Though I provide a pool of available projects as follows, you are strongly encouraged to come up a topic of your own.

Real-world Enterprise Trace Collection

Google Trace: C. Reiss, J. Wilkes, and J. L. Hellerstein. Google Clusterusage Traces: Format + Schema. Technical report, Google Inc., November 2011.

TPC-DS: http://www.tpc.org/tpcds/

TPC-H: http://www.tpc.org/tpch/

BigBench: http://www.msrg.org/publications/pdf_files/2013/Ghazal13-BigBench:_Towards_an_Industry_Standa.pdf

1. Project 1: Tolerate Long-Tail Latency in Apache Flume Using Distributed Deadline Propagation
   - Run google trace in apache flume using “Stage-Service model” (see D2P paper)
   - Experimentally show the long-tail latency problem
   - Solve the problem: try the the D2P paper’s distributed deadline propagation approach

   https://flume.apache.org/

   Paper to refer: http://asg.ict.ac.cn/baoyg/downloads/APSys14-D2P.pdf

2. Project 2: Tolerate Long-Tail Latency in Apache Flume Using Replication and Reconnecting
   - Run google trace in apache flume using “Stage-Service model” (see D2P paper)
   - Experimentally show the long-tail latency problem
   - Solve the problem: try the hot/cold replication approach, the dynamically changing links approach that we discussed in class

   https://flume.apache.org/

   Paper to refer: http://asg.ict.ac.cn/baoyg/downloads/APSys14-D2P.pdf

3. Project 3: Live Migration of Docker Container
    - Install data analysis systems in Docker containers (e.g., Spark Streaming, MapReduce, Storm, Dryad, Graph-related systems)
    - Run Grep, WordCount, TopKCount, Breadth-First Search(BFS), PageRank(PR), Connected Components(CC) or Triangle Counting(TC) application in real-world datasets such as Twitter, Livejournal, USAroad, or Rmat24 in those systems
    - Kill one container, then migrate the failed container to another host to restart, show the performance loses.
    - Kill more than one container each time, show the performance loses.
    - Compare different applications' performance loses due to live migration

    https://criu.org/Live_migration

    Paper to refer: Supreeth Subramanya, Tian Guo, Prateek Sharma, David Irwin, and Prashant Shenoy. 2015. SpotOn: a batch computing service for the spot market. In *Proceedings of the Sixth ACM Symposium on Cloud Computing* (SoCC '15). ACM, New York, NY, USA, 329-341.

4. Project 4: Hot Replication of Docker Container
    - Install data analysis systems in Docker containers (e.g., Spark Streaming, MapReduce, Storm, Dryad, Graph-related systems)
    - Run Grep, WordCount, TopKCount, Breadth-First Search(BFS), PageRank(PR), Connected Components(CC) or Triangle Counting(TC) application in real-world datasets such as Twitter, Livejournal, USAroad, or Rmat24 in those systems
    - Kill one container, use hot replication mechanism to restart, show the performance loses
    - Kill more than one container each time, show the performance loses
    - Compare different applications' performance loses due to hot replication

    https://criu.org/Live_migration (use the checkpointing for hot replication)

    Paper to refer: Supreeth Subramanya, Tian Guo, Prateek Sharma, David Irwin, and Prashant Shenoy. 2015. SpotOn: a batch computing service for the spot market. In *Proceedings of the Sixth ACM Symposium on Cloud Computing* (SoCC '15). ACM, New York, NY, USA, 329-341.

5. Project 5: Performance Comparison of Container Management Tools

   - Choose tools from https://www.weave.works/blog/11-docker-tools-developers/ (at least three, and one of it has to be Kubernetes)
   - Select a trace from trace collection to run
   - Compare the performance and attribute the performance differences to architectural differences

   Paper to refer: http://www.vldb.org/pvldb/vol8/p2110-shi.pdf

6. Project 6: TC Control of Docker Container

   - Install data analysis systems in Docker containers (e.g., Spark Streaming, MapReduce, Storm, Dryad, Graph-related systems)
   - Run Grep, WordCount, TopKCount, Breadth-First Search(BFS), PageRank(PR), Connected Components(CC) or Triangle Counting(TC) application in real-world datasets such as Twitter, Livejournal, USAroad, or Rmat24 in those systems
   - Traffic control of the bandwidth between one container to another container (reduces the bandwidth from 1 GB to 100 MB to 10 MB)
   - Compare different applications' performance loses due to traffic control

   Paper to refer: http://www.istc-cc.cmu.edu/publications/papers/2012/netcohort.pdf

7. Project 7: Dynamically Identifying the Hot/Cold Blocks in HDFS
   - Analyze BigBench trace to experimentally show that hot/cold blocks scenario exists
   - Design a dynamic, online or offline approach to analyze which block in HDFS is hot, and the number of replica and where should be the new replica placed on.

   Paper to refer: Scarlett: Coping with Skewed Content Popularity in MapReduce Clusters.

8. Project 8: Dynamically Identifying the Hot/Cold Blocks in HDFS and Using Memcached to Solve it
    - Analyze BigBench trace to experimentally show that hot/cold blocks scenario exists
    - Use memcached to store extra replica of hot files to improve the performance

    https://memcached.org/
    Paper to refer: Scarlett: Coping with Skewed Content Popularity in MapReduce Clusters.

9. Project 9 (Challenging): Bug analysis: Understanding Issue Correlations:
    A Case Study of the Spark Streaming System (or Docker)

    - If you choose this project, you need to be QUITE familiar with the inside of spark streaming system (or Docker) (source code, design, and implementation)
    - You need to survey a lot of work, read every bug so far, and manually analyze them to think deeply about the occurrence of the bug (e.g., relates to new case scenario or relates to system design or relate to other) which it is a huge amount of work

    Paper to refer: Understanding Issue Correlations: A Case Study of the Hadoop System
    An Evolutionary Study of Linux Memory Management for Fun and Profit