Hbase's Architecture is inspired by

Recap

HBase vs RDBMS

This is how data is stored in traditional databases

id	type	for user	from user	timestamp
1	Friend request status	Ryan	Jessica	146710201
2	Comment	Chaz	Daniel	146711200
3	Comment	Rick	Brendan	1467112205
4	Like	Rick	Brendan	1467112213

Recap

Column oriented storage

id	type	for user	from user	timestamp
1	Friend request status	Ryan	Jessica	146710201
2	Comment	Chaz	Daniel	146711200
3	Comment	Rick	Brendan	1467112205
4	Like	Rick	Brendan	1467112213

Data is stored in a map

Key = <Row id, Col id>

Value = <data>

Recap

Column oriented storage

id	type	for user	from user	timestamp
1	Friend request status	Ryan	Jessica	146710201
2	Comment	Chaz	Daniel	146711200
3	Comment	Rick	Brendan	1467112205
4	Like	Rick	Brendan	1467112213

Data is stored in a map

Key = 2, for_user

Value = Chaz

Recap

Column oriented storage

id	type	for user	from user	timestamp
1	Friend request status	Ryan	Jessica	146710201
2	Comment	Chaz	Daniel	146711200
3	Comment	Rick	Brendan	1467112205
4	Like	Rick	Brendan	1467112213

row	column	value
1	type	Friend request status
1	for user	Ryan
1	from user	Jessica
1	timestamp	146710201
2	type	Comment
2	for user	Chaz
2	from user	Daniel
2	timestamp	146711200
3	type	Comment
3	for user	Rick
3	from user	Brendan
3	timestamp	1467112205

Recap

Column oriented storage

An HBase table is in fact a **sorted map**

Keys Values

row	column	value
1	type	Friend request status
1	for user	Ryan
1	from user	Jessica
1	timestamp	146710201
2	type	Comment
2	for user	Chaz
2	from user	Daniel
2	timestamp	146711200
3	type	Comment
3	for user	Rick
3	from user	Brendan

A sorted nested map

<Row id,
ColumnFamily,
<Column,
<Timestamp,Value>>>

<Row id,

A sorted nested map

ColumnFamily,
<Column,
<Timestamp,Value>>>

When you read data from HBase, it performs a lookup for the specified row id

<Row id,

A sorted nested map

When you write data to HBase, it needs to insert the row id in the right place, so the rows are sorted

