Scalable Web Programming

CS193S - Jan Jannink - 2/25/10

Weekly Syllabus

1. Scalability: *(Jan.)*
2. Agile Practices
3. Ecology/Mashups
4. Browser/Client
5. Data/Server: *(Feb.)*
6. Security/Privacy
7. Analytics
8. **Cloud/Map-Reduce**
9. Published APIs: *(Mar.)**
10. Future

*** PROJECT DUE DATE**

Cloud Recap

- Progressive commoditization of IT services
- Choose based on value creation in project
- Build it, Scale it, Code it, Customize it
  - Google has most efficient data centers
  - Ning, SocialGo offer customizable communities

Server & Data Scaling

- Traditionally depended on next hardware release
- AltaVista search engine
    - limited to the most expensive DEC Alpha box
- Original eBay build-out
    - massive SUN/Teradata clusters

Map/Reduce

- Background
  - Google founders’ disdain for traditional RDBMS
- Original paper published 5 years ago
- Main Features
  - limitless scalability on cheap hardware
  - real time fault tolerance

Google Example

- Key ‘contrarian’ insight
  - scaling on cheap hardware is best
  - need generic API to divide and conquer
- If McDonald’s needed a top chef in each store
  - how much more would it cost?
  - how many restaurants would it now have?

Hadoop

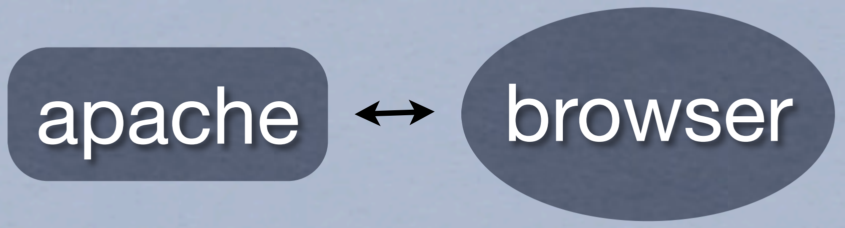
- First Open Source implementation
  - by Doug Cutting also of Lucene fame
- *Storage*, **Execution**, Management components
  - *HDFS*, HBase, Hive
  - **MapReduce**
  - Pig, ZooKeeper

1-2-3

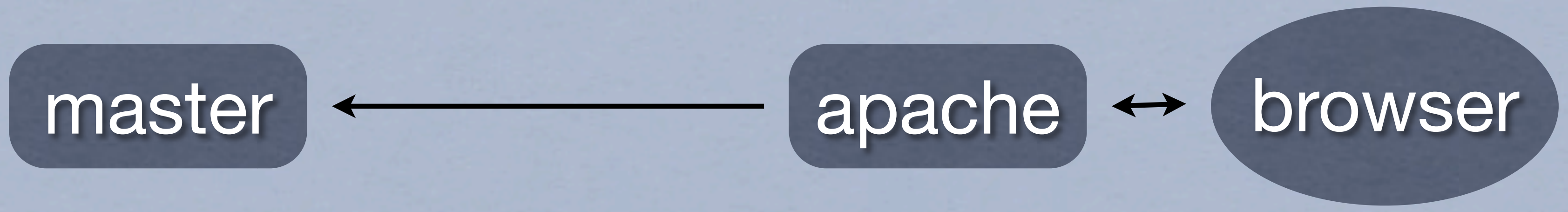
- Configure server clusters
- Code Map & Reduce classes
  - input list of (key, value) pairs
  - output set of (key, value) pairs
- Start HDFS, MapReduce servers

