云计算与大数据分析课程研究报告

Hadoop 1.0 与 2.0 架构演进及性能实验分析

数据科学学院 海南 B 班
智能科技与服务专业（民族大学合办）
张心悦 D23090600661

2026 年 4 月 24 日

1. 选题背景与研究问题

方向 C：

Hadoop 1.0 与 2.0 架构演进及性能实验分析。

1. JobTracker 和 YARN 在高并发作业下差异多大。
2. HDFS Federation 能否缓解单命名空间瓶颈。
3. 在重复分析任务中，Spark 内存计算相对磁盘计算能快多少。

Hadoop 1.0 以 HDFS 和 MapReduce 为核心，适合大规模批处理任务。

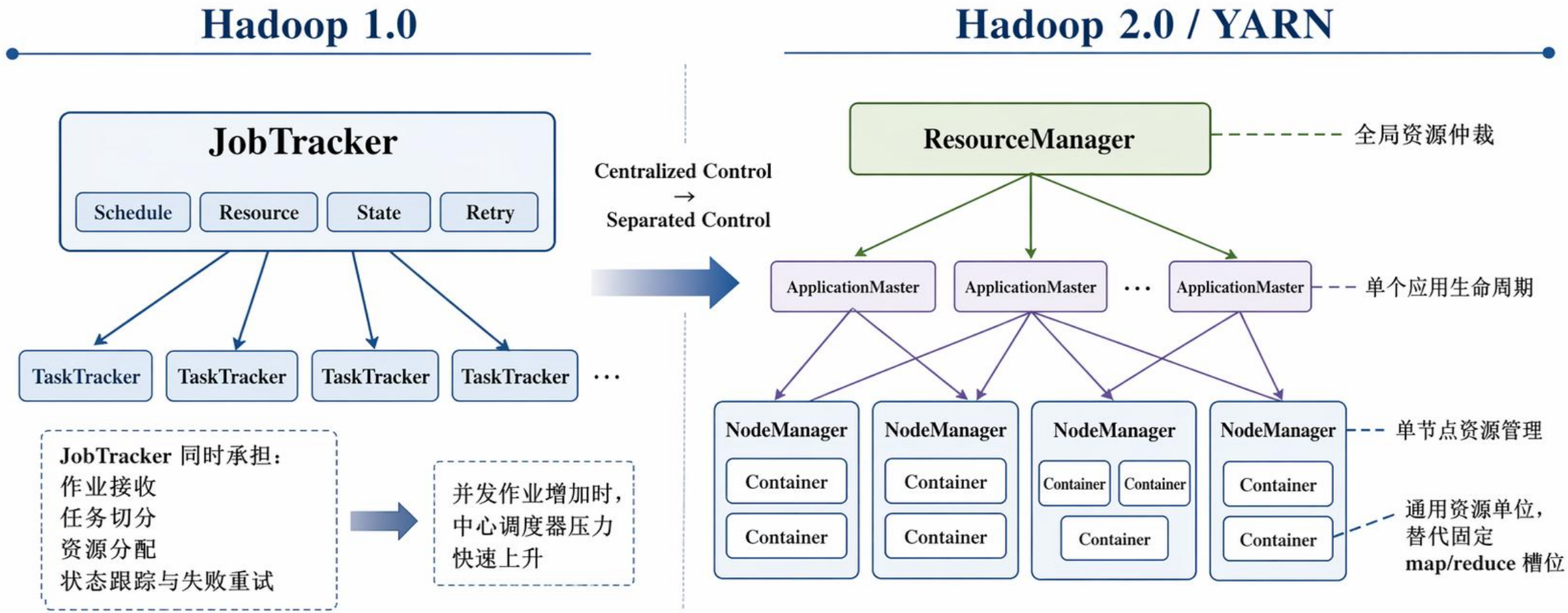
随着集群规模扩大，早期架构逐渐暴露出三个限制：

1. JobTracker 同时负责资源分配、任务调度、状态维护和失败重试。

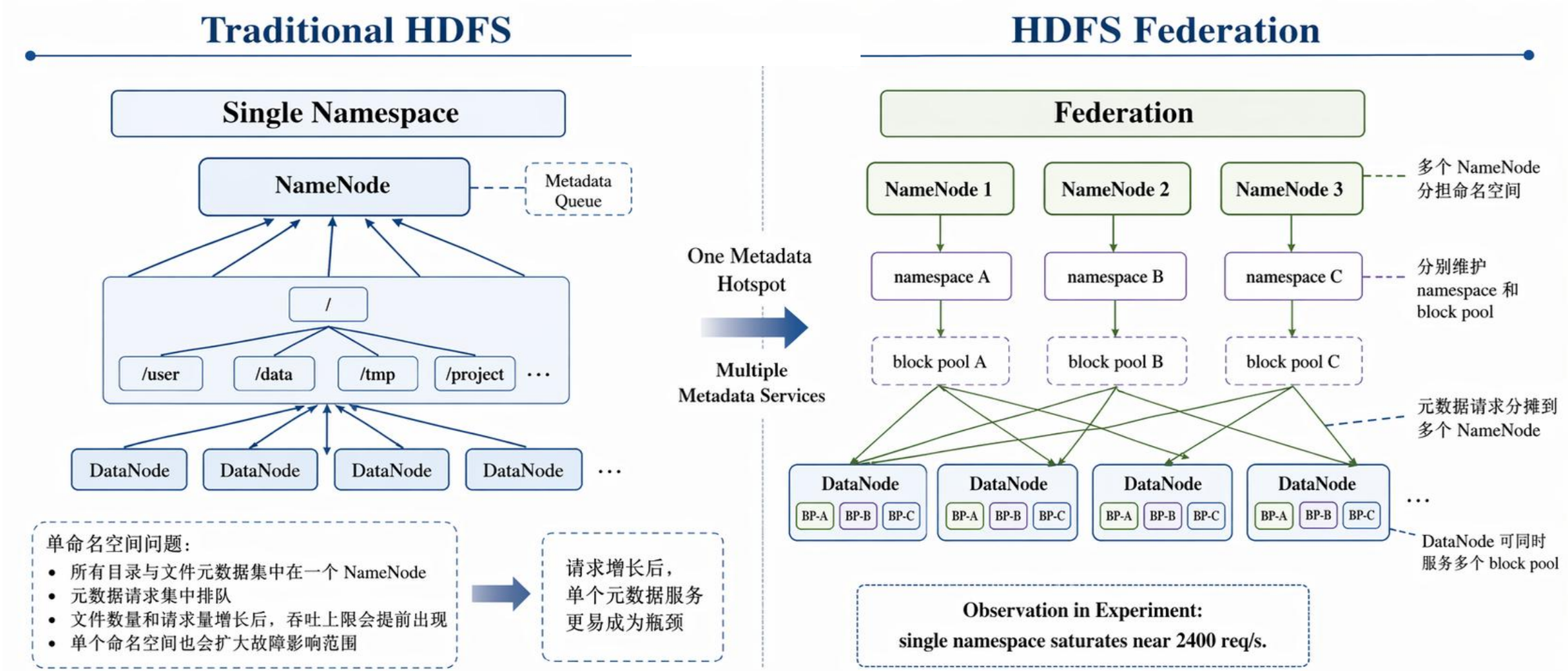
2. 单 NameNode 需要维护整个命名空间，元数据请求容易集中。

3. MapReduce 多轮分析依赖反复读盘和中间结果落盘。

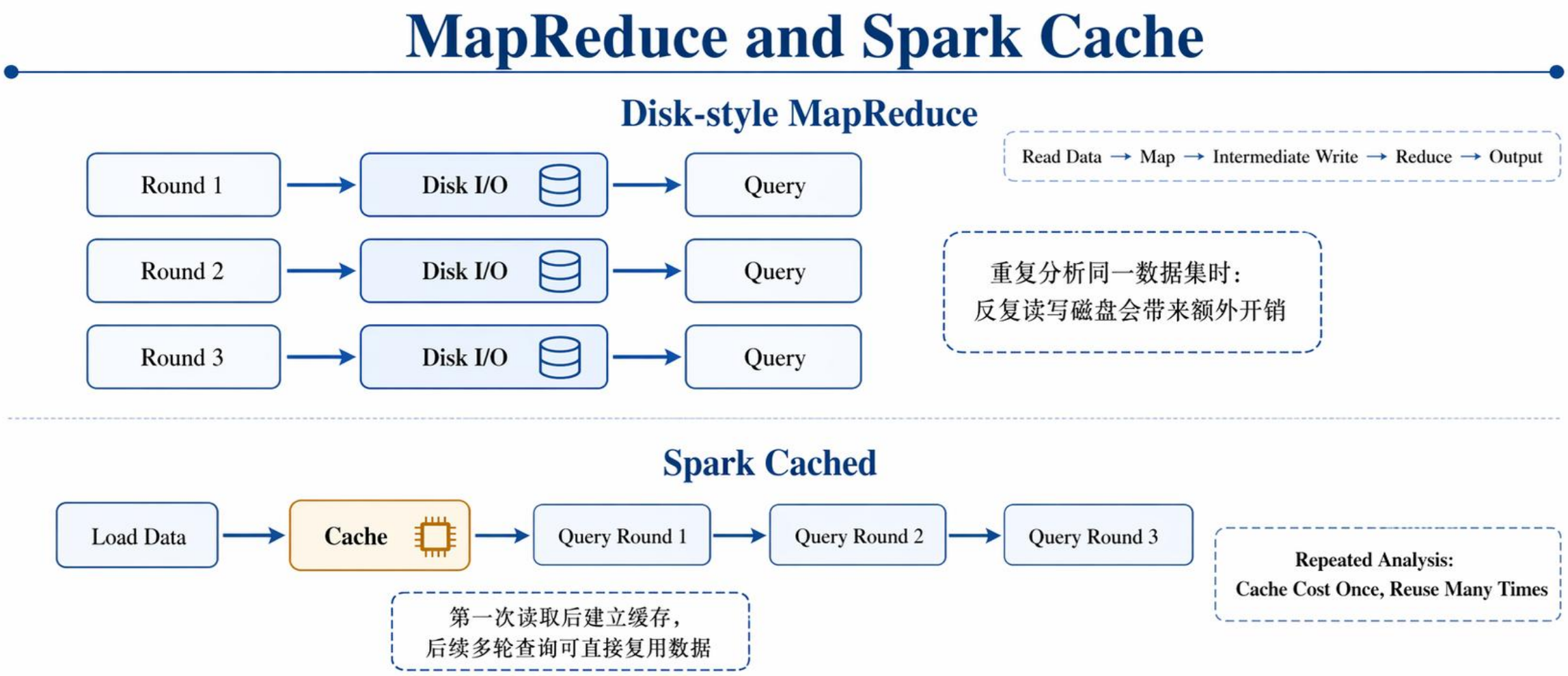
2.1 从 JobTracker 到 YARN



2.2 从单命名空间到 HDFS Federation