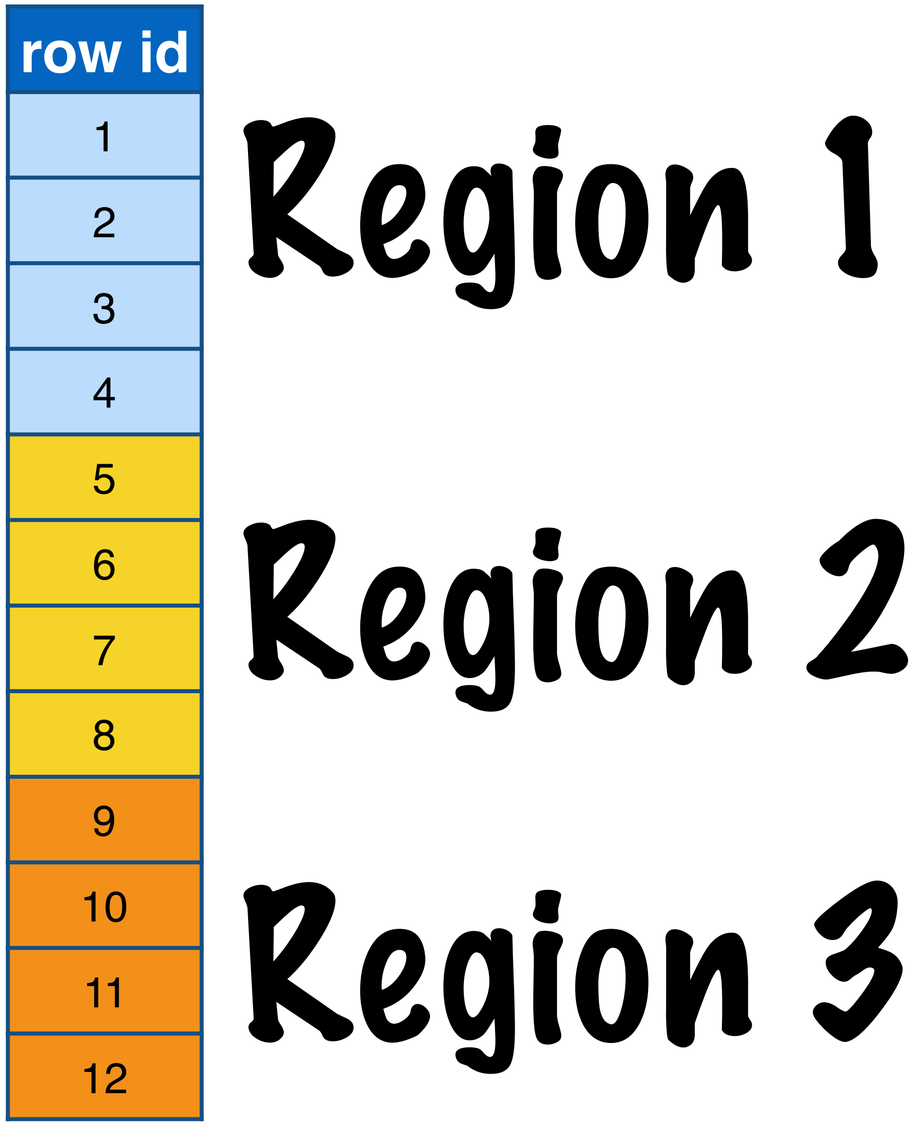
<Row id,

A sorted nested map

ColumnFamily,
<Column,
<timestamp,Value>>>

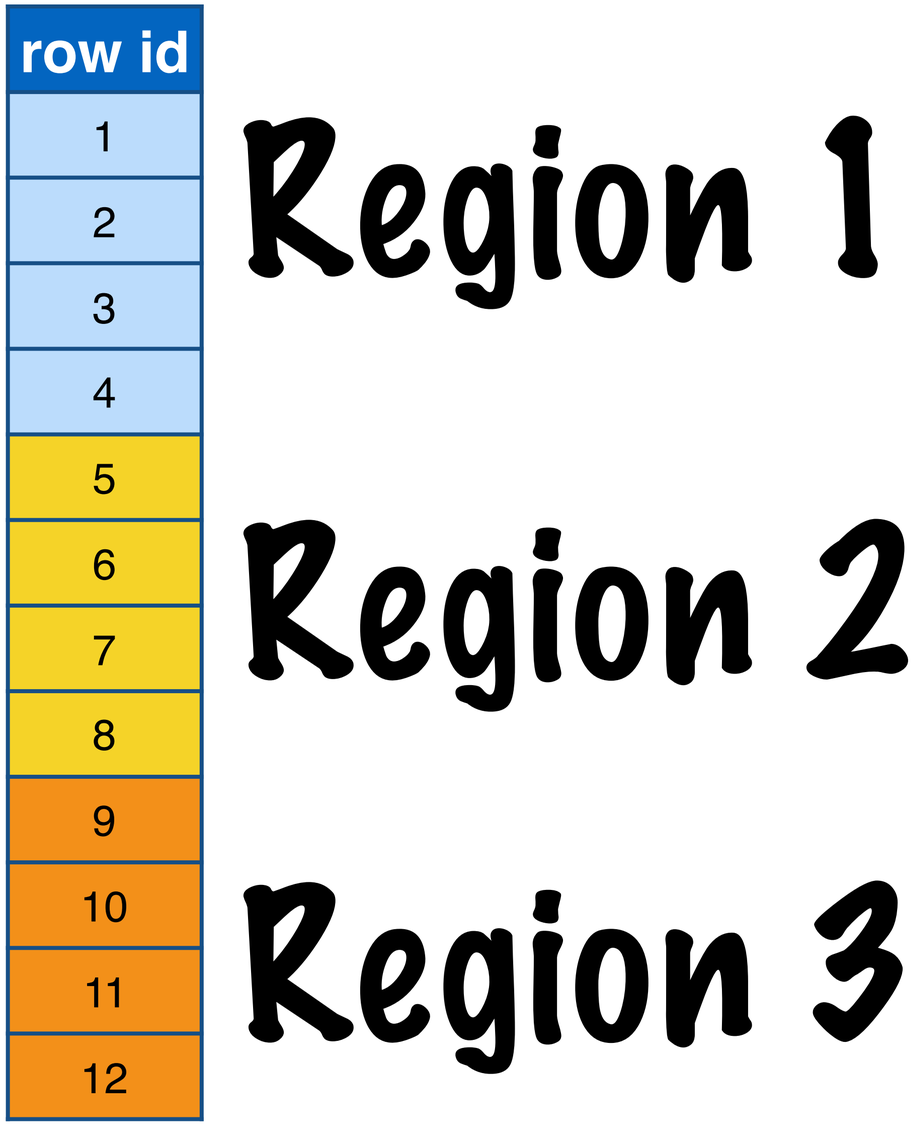
HBase does this using Region Servers

Region Servers



Row ids in a table are divided into ranges called regions

Region Servers



Each region is handled by a Region Server