Execution Schema

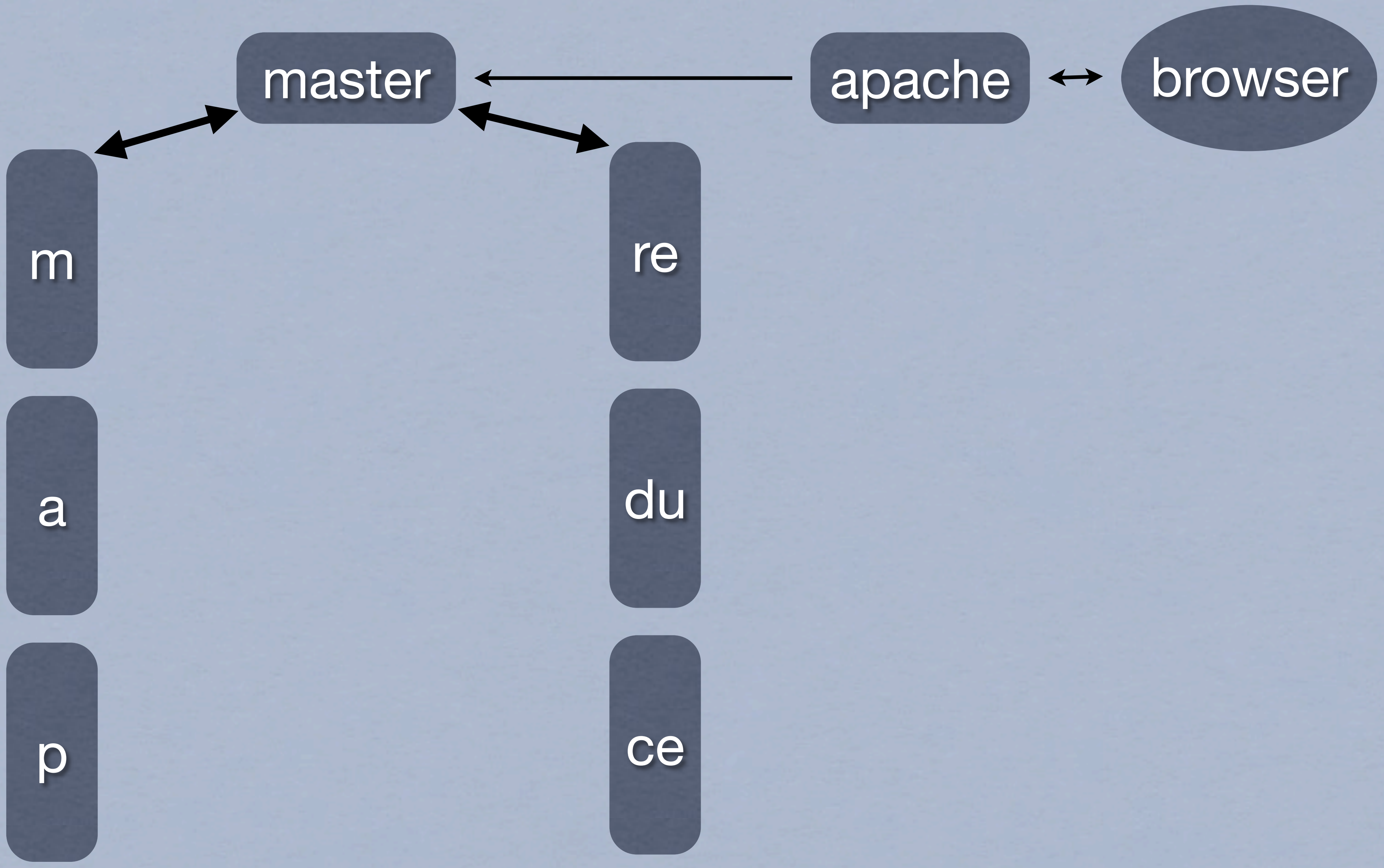
Execution Schema



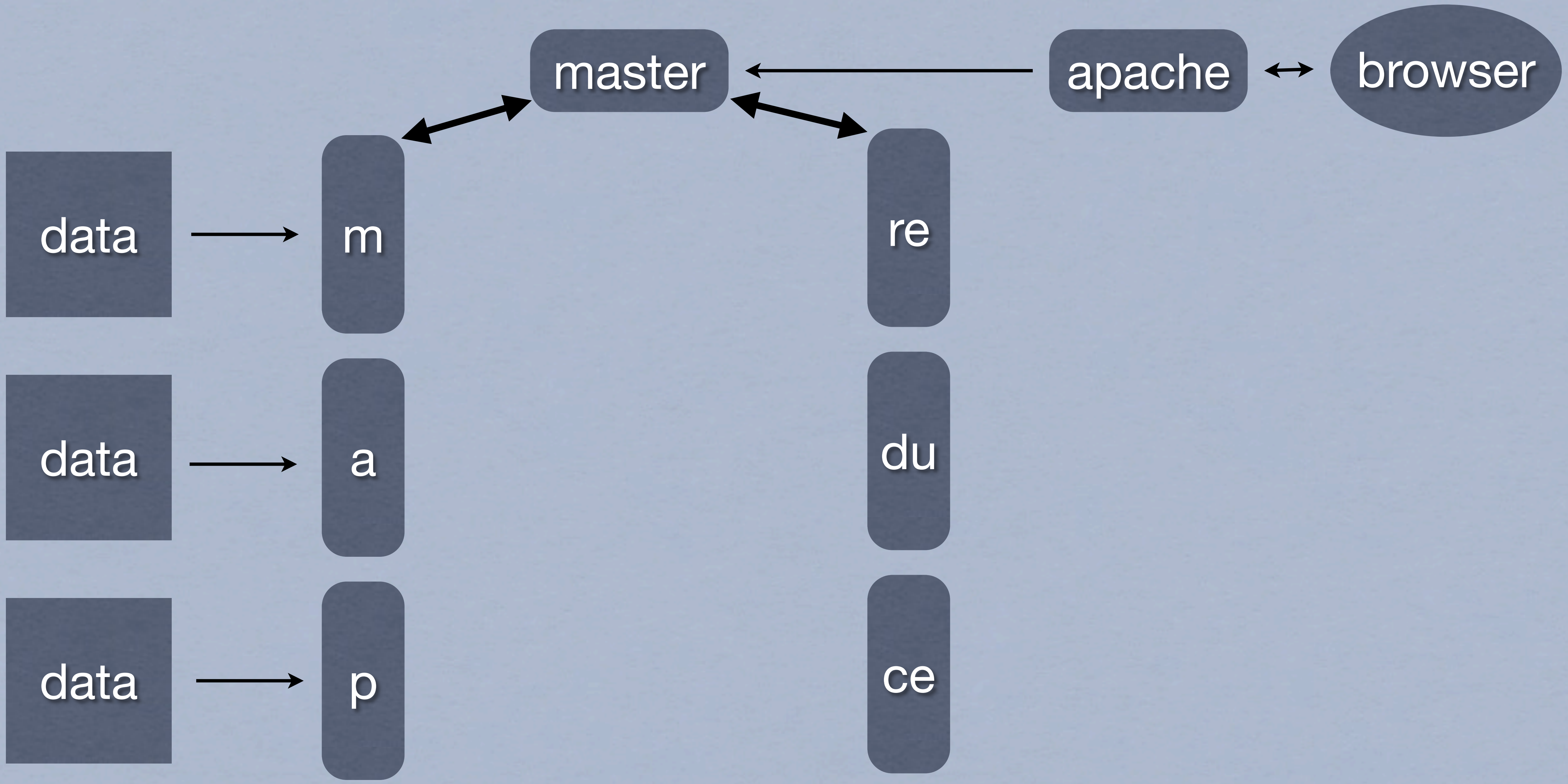