3.3 从磁盘批处理到 Spark 内存复用



4.1 实验环境与方法

基础环境：

本次实验采用单机容器化环境。宿主机为 Windows 11，使用 Docker Desktop 与 WSL 2 运行 Ubuntu 22.04 容器。

软件环境：

容器内统一配置 Python 3、OpenJDK 8 和 PySpark 3.5.5，用于实验脚本执行、Hadoop/Spark 运行支持和迭代分析实验。

实验一与实验二采用多容器方式组织调度与元数据服务，实验三在独立计算容器中完成磁盘计算与 Spark 缓存计算对比。

4.2 实验一设计：JobTracker vs YARN

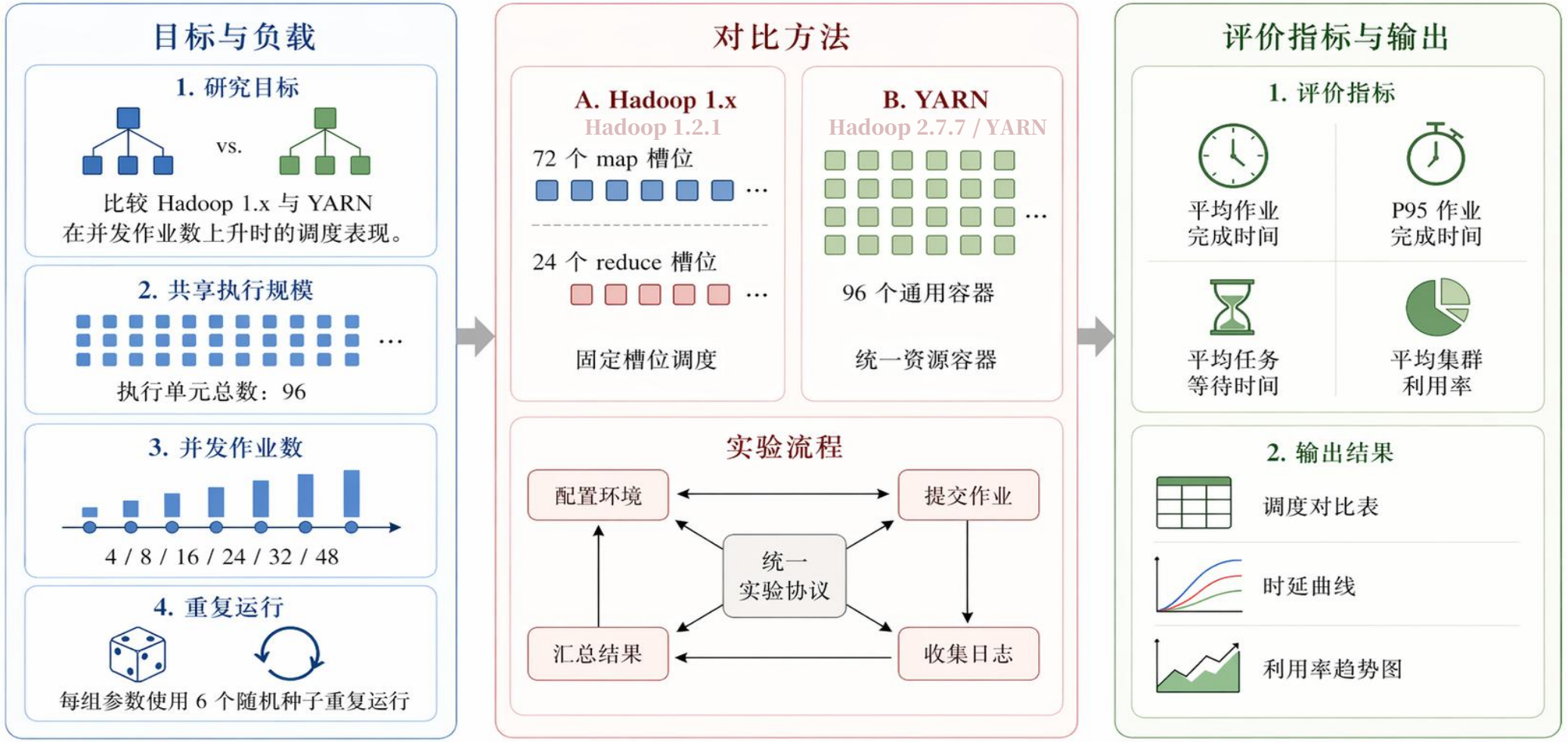


图 1 实验一设计流程图

4.2 实验一结果：YARN 降低高并发调度压力

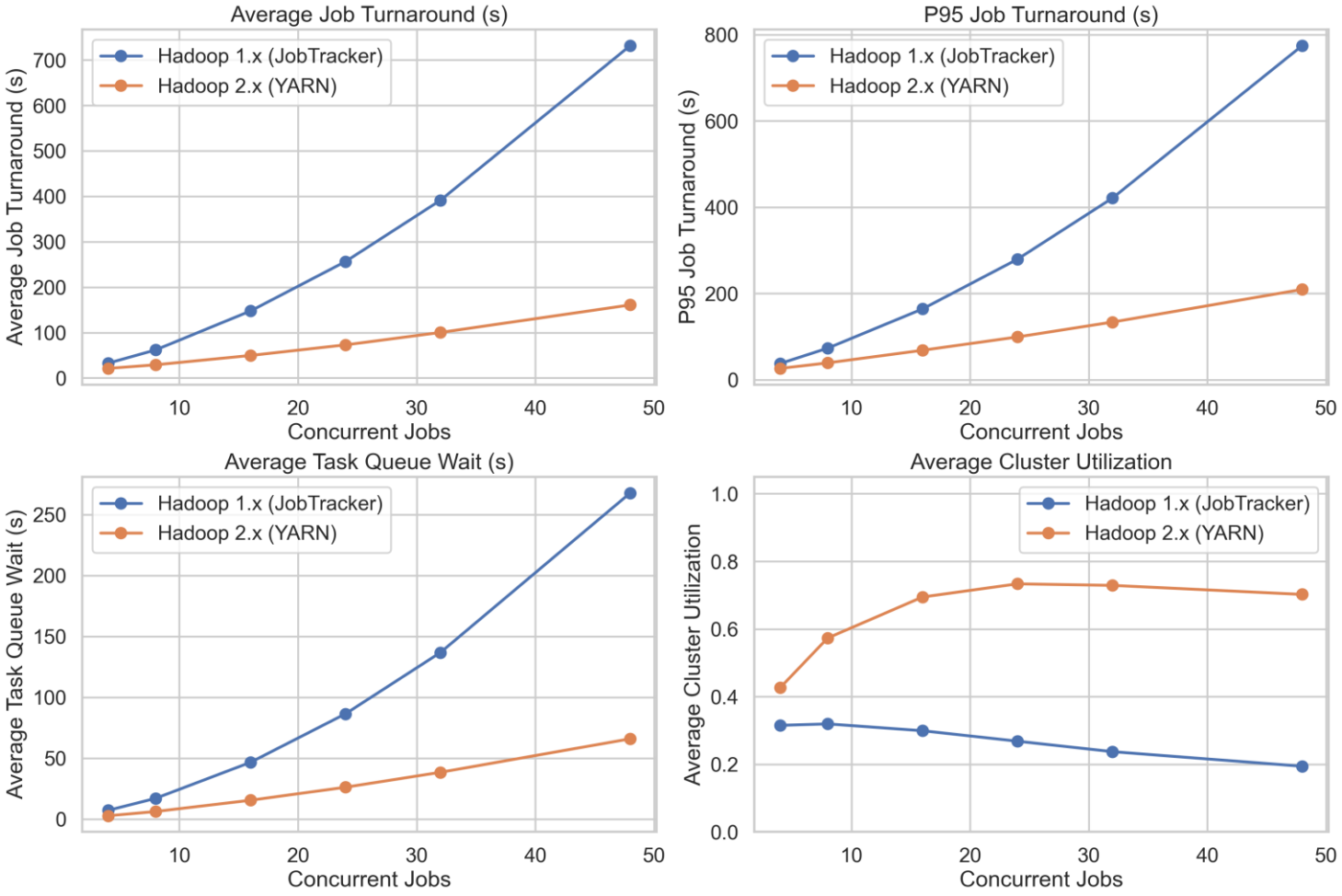


图 2 Hadoop 1.x 与 YARN 在不同并发作业数下的关键指标变化

- 并发 8 时，YARN 平均作业完成时间下降 53.01%。
- 并发 24 时，YARN 平均作业完成时间下降 71.57%。
- 并发 48 时，YARN 平均作业完成时间下降 78.02%。
- 并发 48 时，集群利用率由 19.41% 提升到 70.23%。