Region Server 1

Region 1

Region 3

Region Server 2

Region 2

Region Servers

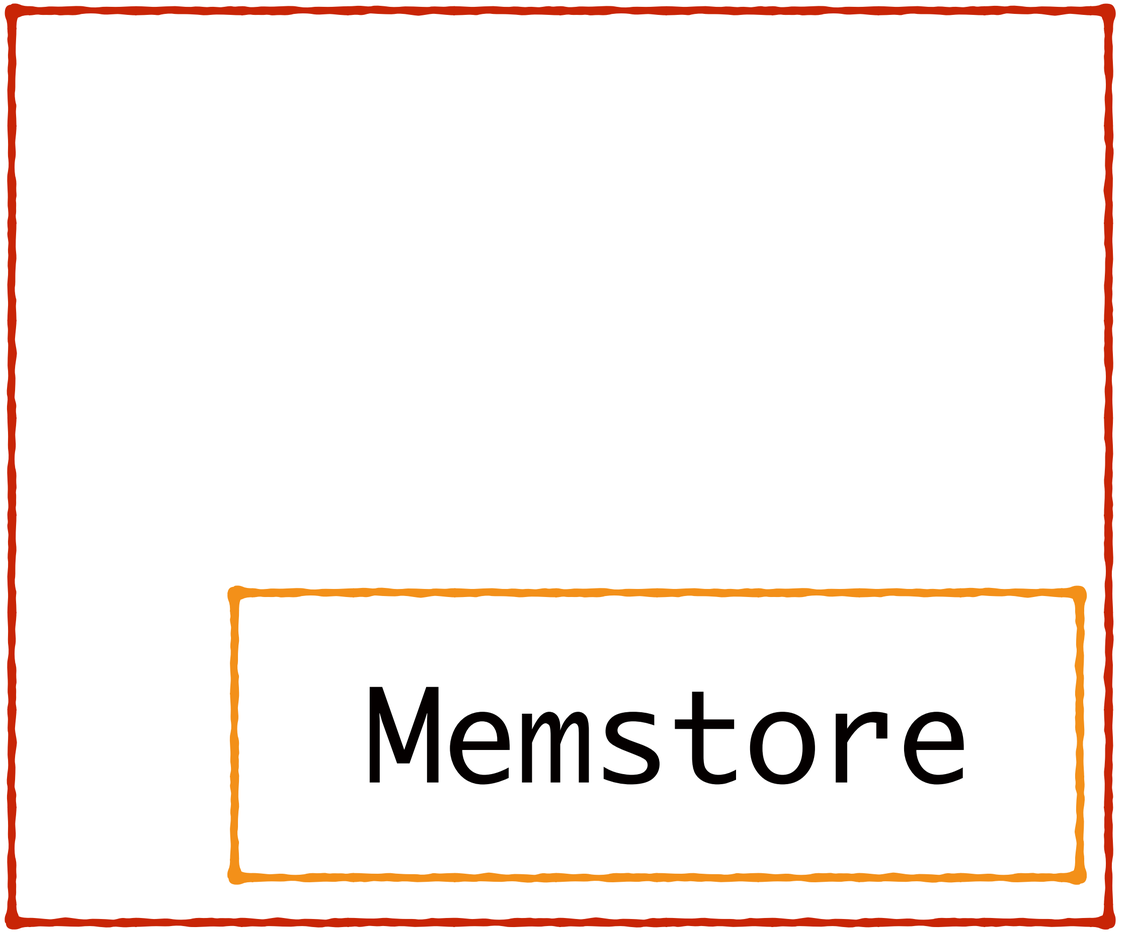
Regions serve as an index to perform fast lookup for where a row key belongs

Region Servers

A region server handles all read-write operations to Regions that are allotted to it

