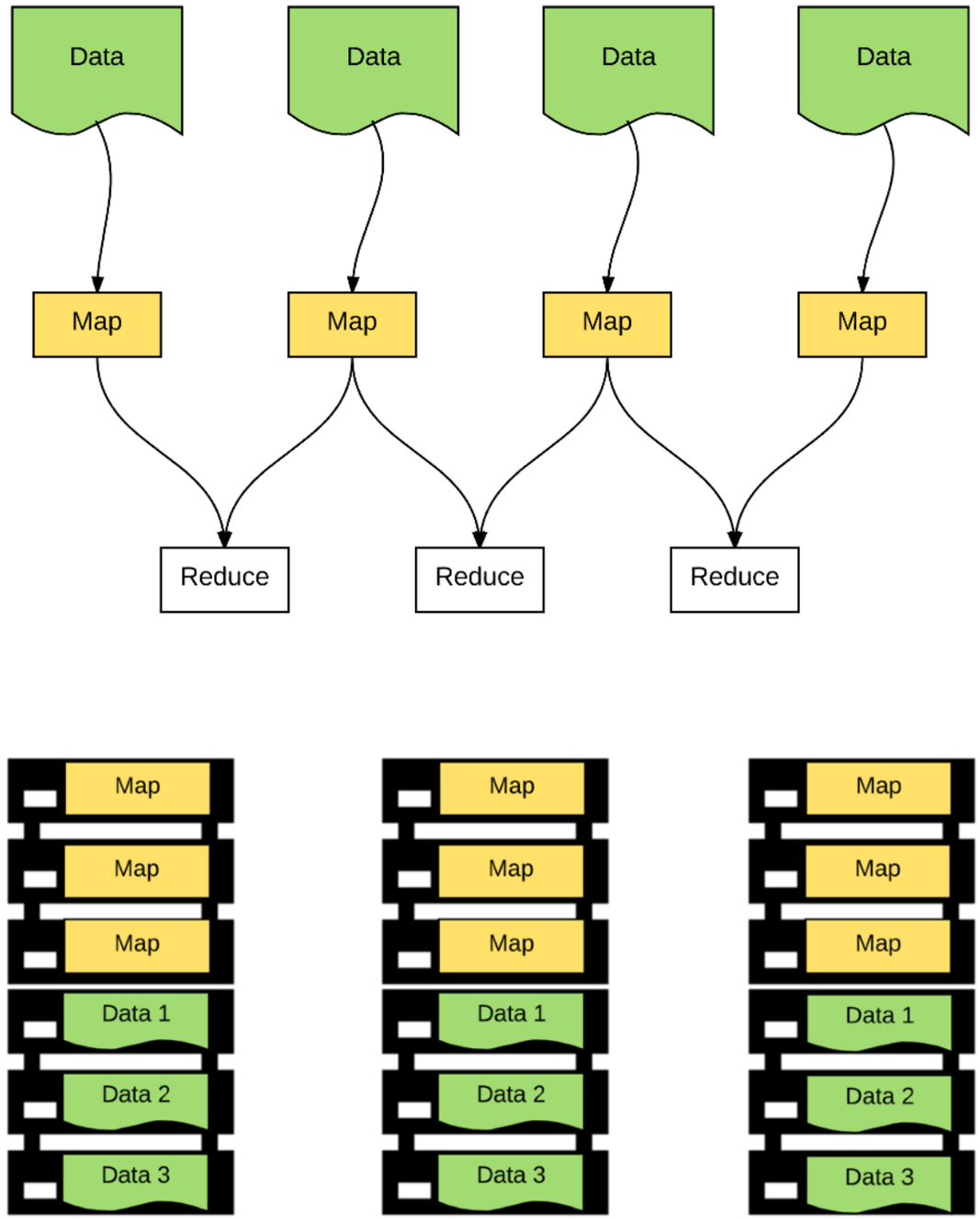
- 因此，不能让 Map 紧贴 Data。

关键思想：机架感知


Data
Data
Data
Data
Map
Map
Map
Map
Reduce
Reduce
Reduce
Map
Map
Map
Data 1
Data 2
Data 3
Map
Map
Map
Data 1
Data 2
Data 3
Map
Map
Map
Data 1
Data 2
Data 3