高并发阶段的主要差异来自**控制平面压力**和**资源模型**。

4.3 实验二设计：HDFS Federation

图 3 实验二设计流程图

4.3 结果：baseline

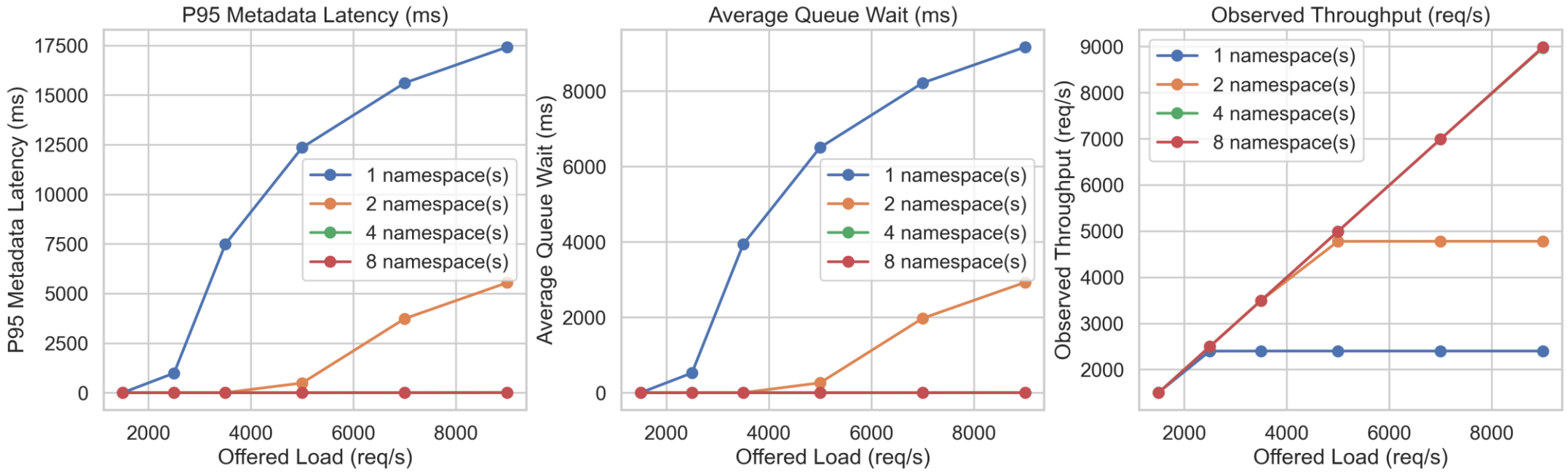


图 4 不同命名空间数量下的 P95 元数据时延、平均排队等待时间与观测吞吐量。

- 单命名空间在约 2400 req/s 附近达到吞吐上限。
- 5000 req/s 下，单命名空间 P95 时延为 12357.44 ms，4 个命名空间仅为 1.98 ms。
- 9000 req/s 下，8 个命名空间仍保持 1.80 ms 的 P95 时延和 8987.00 req/s 的吞吐。