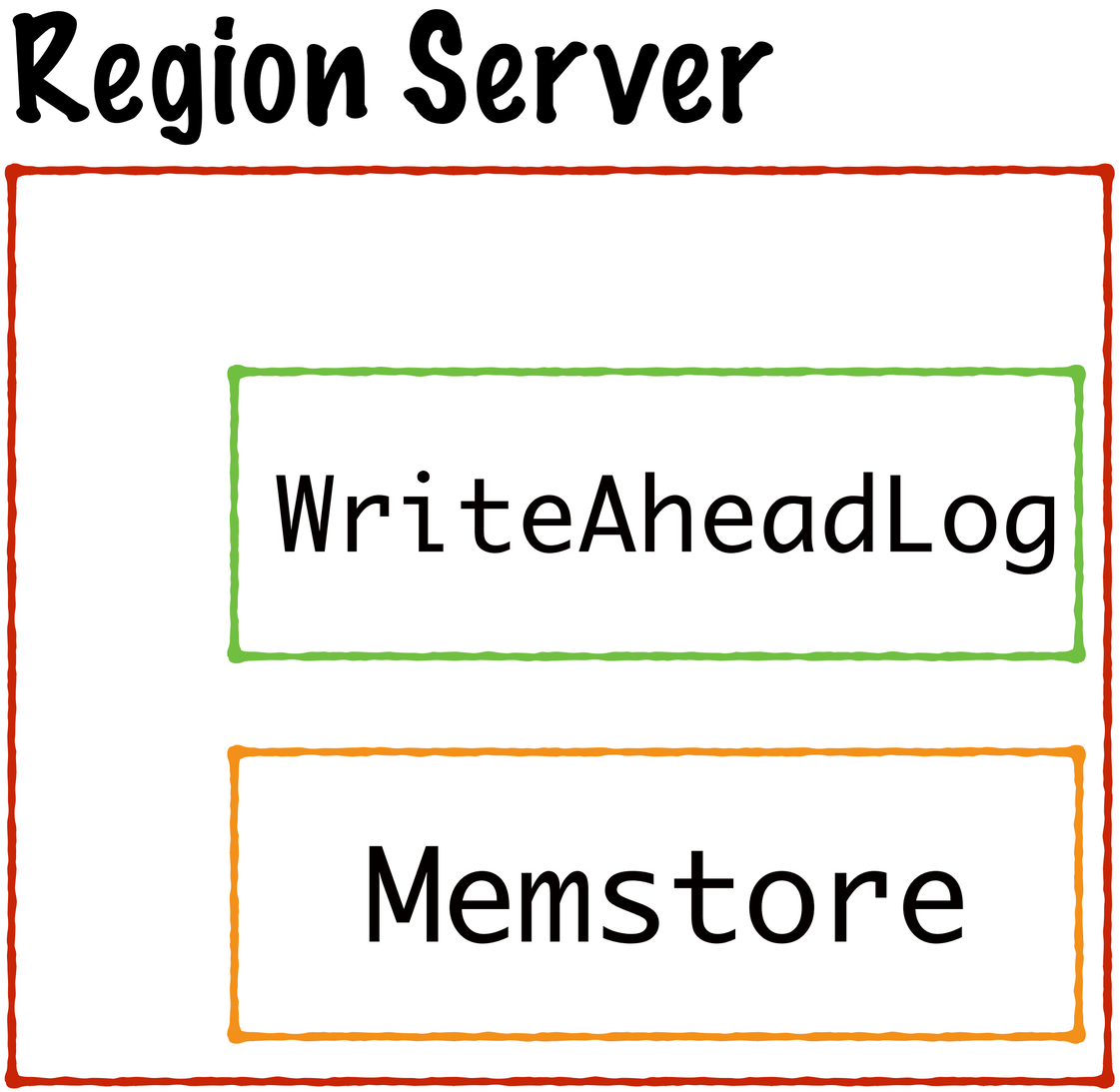
Region Servers

Region Server



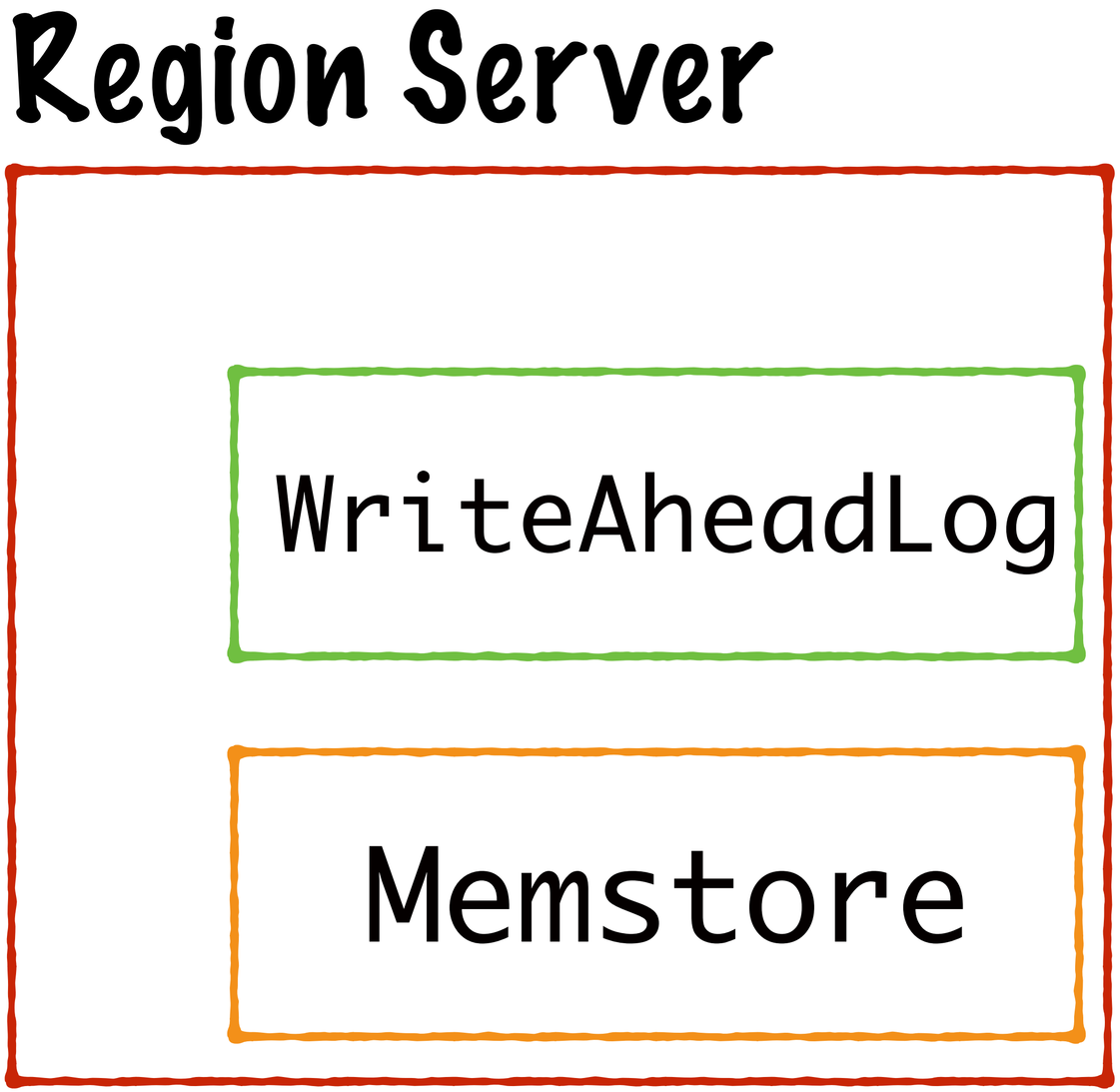
Initially all writes are stored in memory

