Search in the Cloud

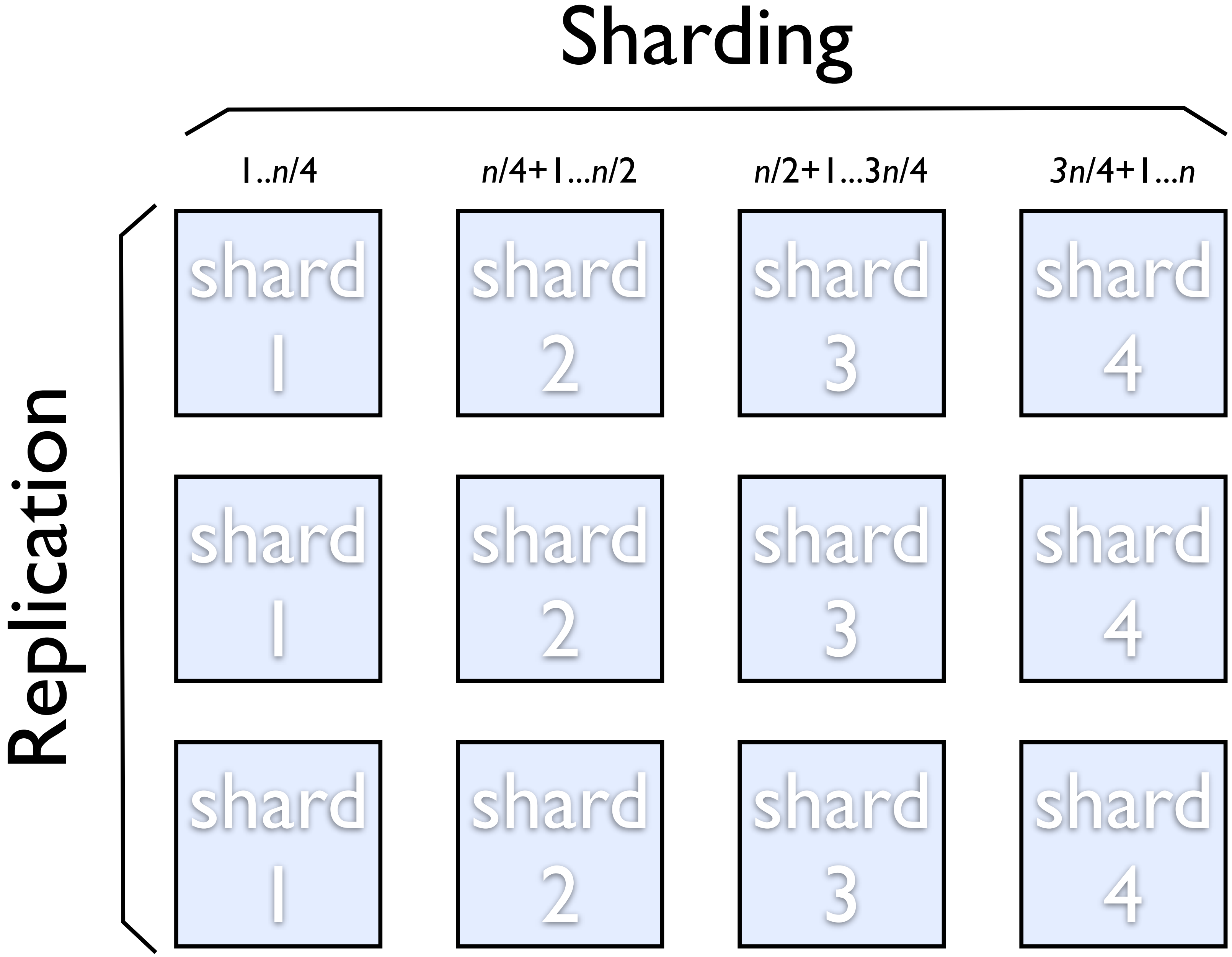
Text Retrieval Task

- Text viewed as a sequences of terms in fields
- Document and position for each term are indexed
- Query is a sequence of terms (typically many more than user actually types)

Text Retrieval

- Scores computed by merging occurrences of terms in query
- Only top scoring documents are kept
- Deletion and document edits done by adding new documents and keeping deletion list