Federation 的主要作用是把元数据热点拆分，使请求队列不再集中在某一个 NameNode 上。

4.4 实验三设计：磁盘计算 vs Spark 内存计算

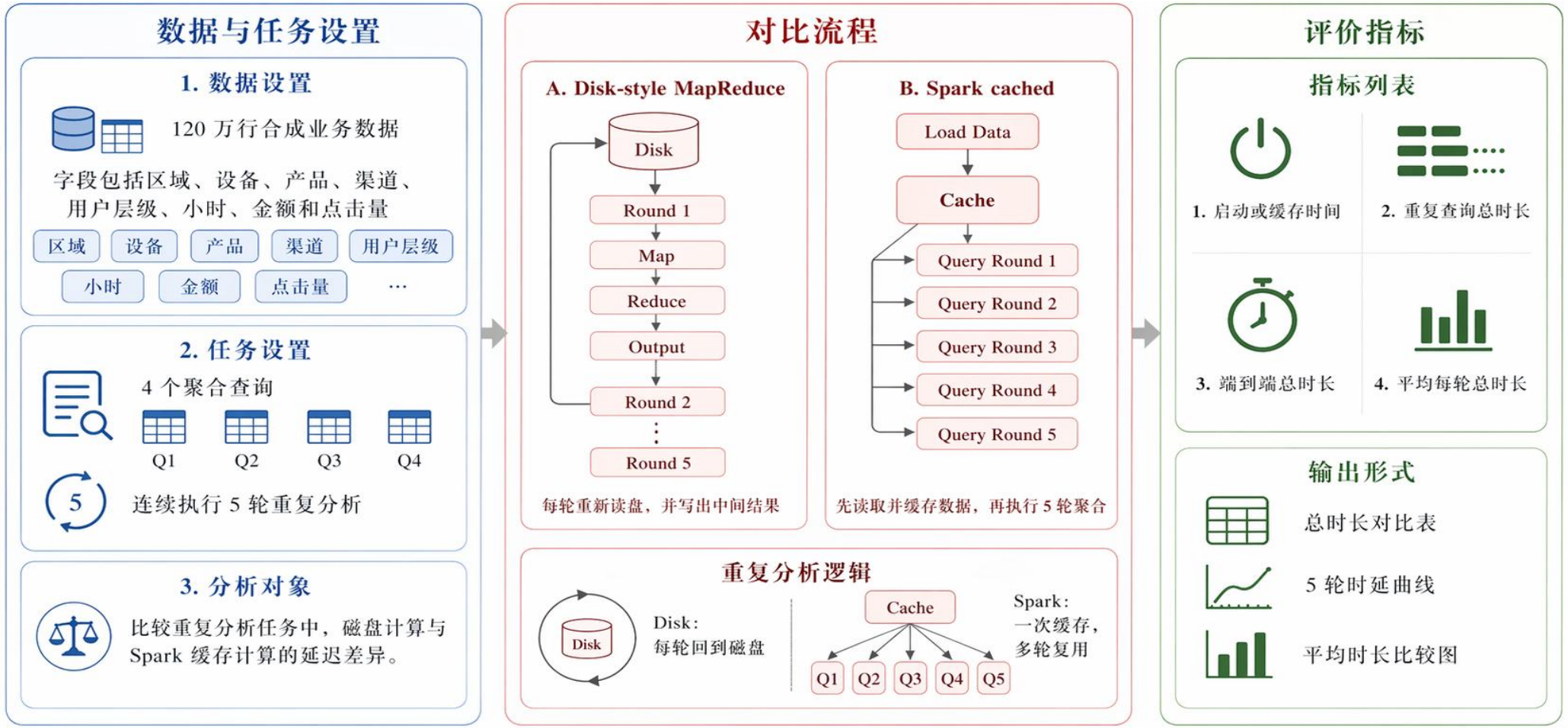