Region Servers



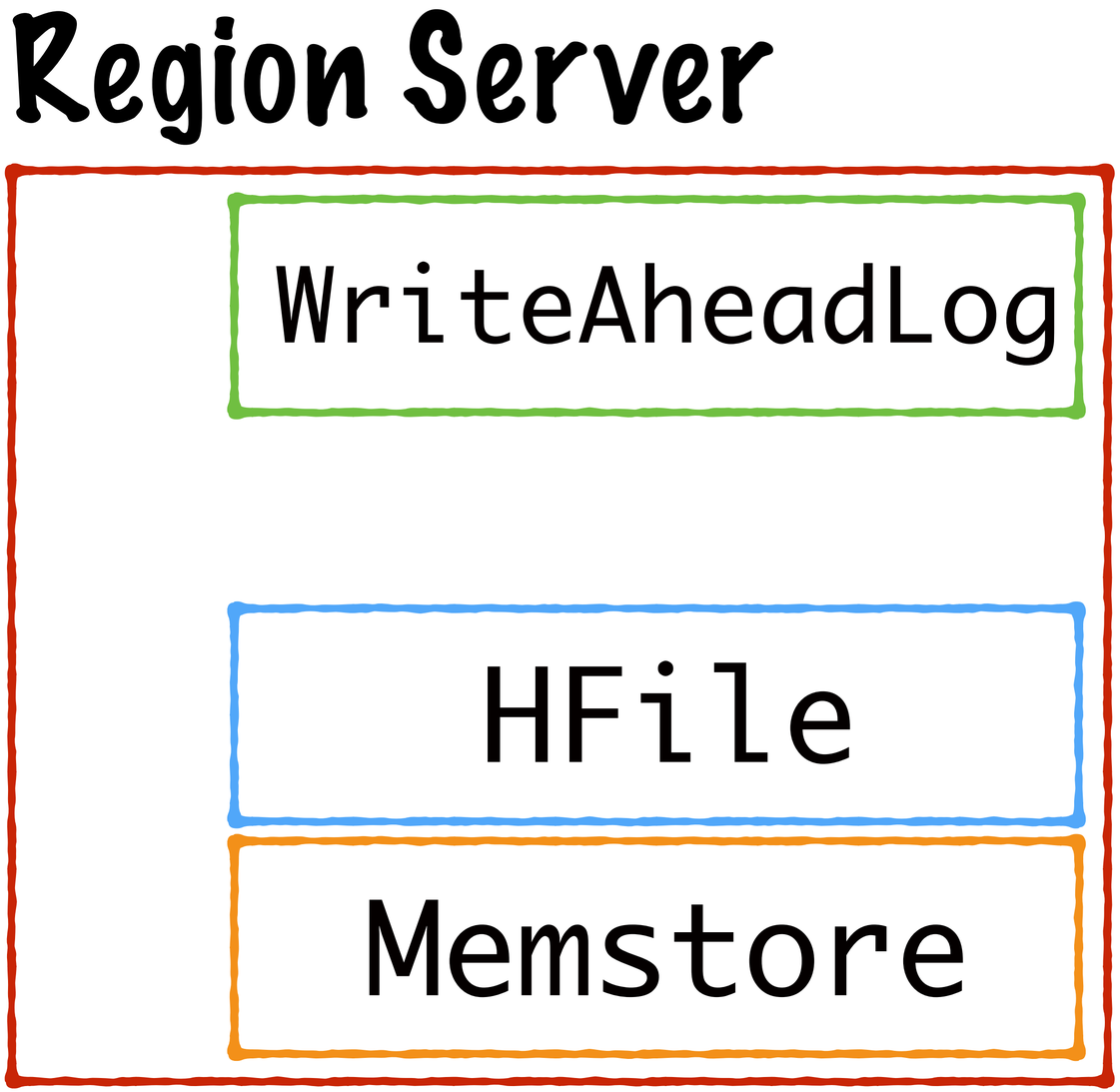
Whenever there is a new change, the data is updated in the Memstore and a change log is written to disk

Region Servers



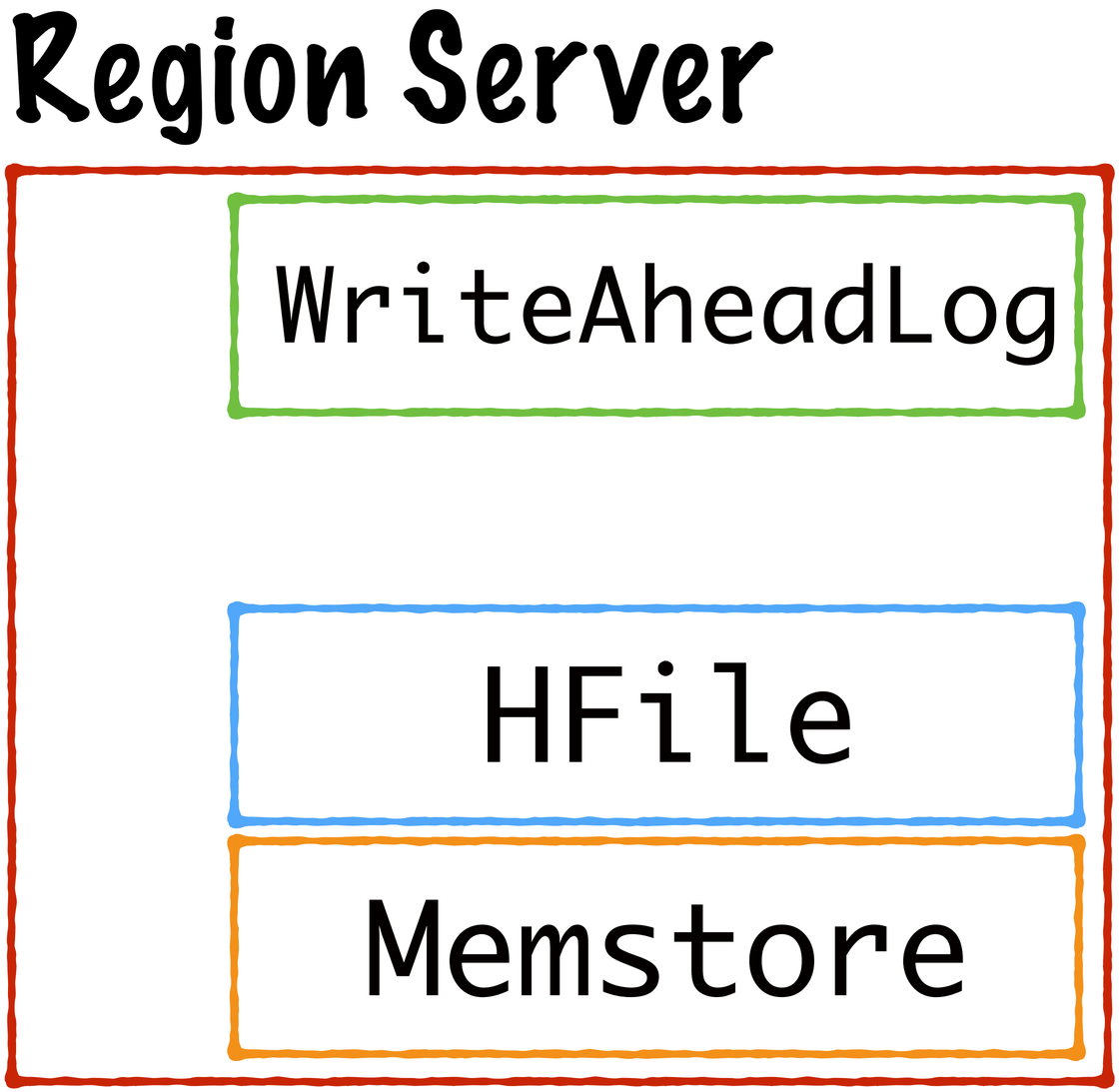
The WriteAheadLog is created for recovery in case the Region Server crashes