Traditional Scaling

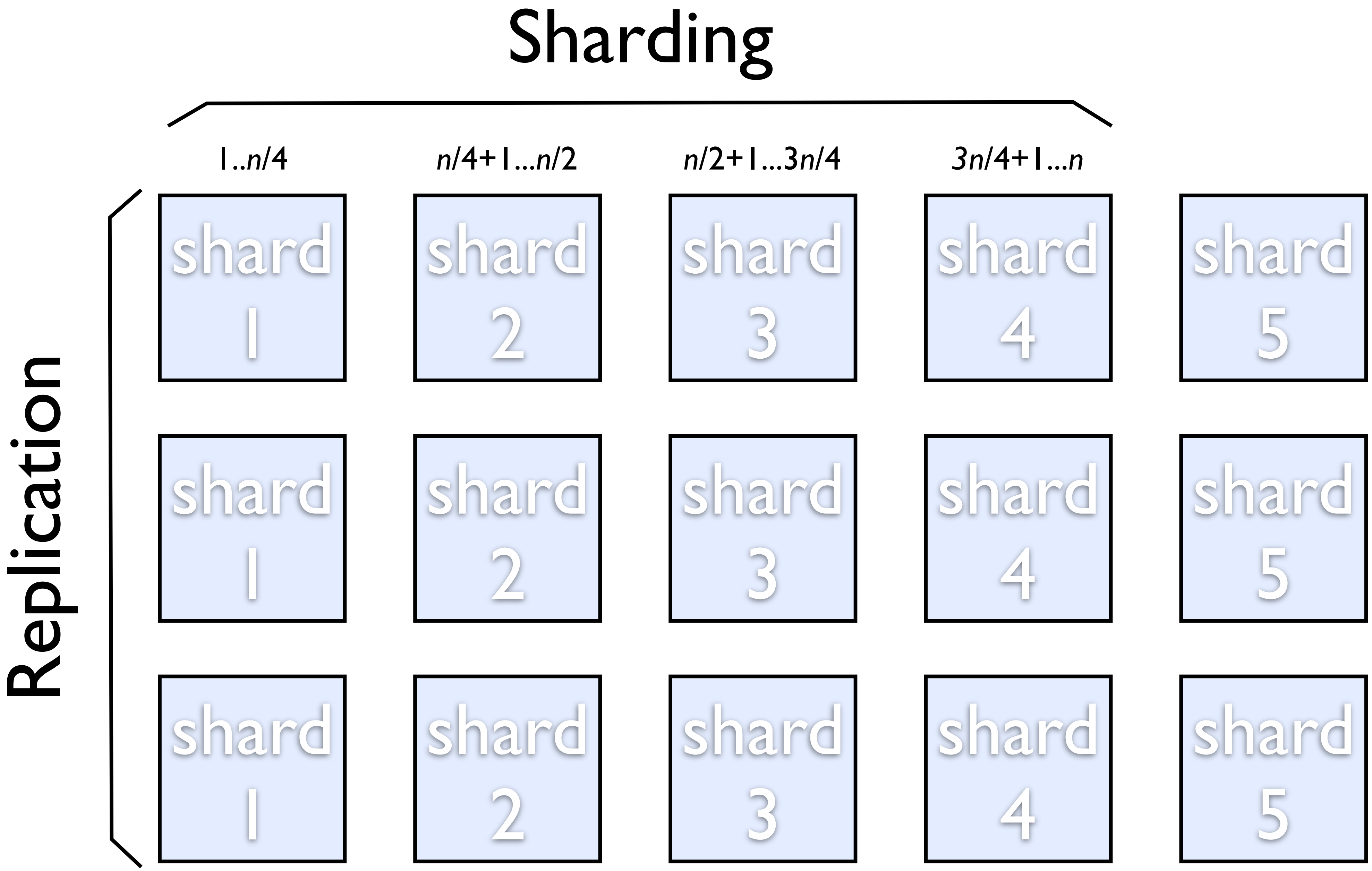


Sharding

Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4

Traditional Scaling

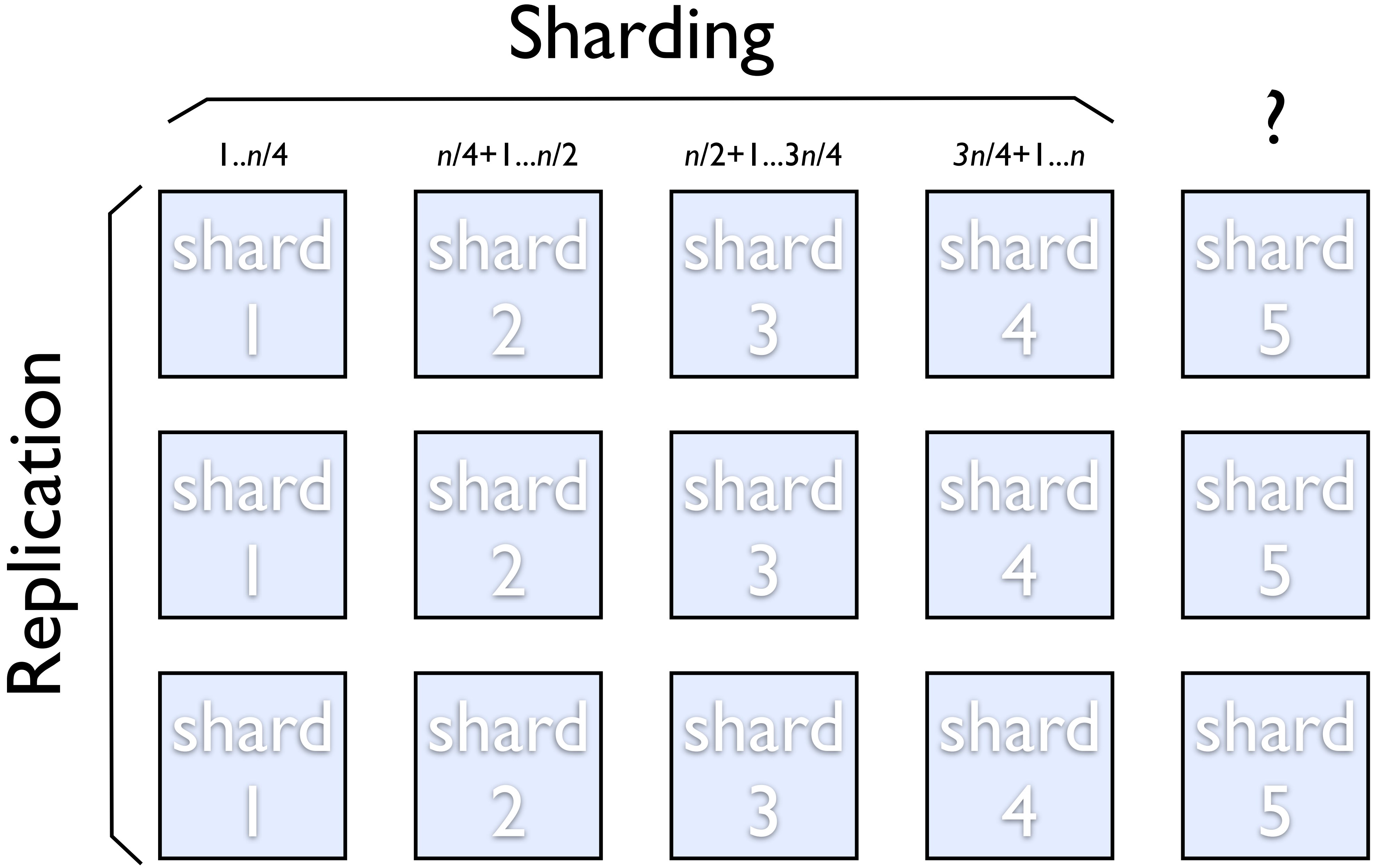


Sharding

Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*	
shard 1	shard 2	shard 3	shard 4	shard 5
shard 1	shard 2	shard 3	shard 4	shard 5
shard 1	shard 2	shard 3	shard 4	shard 5