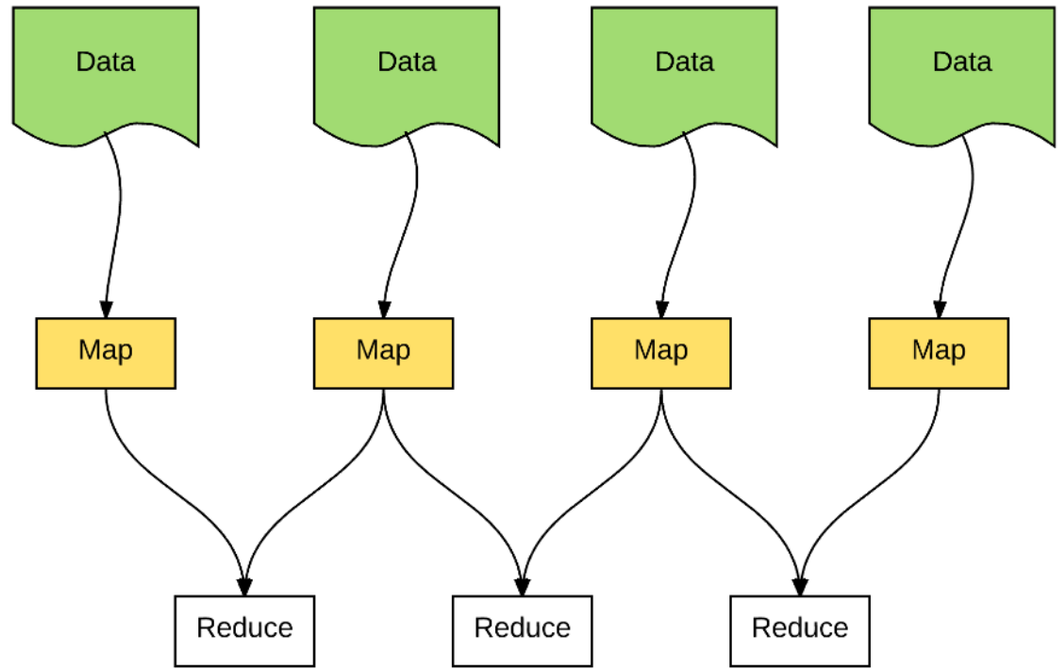
关键思想：机架感知

- 解决思路：
  - 既然不能让 Map 绑定 Data，那就让他们尽可能贴近就好了。

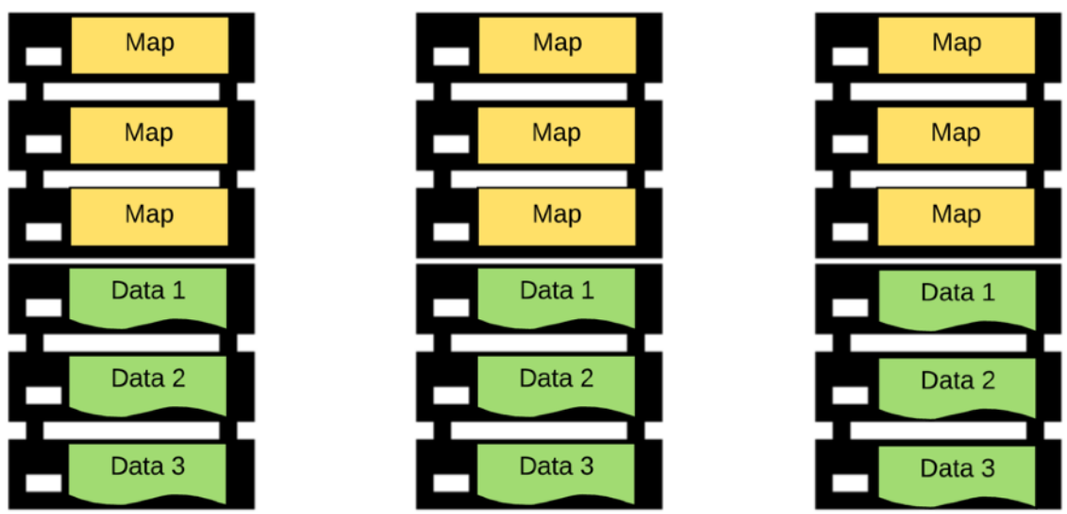


关键思想：机架感知

- 解决思路：
  - 既然不能让 Map 绑定 Data，那就让他们尽可能贴近就好了。