Region Servers



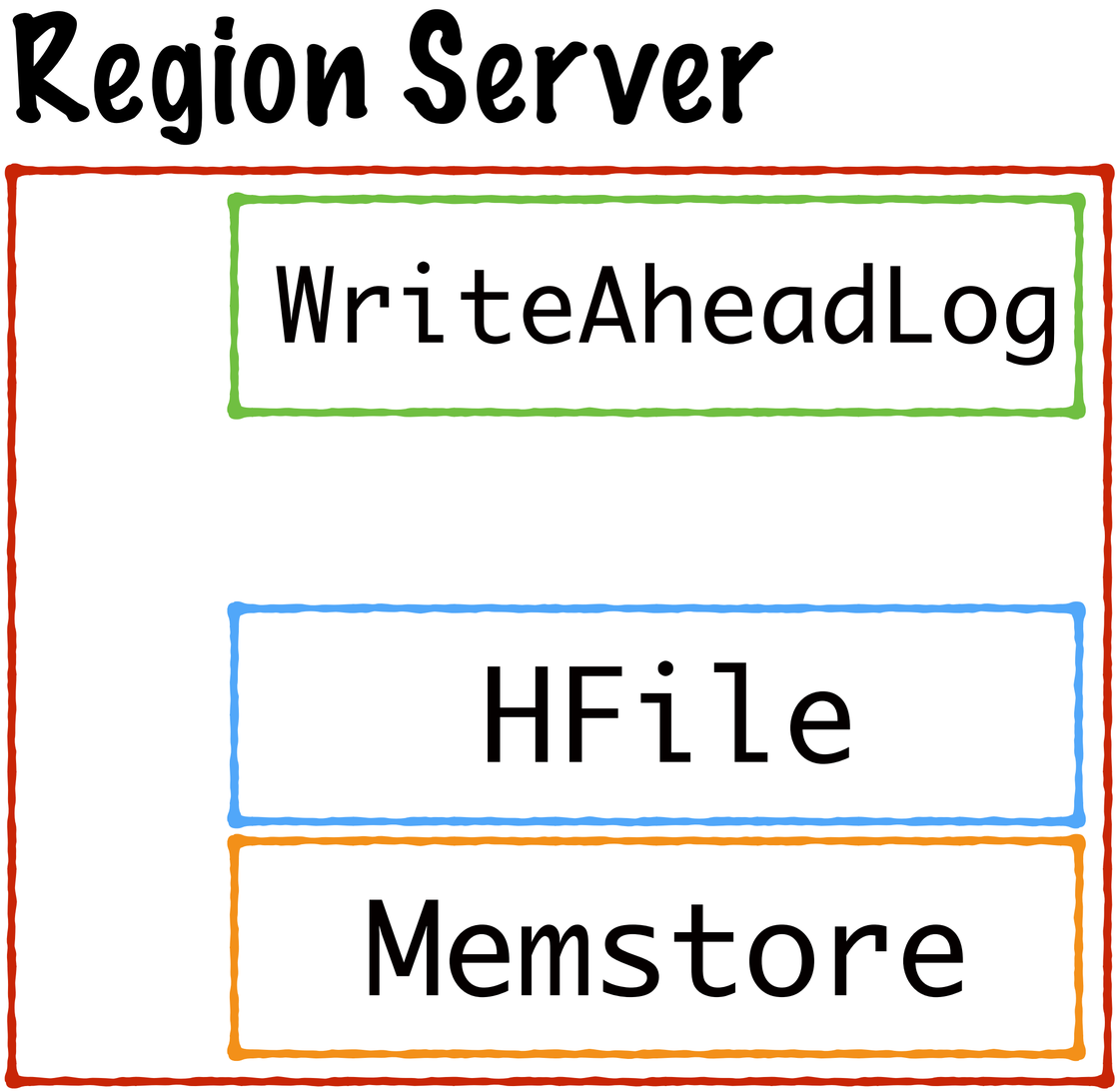
Periodically the Memstore gets full, and the data in Memstore is flushed to disk