Traditional Scaling

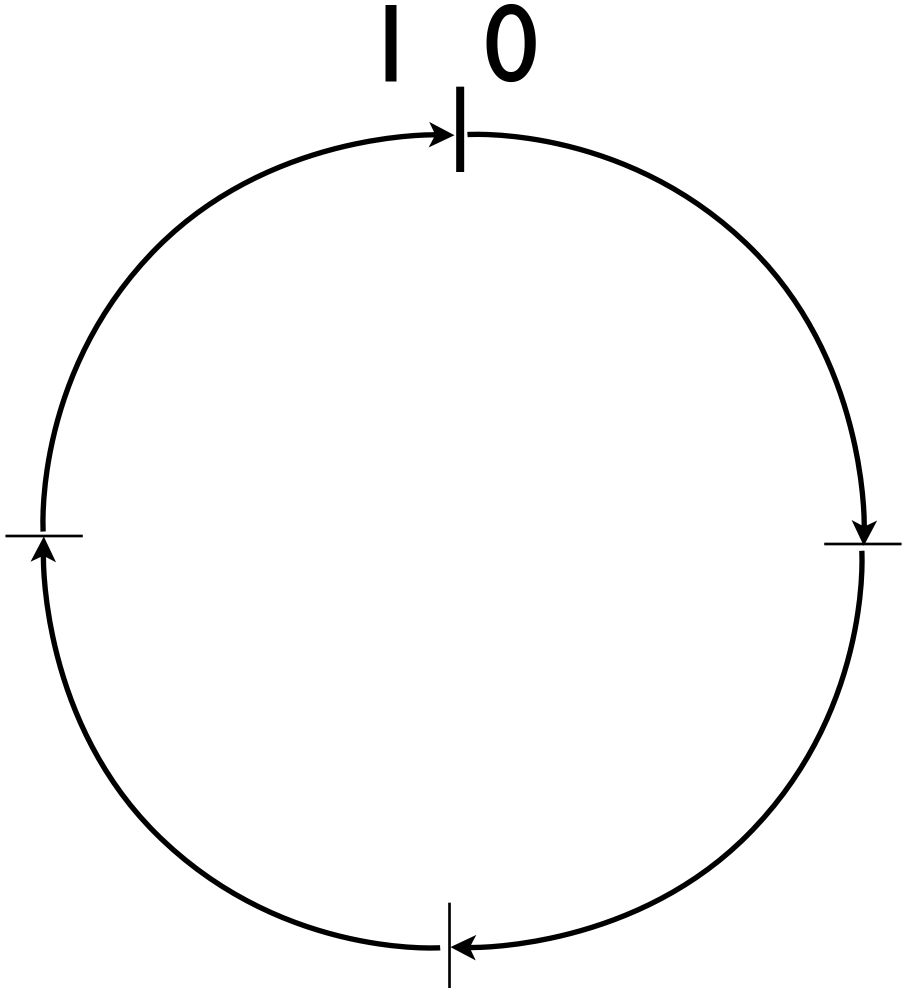


Sharding

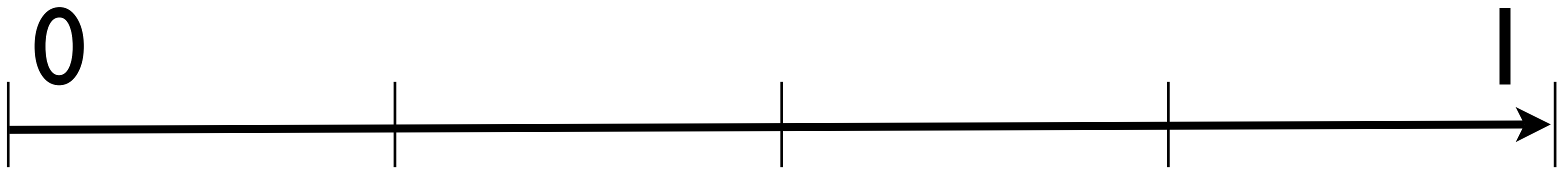
Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*	?
shard 1	shard 2	shard 3	shard 4	shard 5
shard 1	shard 2	shard 3	shard 4	shard 5
shard 1	shard 2	shard 3	shard 4	shard 5