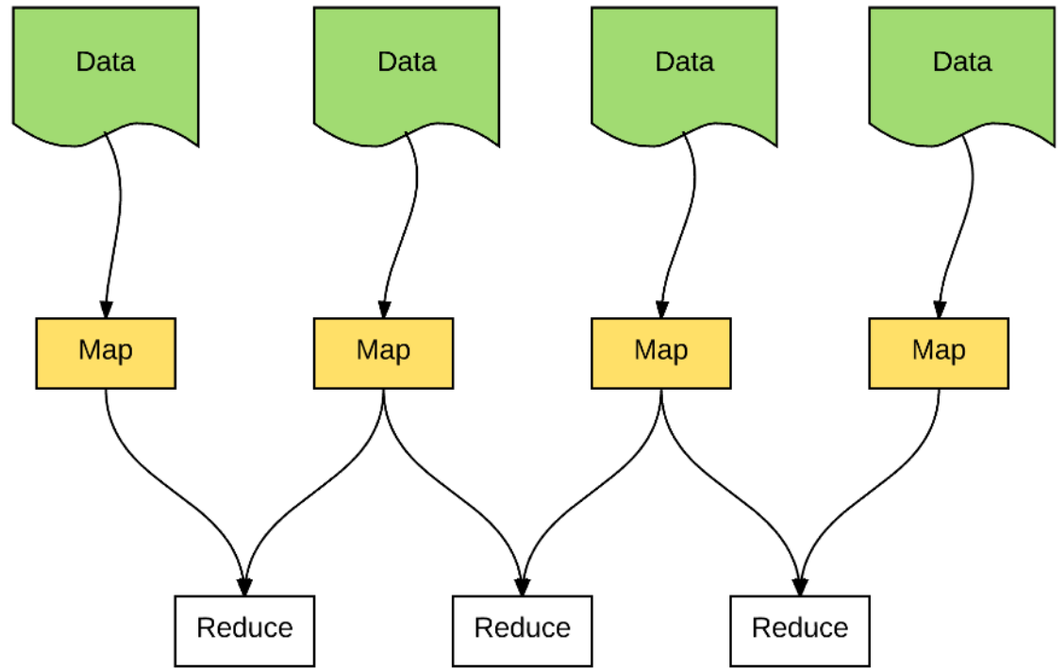


- 具体实现：
  - 尽可能让离 Data 物理距离近的 Map 来处理相应的 Data。
  - 物理距离，一般以：机架、机房来划分，即，同机架、同机房。

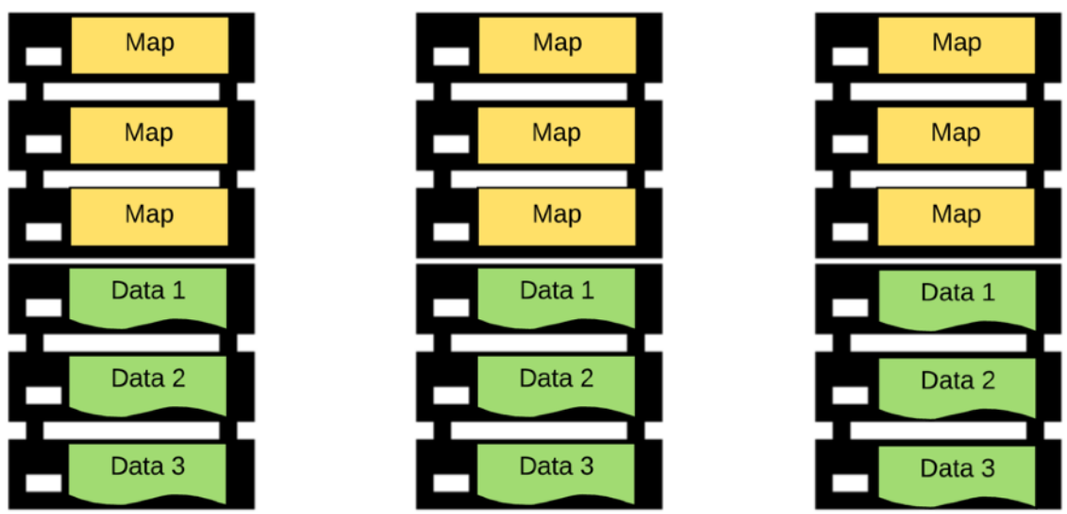


关键思想：机架感知

- 解决思路：
  - 既然不能让 Map 绑定 Data，那就让他们尽可能贴近就好了。



- 具体实现：