Region Servers



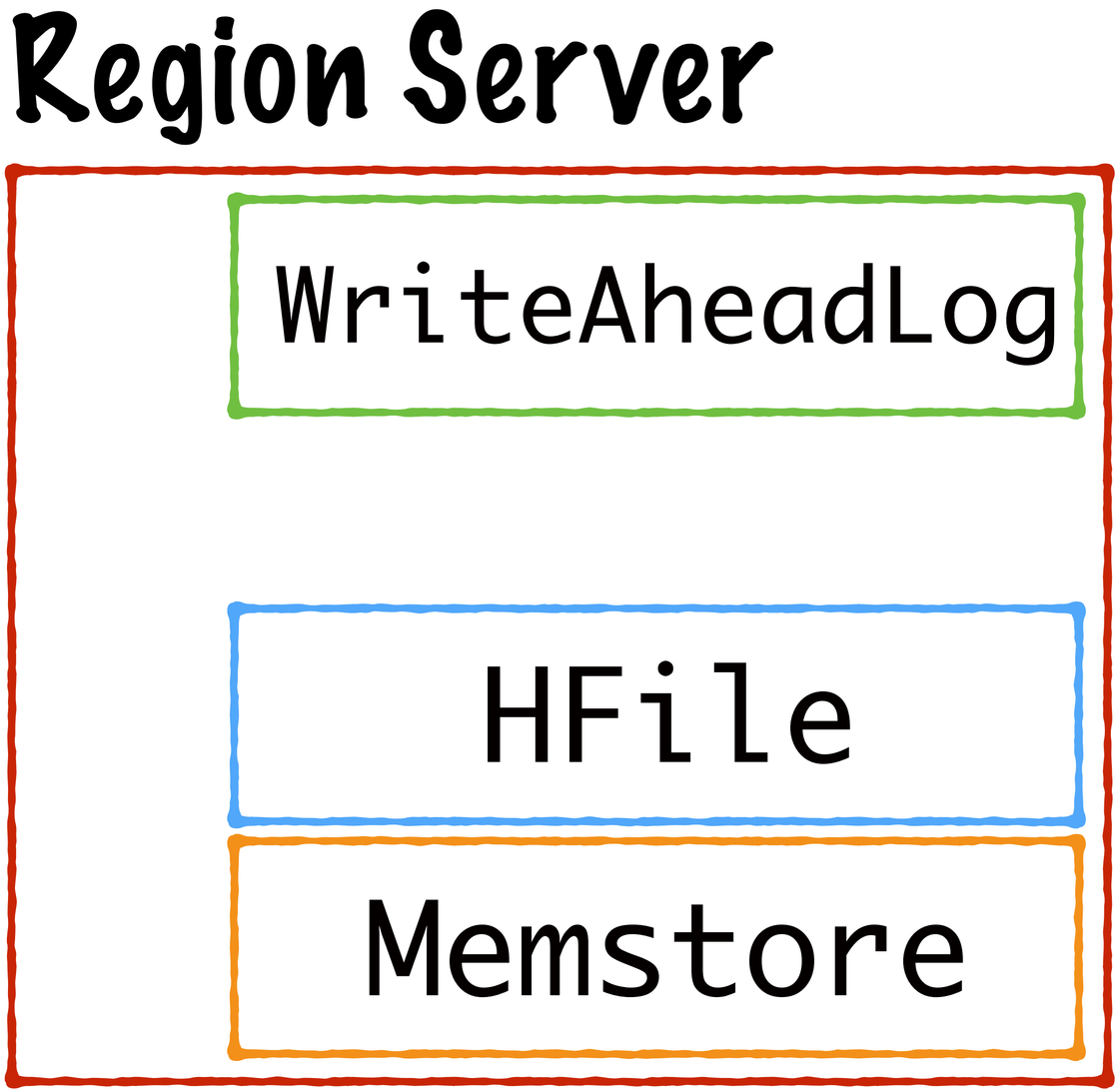
The data for a row key is either in the Memstore or in a HFile