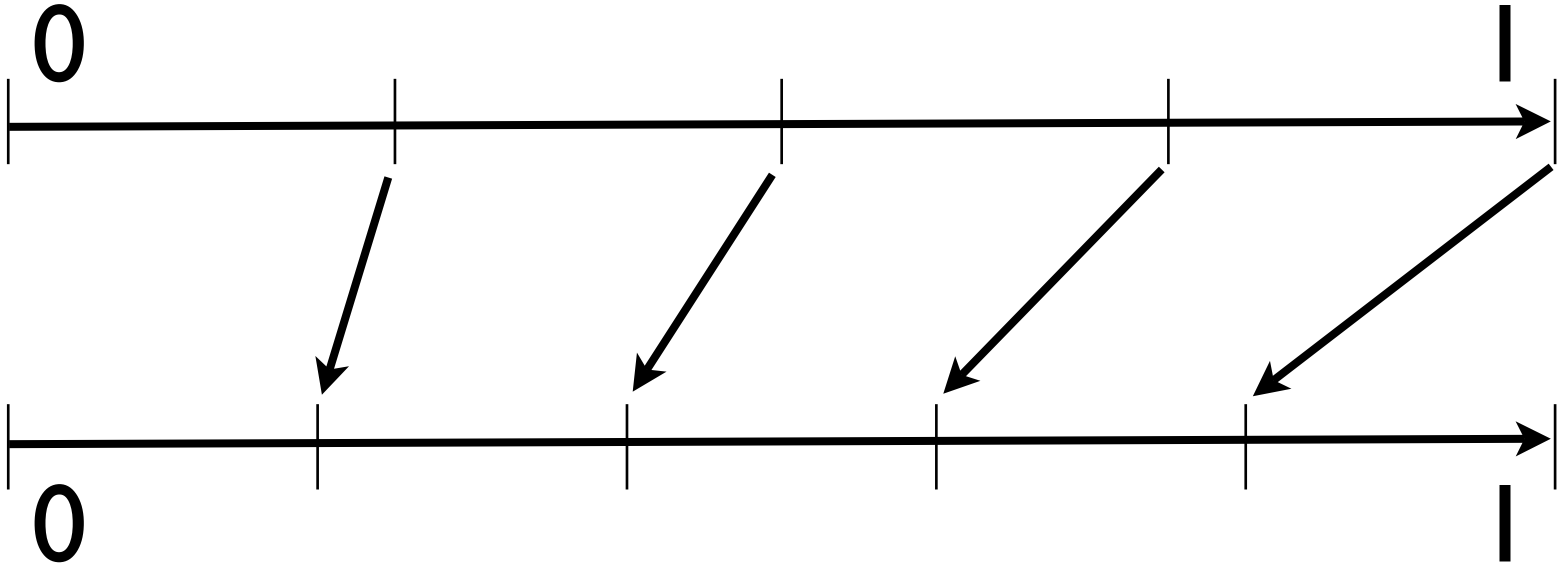
Consistent Hashing



Consistent Hashing



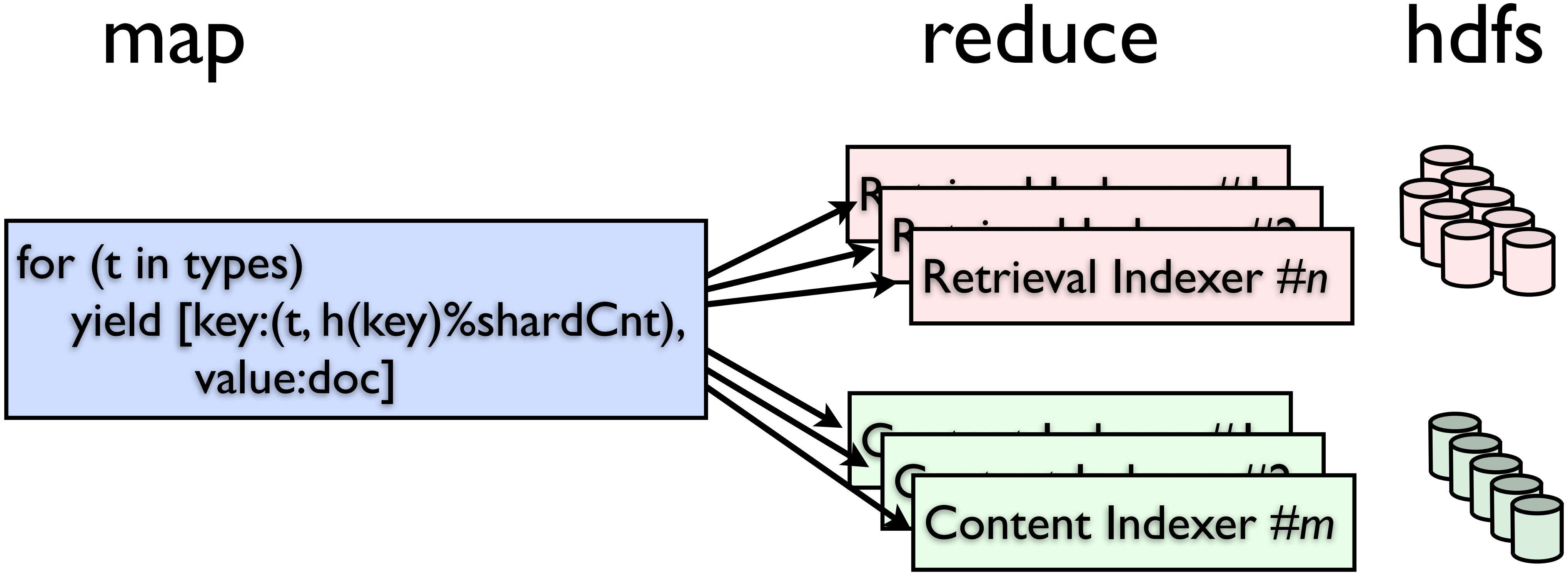
Consistent Hashing



Problems

- Presumes objects can be moved individually
- Has very high insertion/deletion rate
- Has disordered access patterns
- Often exhibits content/placement correlations