  - 尽可能让离 Data 物理距离近的 Map 来处理相应的 Data。
  - 物理距离，一般以：机架、机房来划分，即，同机架、同机房。



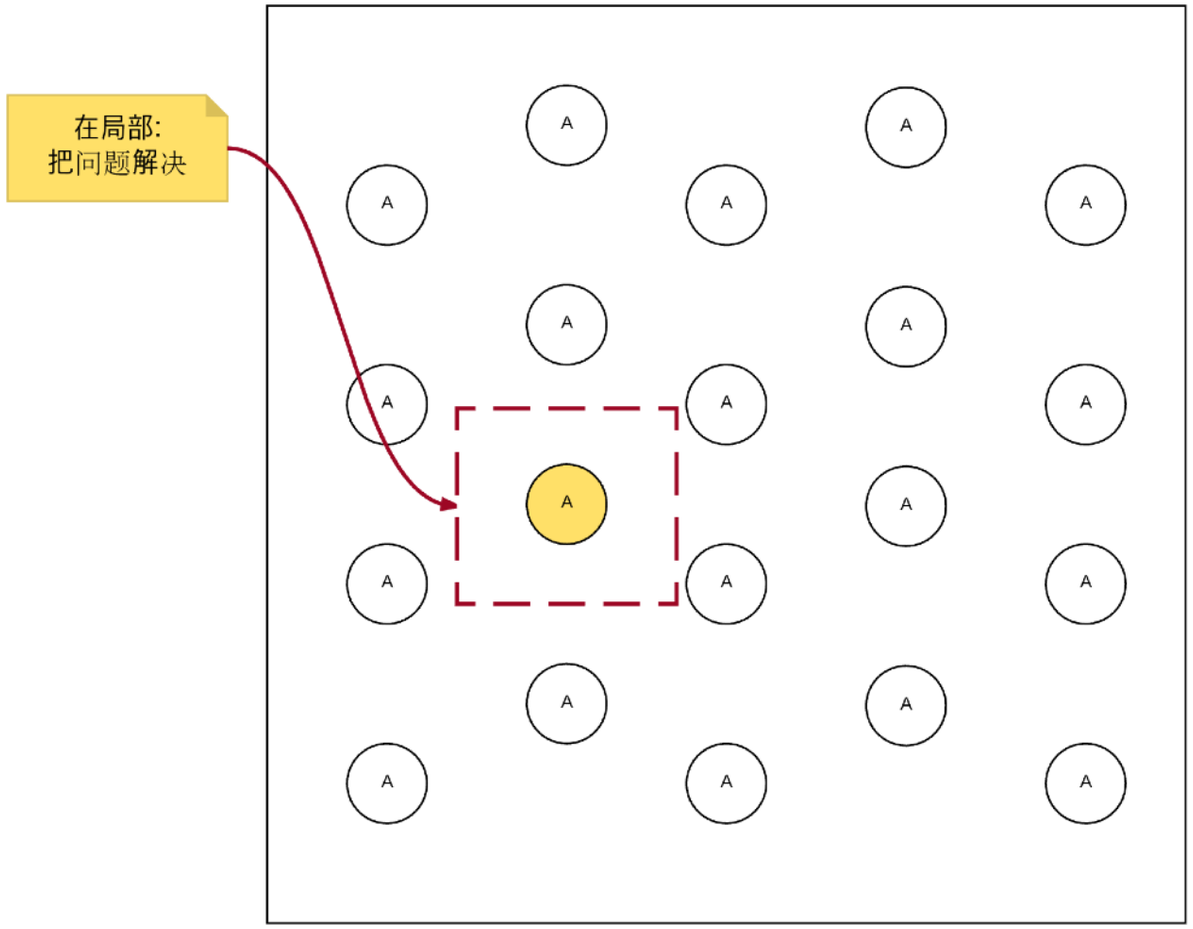
- Note：
  - 补充说明，分布式存储，为了提高系统的可靠性，一般会设置数据的副本个数，例如，设置数据副本数为 3， 则，采取适当策略，会在至少 3 个机架中，都存在同样的一份数据。