Execution Schema



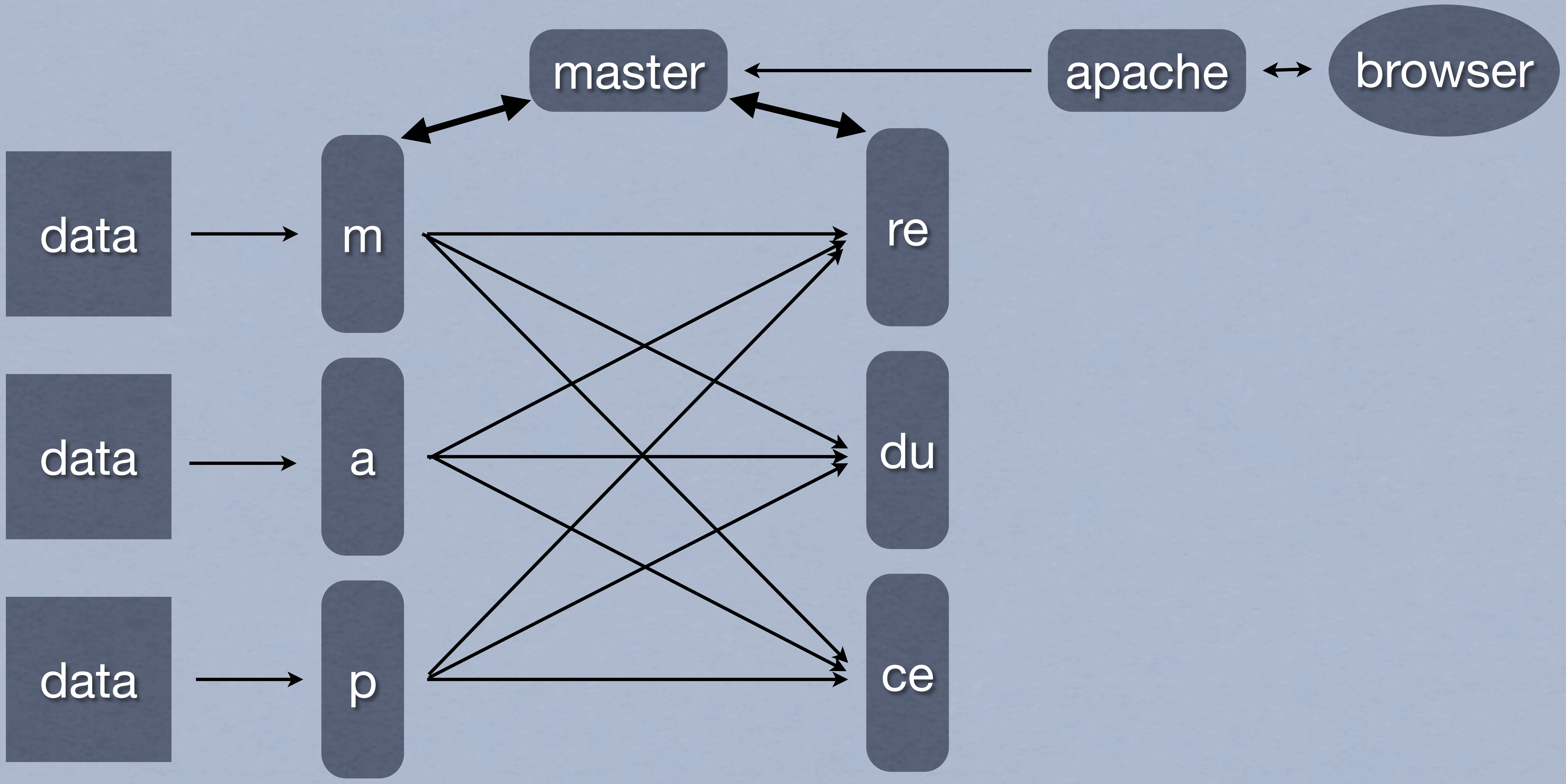
Execution Schema



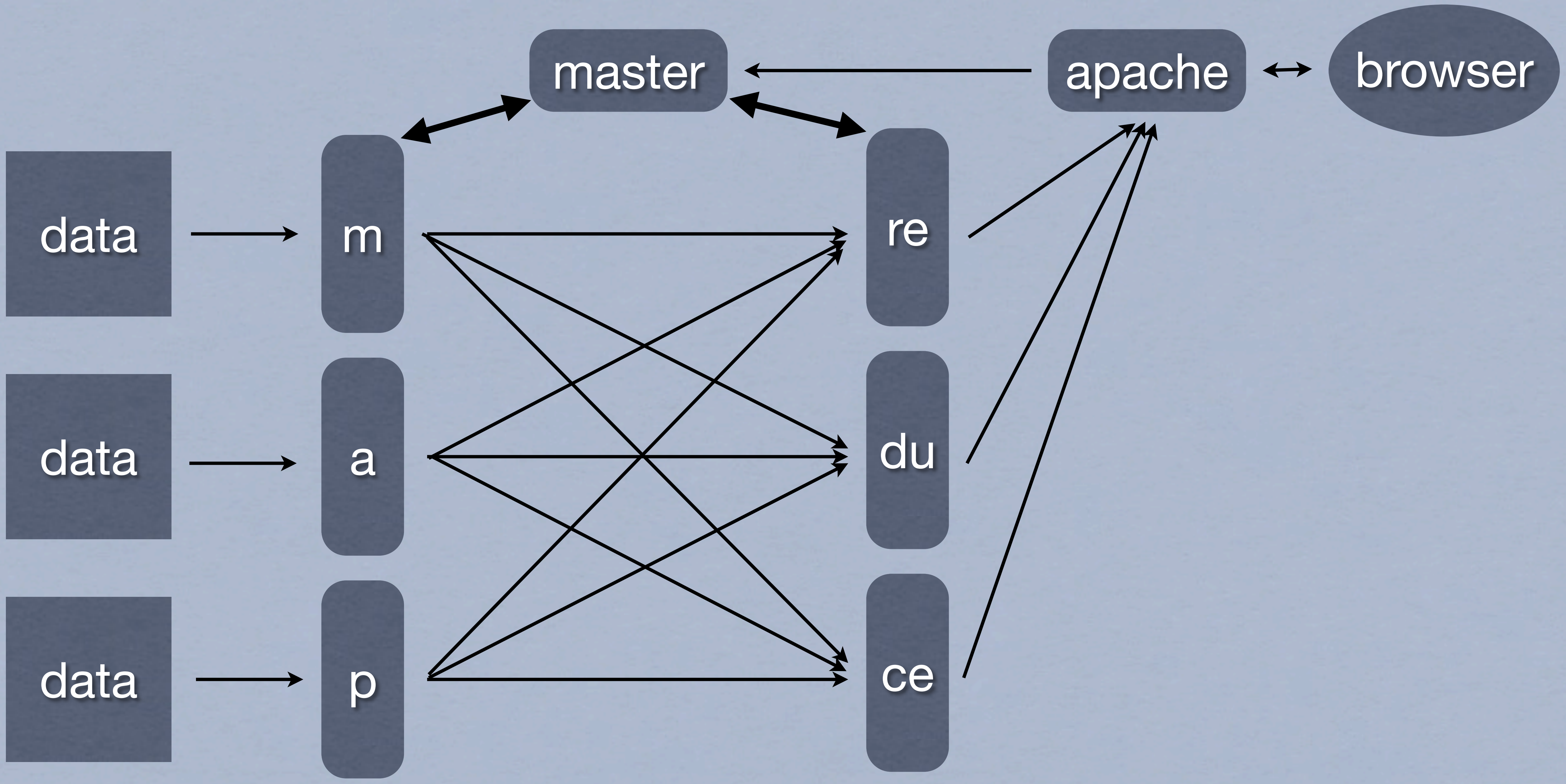
Execution Schema



Execution Schema



Execution Schema



Example: Word Count

- cat data.txt | sort | uniq -c
- When data.txt is huge
  - split it into even chunks
  - sort chunks (or better yet, prefix sort)
  - count on a per item/prefix basis
  - return result