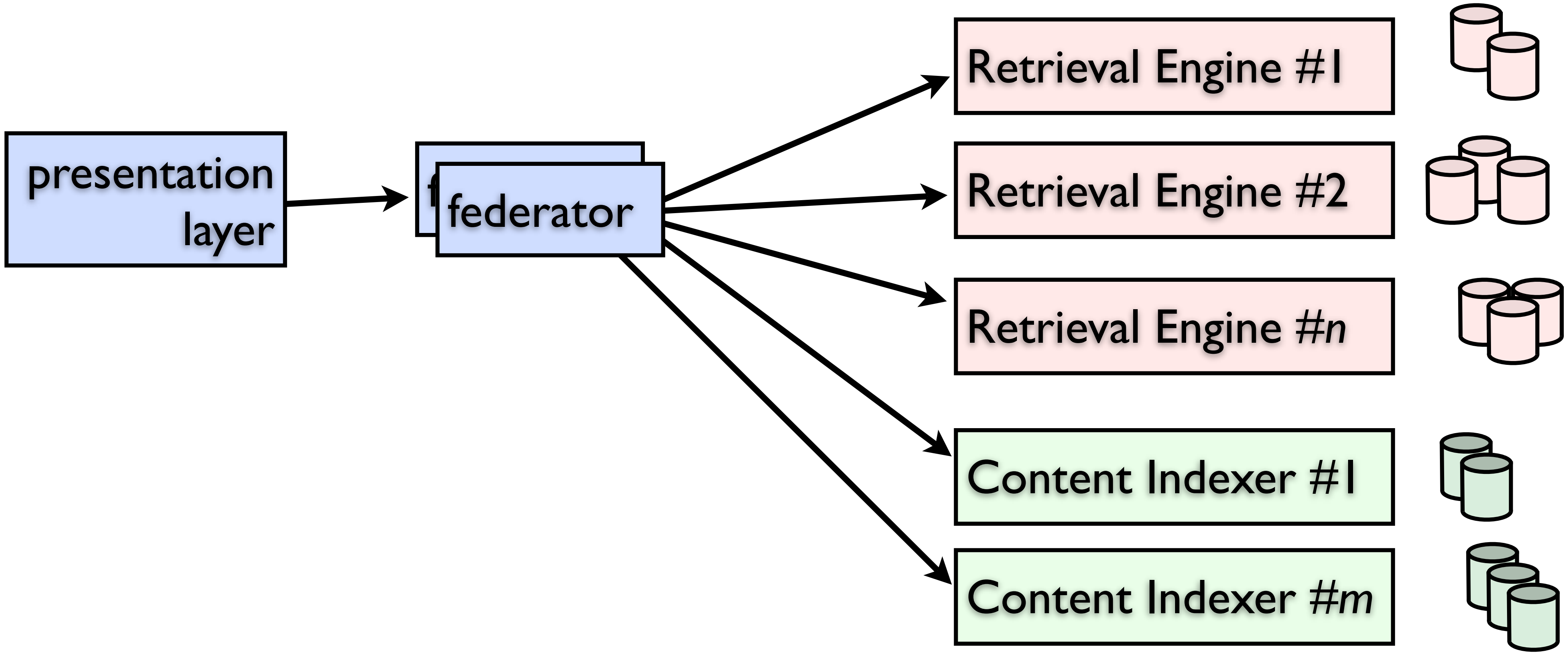
Micro Sharding



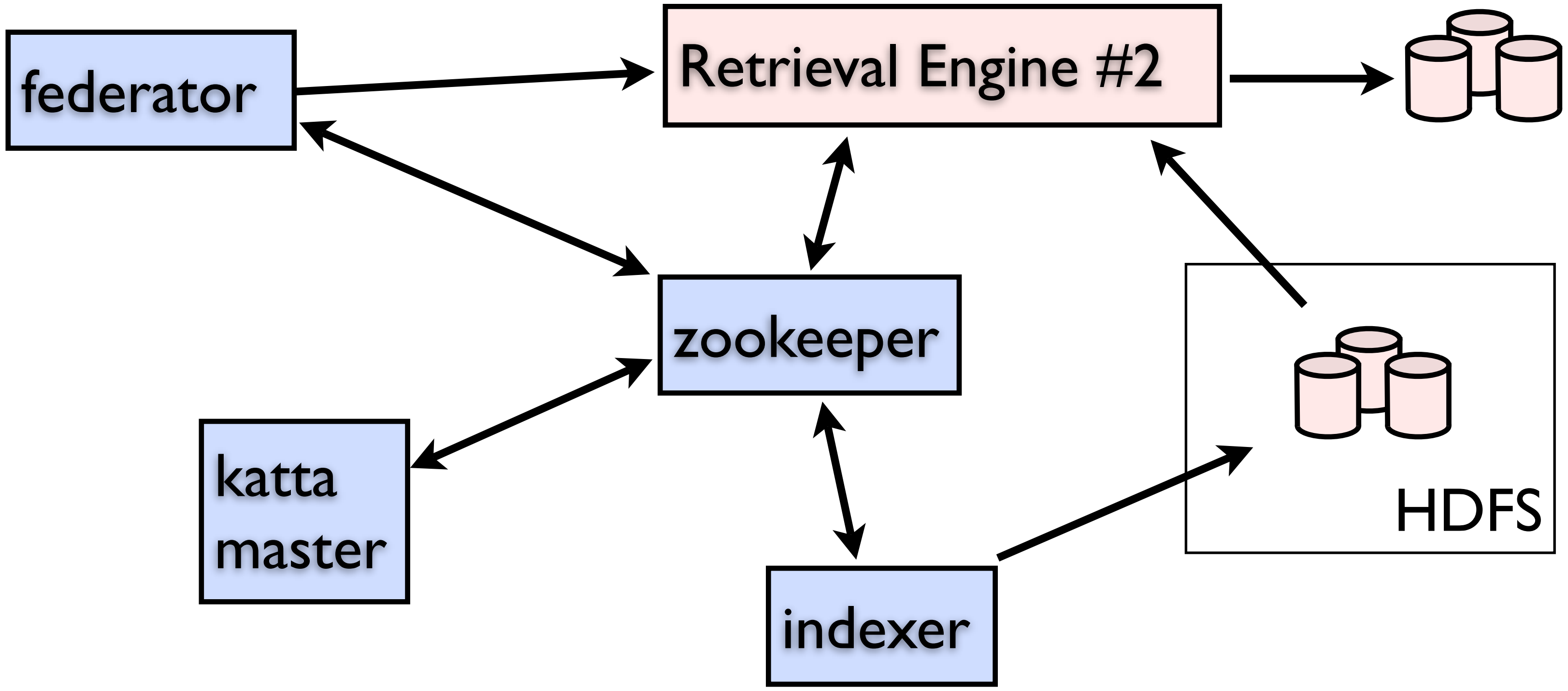
```
for (t in types)
    yield [key:(t, h(key)%shardCnt),
            value:doc]
```

n,m >> number of search nodes

Search Architecture



Control Architecture



Quick Results

- No deletion/insertion in indexes at runtime
- Reloading micro-shards allows large sequential transfers
- Random placement guided by balancing policy gives near optimal motion
- Node addition and failure are simple, reliable
- Random sharding also near optimal
  - local = global statistics, 2x query time improvement
  - load balancing
  - uniform management