目录

- 数据特征：海量数据处理 *vs.* 普通数据处理
- 基本原理：海量数据处理，典型场景和关键过程
- 关键思想：
  - 分发计算
  - 机架感知
- 本质分析

本质分析：Principle of Locality

- 本质：
  - 局部性原理（Principle Of Locality），就近原则
- 具体：
  1. 网络局部性原理
  2. 磁盘局部性原理



本质分析：Principle of Locality

- 扁鹊早就知道局部性原理的厉害之处：
    - 魏文王问扁鹊曰：「子昆弟三人其孰最善为医？」
    - 扁鹊曰：「长兄最善，中兄次之，扁鹊最为下。」
    - 魏文侯曰：「可得闻邪？」
    - 扁鹊曰：
        - 长兄於病视神，未有形而除之，故名不出於家。
        - 中兄治病，其在毫毛，故名不出於闾。
        - 若扁鹊者，鑱血脉，投毒药，副肌肤，闲而名出闻於诸侯。

总结 & 回顾

- 海量数据处理，基本原理：
    - 海量数据处理，涉及 4 个典型过程：
        - 数据分片
        - Map
        - Shuffle
        - Reduce
    - 海量数据处理，调优时，根据具体的业务场景、数据特征，减弱「数据倾斜」现象
    - Map-Reduce 是一种编程模型，实现方式可以有多种，代表性的实现，就是 Hadoop

- 海量数据处理，关键思想：
    - 分发计算：把计算逻辑（计算代码）分发到 Data 所在的节点上，就能极大降低网络消耗
    - 机架感知：尽可能让离 Data 近的 Map 节点来处理数据，感知 Data 跟 Map 的物理距离
    - 本质：网络局部性原理