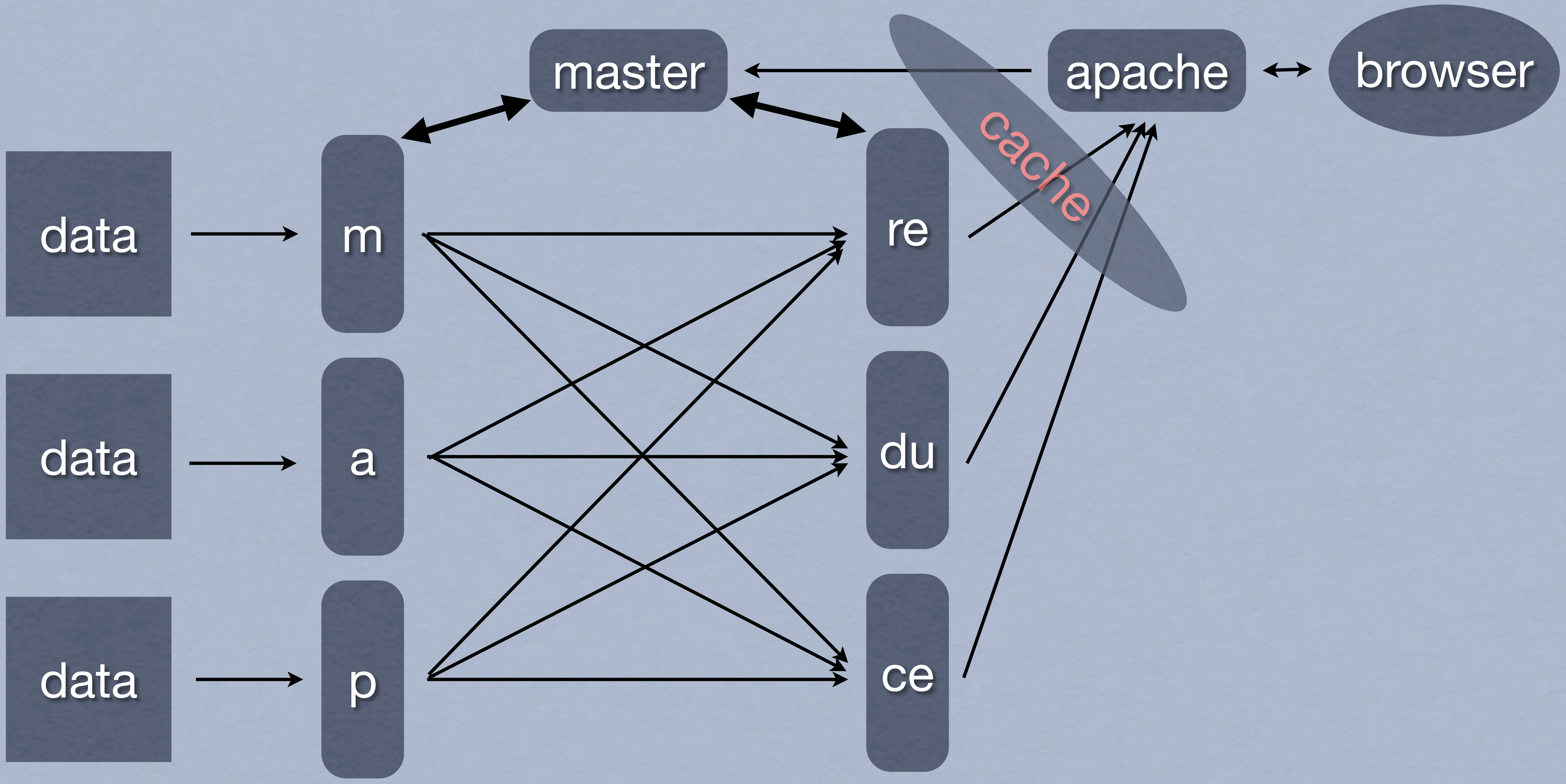
Insights

- Relates to network switching, routing concepts
- DB queries can generate initial (key, value) pairs
- Master restarts failed map, reduce tasks
  - ping nodes after first results start coming in
- Rule of thumb
  - data / 64MB ≅ # mappers > # reducers

Additional Details

- Storing intermediate results
    - RAM/File systems on mappers, reducers
- Can combine, e.g., duplicate removal, on mappers
- Partition mapper results by key for reducers
- Backup masters possible (not frequently used)
- Caching becomes a critical system component

Caching is a Huge Win



Current Status

- Current production development at Google
  - entirely Map/Reduce
- Yahoo runs ~4000 node Hadoop cluster
  - 100TB data sorting record < 3 hours
- Facebook has ~1000 node Hadoop install
- Amazon offers Elastic MapReduce

Controversy?

- DB community on both sides of argument
  - DeWitt, Stonebraker: a giant step backwards
  - Abadi: HadoopDB
- In web apps, however
  - DBs are used for persistence, not transactions
  - MapReduce provides scale and fault tolerance

Future Directions

- AsterData, Google
    - hybrid Map/Reduce DBs, Data Warehouses
- HadoopDB
    - open source PostgreSQL & Hadoop hybrid
- NoSQL movement

Data Glut

- Over 3 million English Wikipedia articles
  - 1000+ new articles a day
  - unthinkable to use without search
- Facebook Data Warehouse adds 15 TB a day
  - in 2007 it was 15 TB total
- Google has several multi Petabyte data stores

Here to Stay

- Easy to learn, easy to maintain
    - very incremental learning curve
- Scales as fast as data is growing
    - only option for large data mining tasks
- Accommodates multiple persistence backends

Worth Checking Out

- Hadoop
  - http://hadoop.apache.org/
- Wikimedia Report Card
  - http://stats.wikimedia.org/reportcard/
- Data blog
  - http://www.dbms2.com/

Q & A Topics

- Merging DB optimizers and Map/Reduce
- Managing multi stage Map/Reduce pipelines
- Map/Reduce and virtual machines