Building Blocks

- EC2 - elastic compute
- Zookeeper - reliable coordination
- Katta - shard and query management
- Hadoop - map-reduce, RPC for Katta
- Lucene - candidate set retrieval, index file storage
- Deepdyve search algorithms - segment scoring

Building Blocks

- EC2 - elastic compute
- **Zookeeper - reliable coordination**
- Katta - shard and query management
- Hadoop - map-reduce, RPC for Katta
- Lucene - candidate set retrieval, index file storage
- Deepdyve search algorithms - segment scoring

Zookeeper

- Replicated key-value in-memory store
- Minimal semantics
  create, read, replace specified version
  sequential and ephemeral files
  notifications
- Very strict correctness guarantees
  strict ordering
  quorum writes
  no blocking operations
- High speed
  50,000 updates per second
  200,000 reads per second

Building Blocks

- EC2 - elastic compute
- Zookeeper - reliable coordination
- **Katta - shard and query management**
- Hadoop - map-reduce, RPC for Katta
- Lucene - candidate set retrieval, index file storage
- Deepdyve search algorithms - segment scoring

Katta Interface

- Simple Interface

  Client - horizontal broadcast for query, vertical broadcast for update
  InodeManaged - add/removeShard

- Pluggable Application Interface

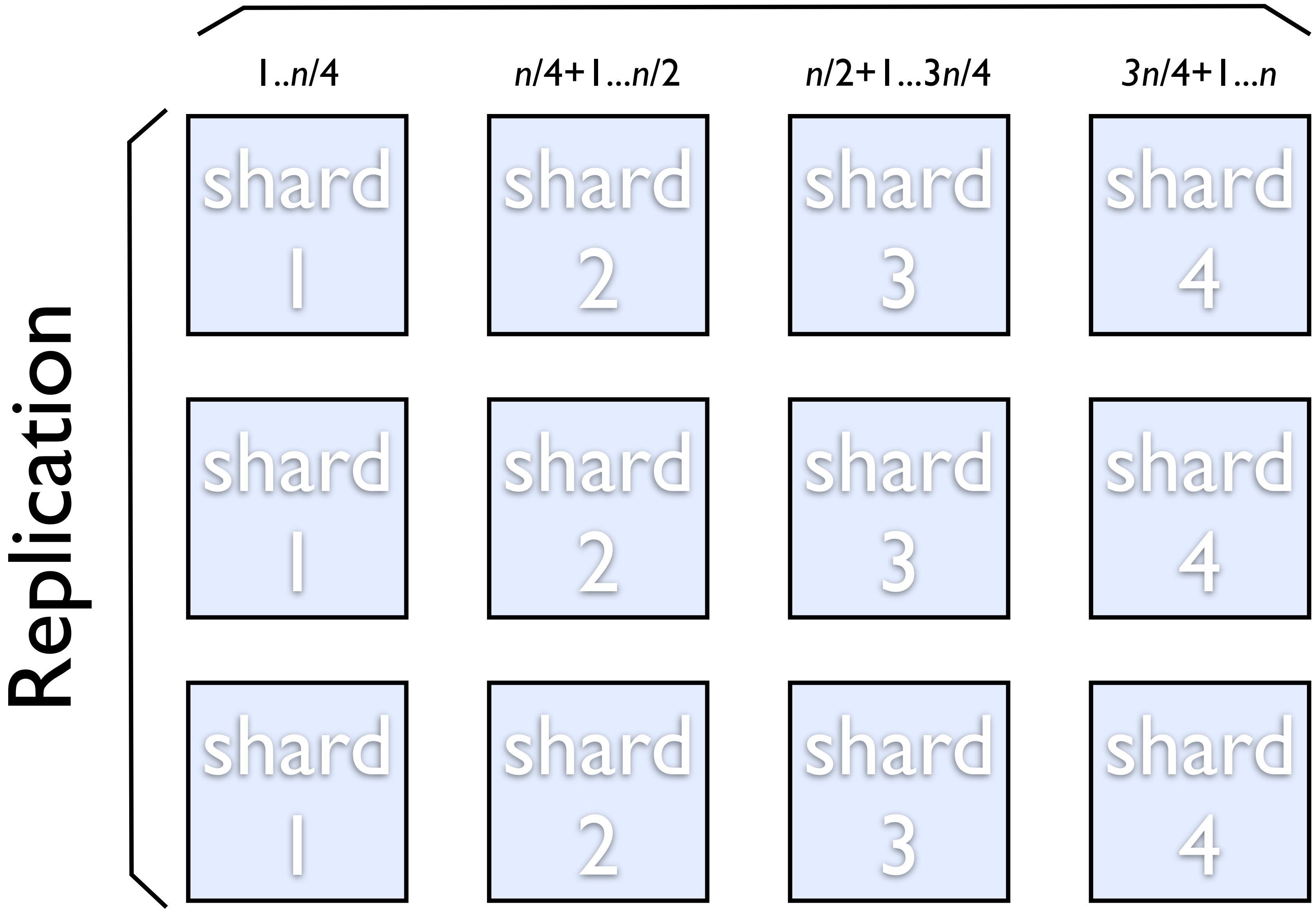
- Pluggable Return Policy

  Given current return state
  return < 0 => done
  return 0   => return result, allow updates
  return n   => wait at most n milliseconds