参考资料

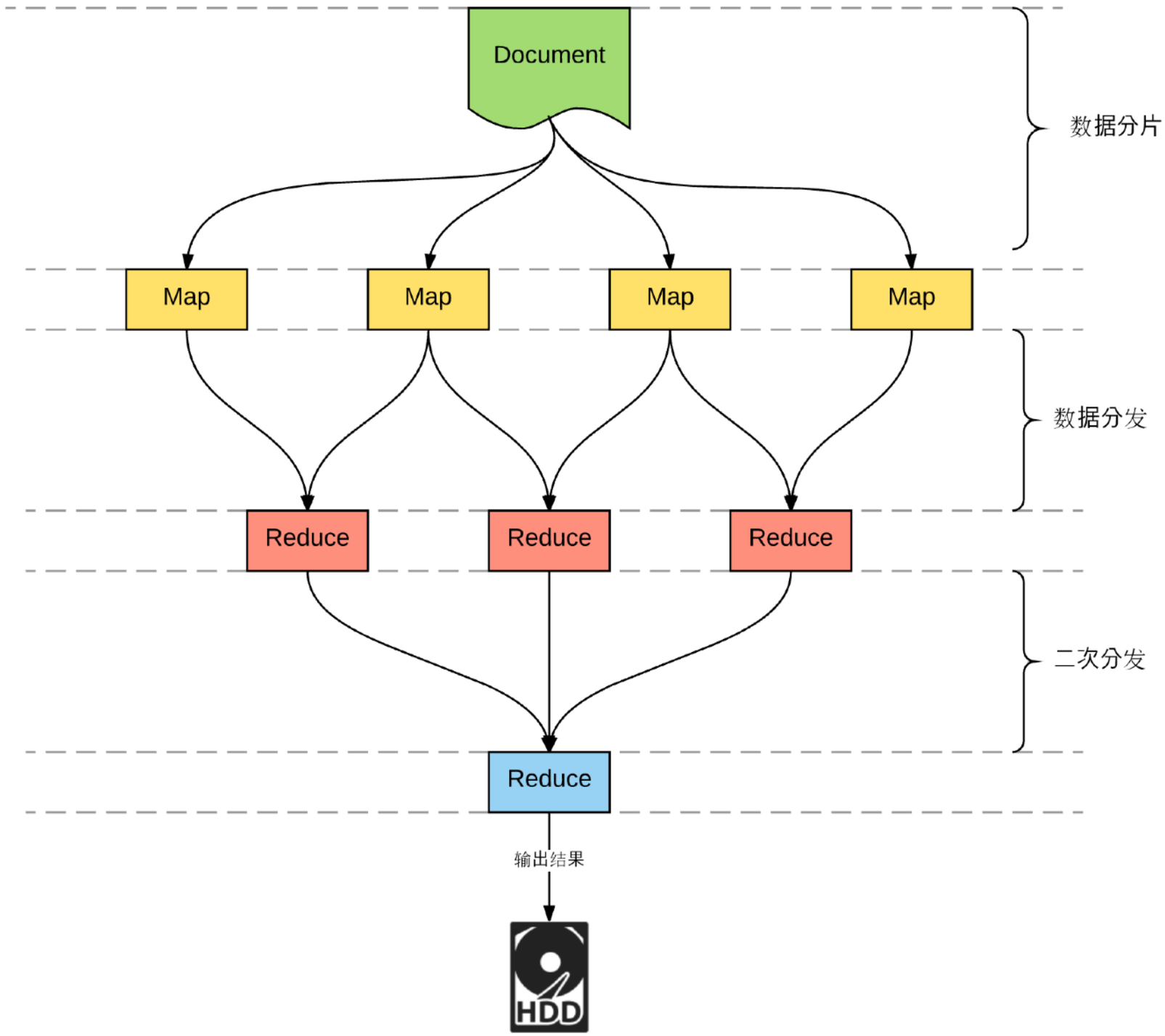
- http://dl.acm.org/citation.cfm?id=1327492 Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

- http://www.cloudera.com/content/dam/www/static/documents/whitepapers/hadoop-and-hdfs.pdf Hadoop and HDFS: Storage for Next Generation Data Management (White Paper of Cloudera)

- https://en.wikipedia.org/wiki/Principle_of_locality Principle Of Locality wiki

- http://hadoop.apache.org/

Q & A


Document
数据分片
Map
Map
Map
Map
数据分发
Reduce
Reduce
Reduce
二次分发
Reduce
输出结果
HDD