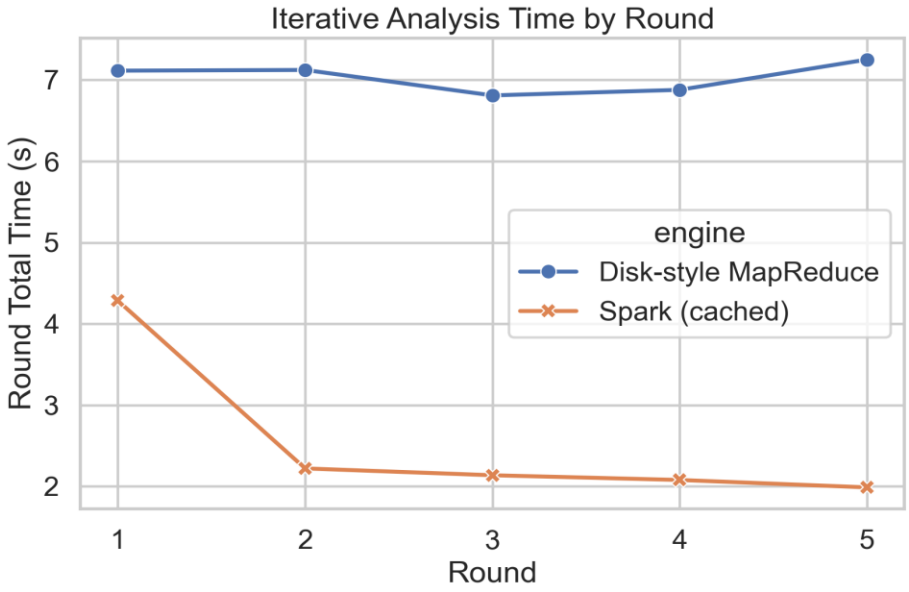
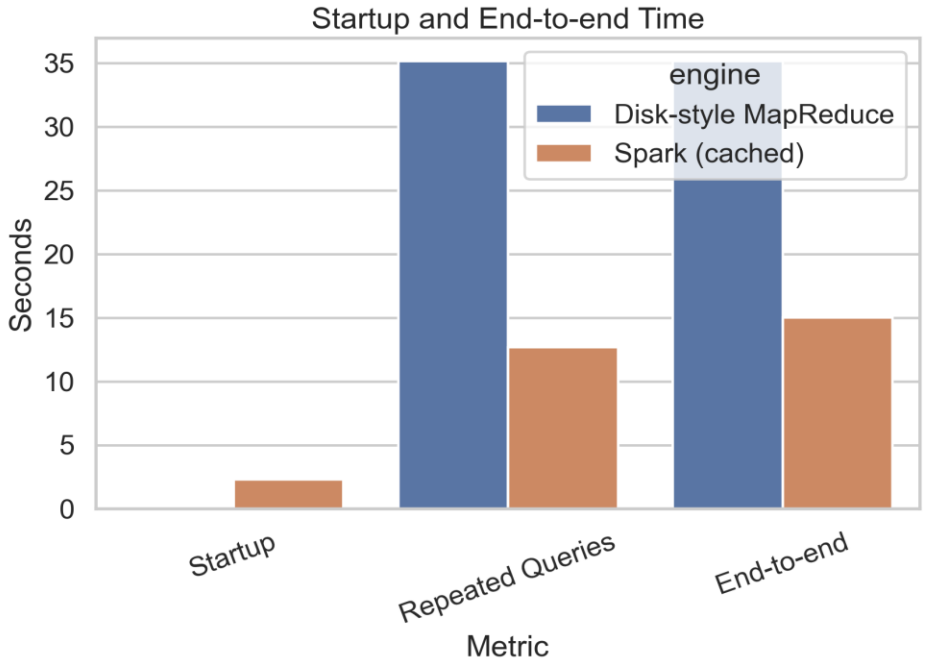