- Comprehensive Results

  Results, exceptions, arrival times

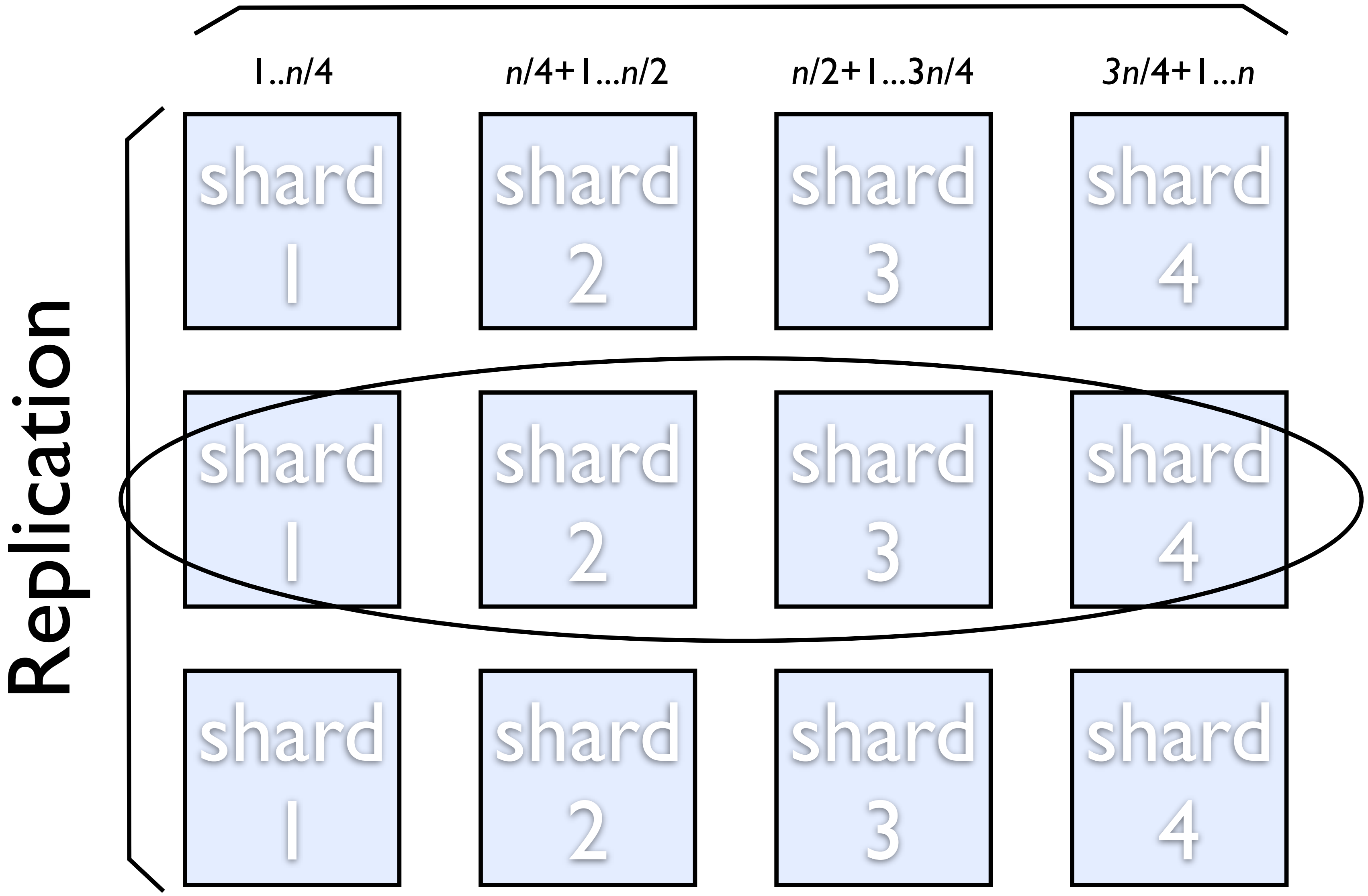
Horizontal/Vertical Broadcast



Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4

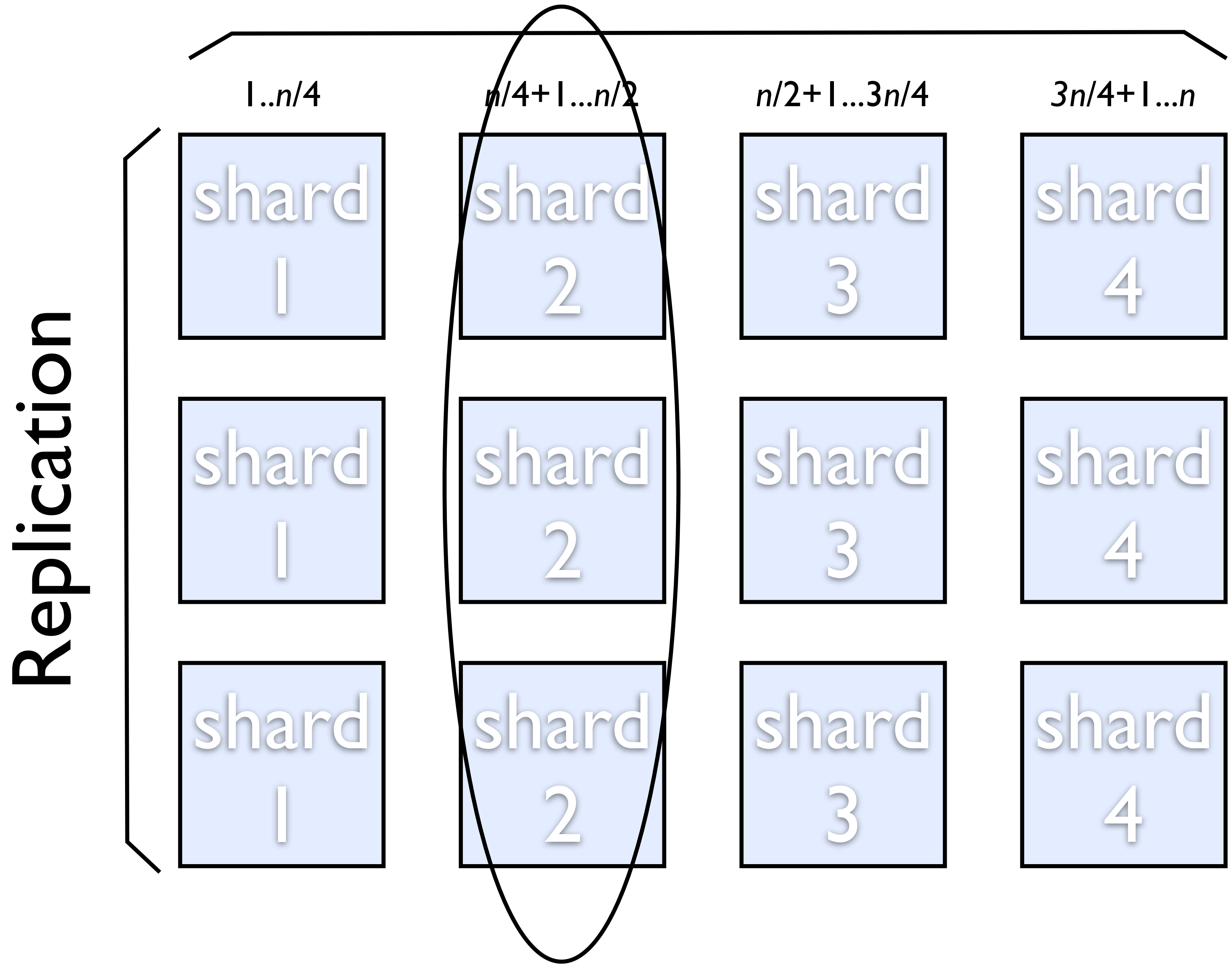
Horizontal/Vertical Broadcast



Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4

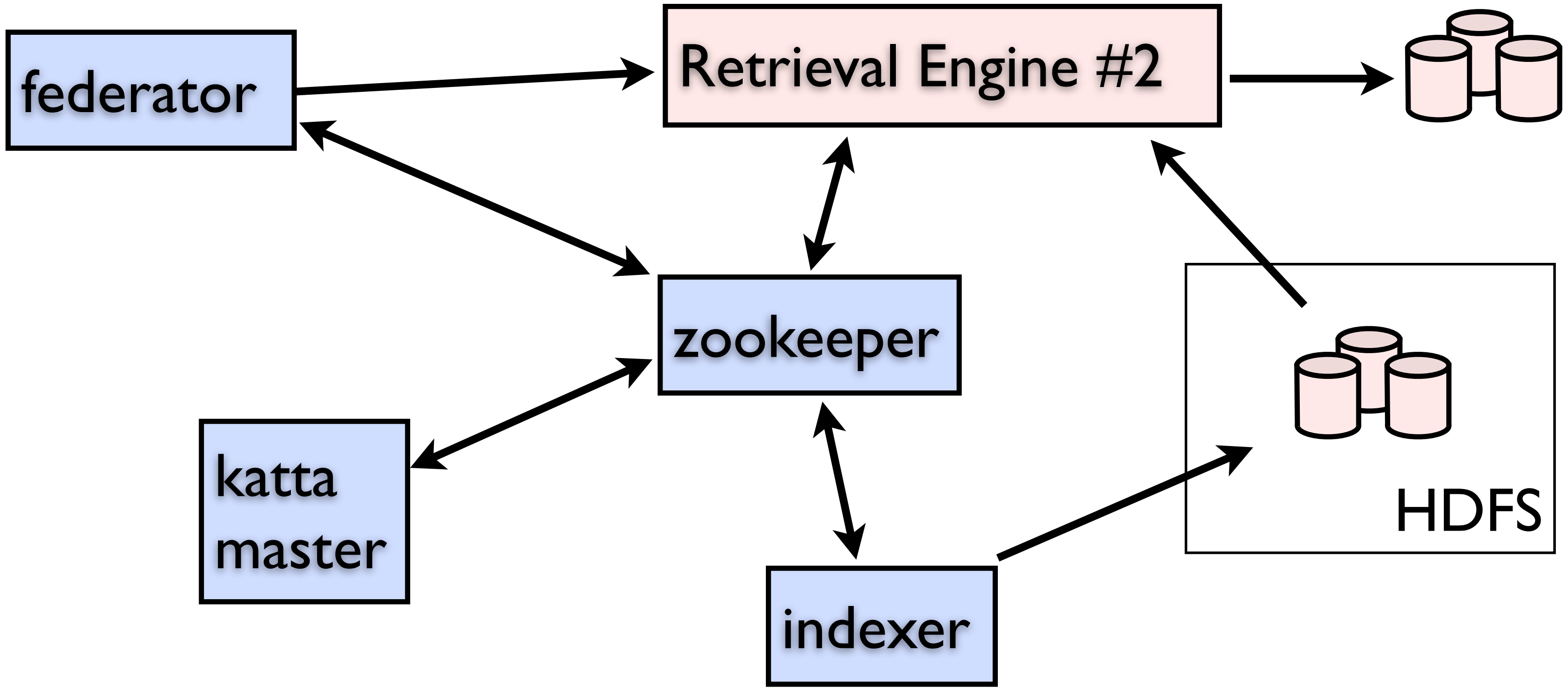
Horizontal/Vertical Broadcast



Replication

1..*n*/4	*n*/4+1...*n*/2	*n*/2+1...3*n*/4	3*n*/4+1...*n*
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4
shard 1	shard 2	shard 3	shard 4

Operations


federator
Retrieval Engine #2
zookeeper
katta master
indexer
HDFS

Impact of Cloud Approach

- Scale-free programming
- Deployed in EC2 (test) or in private farm (production)
- No single point of failure
- Real-time scale up/down
- Extensible to real-time index updates

Resources

- My blog
  - http://tdunning.blogspot.com/
- The web-site
  - www.deepdyve.com
- Source code
  - Katta (sourceforge)
  - Hadoop (Apache)
  - Lucene (Apache)