图 5 实验三设计流程图

4.4 实验三结果：Spark 缓存降低重复查询成本

图 6 磁盘计算与 Spark 内存计算在 5 轮重复分析任务中的时延对比。





- 磁盘计算 5 轮总时长为 35.18 s。
- Spark 缓存准备时间为 2.33 s。
- Spark 5 轮重复查询总时长为 12.71 s。
- Spark 端到端总时长为 15.04 s。
- 整体加速比为 2.34 倍。

Spark 的启动和缓存成本没变，但在多轮重复分析中，后续查询能够持续复用缓存，从而降低总消耗。

5. 讨论：三类机制对应三类瓶颈

YARN 解决调度控制热点问题。

JobTracker 集中承担过多任务级状态YARN 通过 ResourceManager 和 ApplicationMaster 分工降低中心压力。

HDFS Federation 解决元数据热点问题。

单 NameNode 面对集中请求和目录元数据压力 Federation 通过多个命名空间分摊元数据服务。

Spark 解决重复 I/O 热点问题。

MapReduce 多轮分析需要反复读写磁盘Spark 通过缓存复用降低后续轮次成本。

Hadoop 2.0 之后的演进主线，是**减少集中式热点**并**提升资源利用效率**。

7. 结论

结论一：YARN 通过控制平面解耦和 container 资源模型提升高并发调度能力。

结论二：HDFS Federation 通过命名空间拆分缓解单 NameNode 元数据瓶颈。

结论三：Spark 通过工作集缓存降低重复分析任务中的 I/O 开销。

Hadoop 2.0 之后的大数据平台演进，核心在于调度控制、命名空间管理和数据复用方式的重构。

附录1. 参考文献

[1] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating System Design and Implementation, OSDI '04, pages 137–150. USENIX Association, 2004.

[2] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pages 1–10, 2010.

[3] Apache Hadoop. Hdfs architecture guide. https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html, 2025. Accessed:2026-04-24.

[4] Vinod Kumar Vavilapalli, Arun C. Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, Bikas Saha, Carlo Curino, Owen O'Malley, Sanjay Radia, Benjamin Reed, and Eric Baldeschwieler. Apache hadoop yarn: Yet another resource negotiator. In Proceedings of the 4th Annual Symposium on Cloud Computing, pages 5:1–5:16, 2013.

[5] Apache Hadoop. Writing yarn applications. https://hadoop.apache.org/docs/table/hadoop-yarn/hadoop-yarn-site/WritingYarnApplications.html, 2025. Accessed: 2026-04-24.

[6] Apache Hadoop. Hdfs federation. https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/Federation.html, 2025. Accessed: 2026-04-24.

[7] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud '10, page 10.USENIX Association, 2010.

[8] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Mur_x0002_phy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient dis_x0002_tributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Im_x0002_plementation, NSDI '12. USENIX Association, 2012.

[9] Apache Hadoop. Class jobtracker. https://hadoop.apache.org/docs/r1.2.1/api/org/apache/hadoop/mapred/JobTracker.html, 2013. Accessed: 2026-04-24.

[10] Apache Hadoop. Class tasktracker. https://hadoop.apache.org/docs/r1.2.1/api/org/apache/hadoop/mapred/TaskTracker.html, 2013. Accessed: 2026-04-24.

谢谢聆听！

Q & A

数据科学学院 海南 B 班
智能科技与服务专业（民族大学合办）
张心悦 D23090600661
2026 年 4 月 24 日