Region Servers



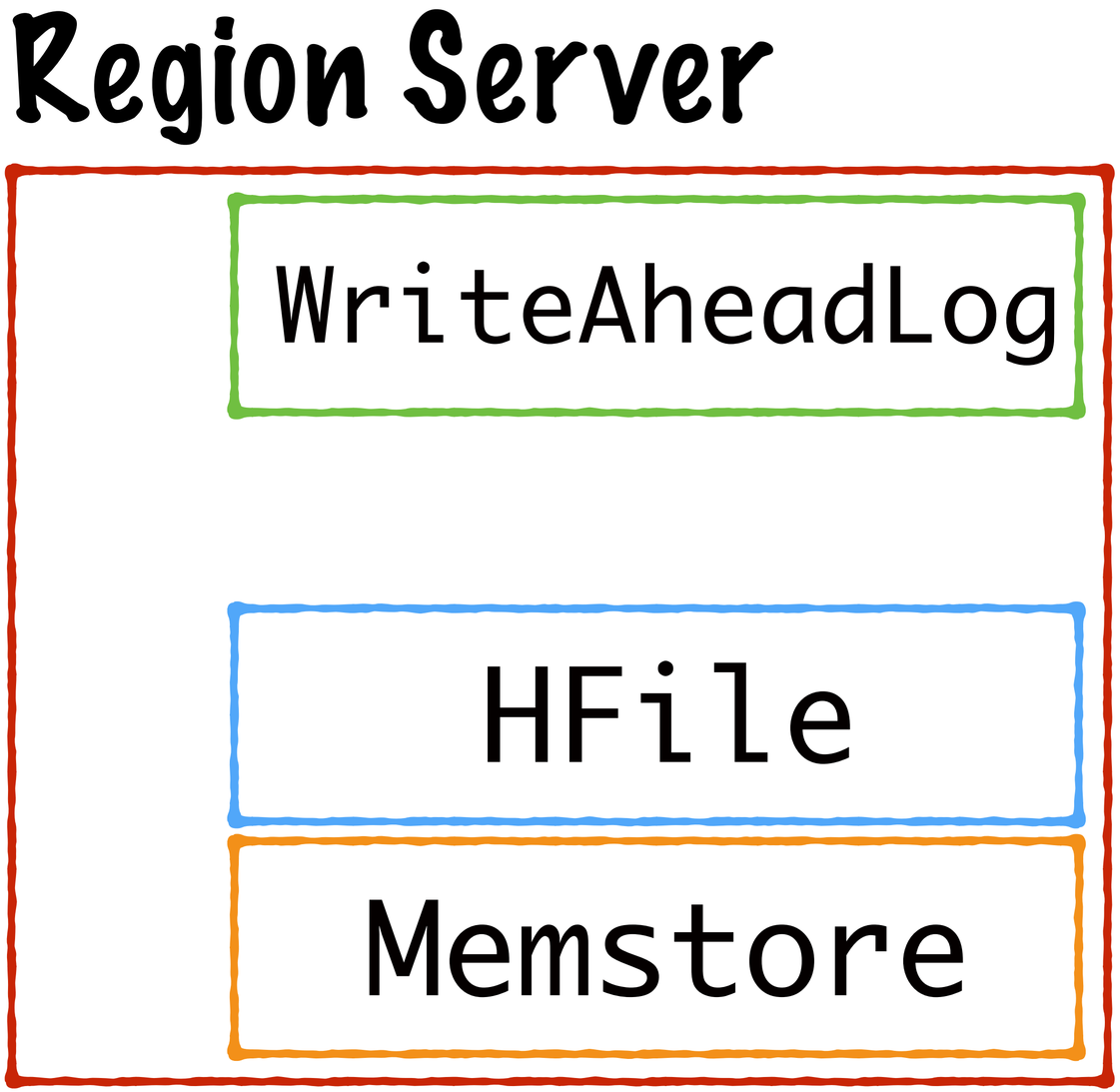
HFiles are stored in HDFS

Region Servers



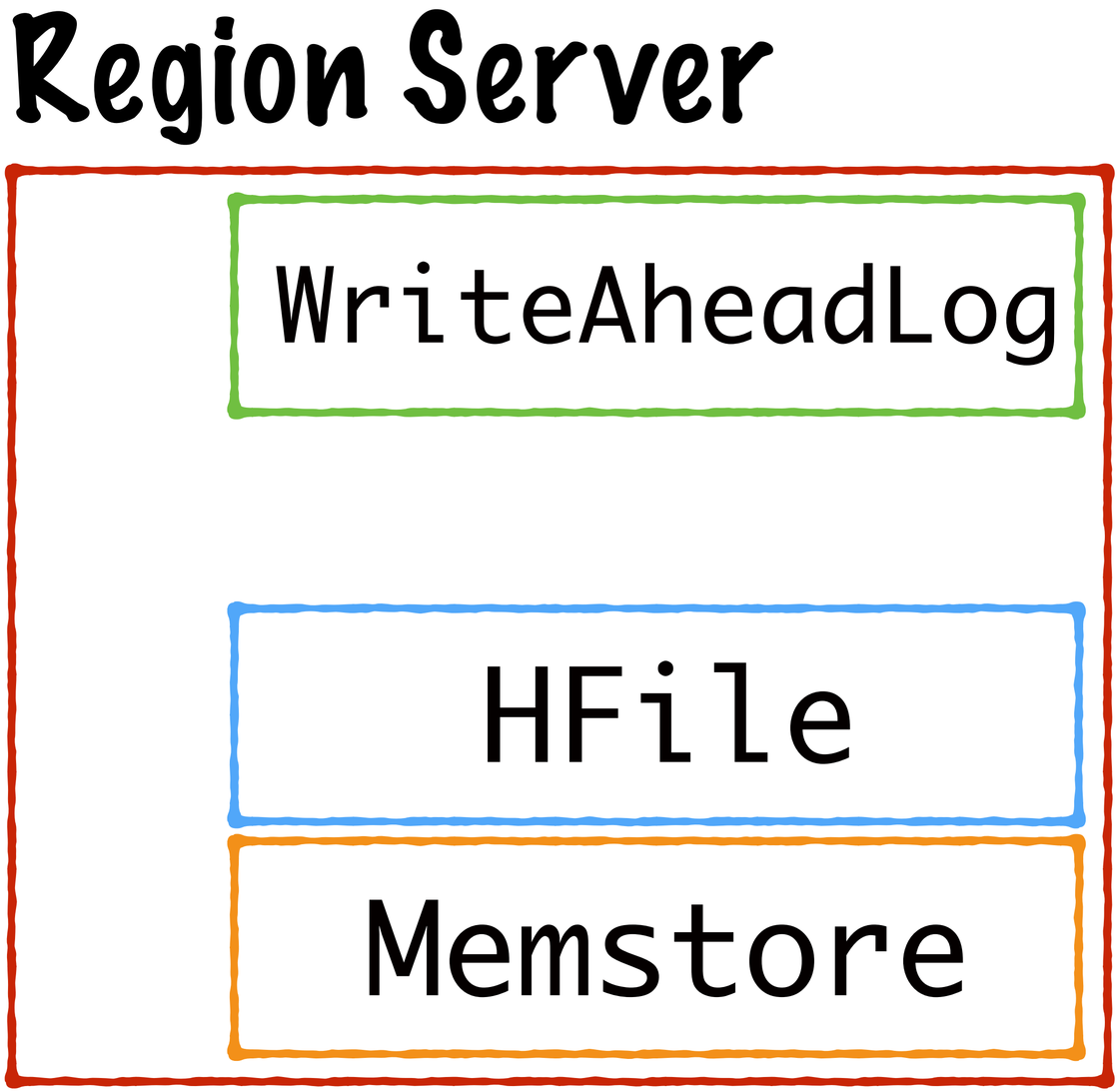
HDFS will break up the HFile into blocks and store it on different nodes

Region Servers



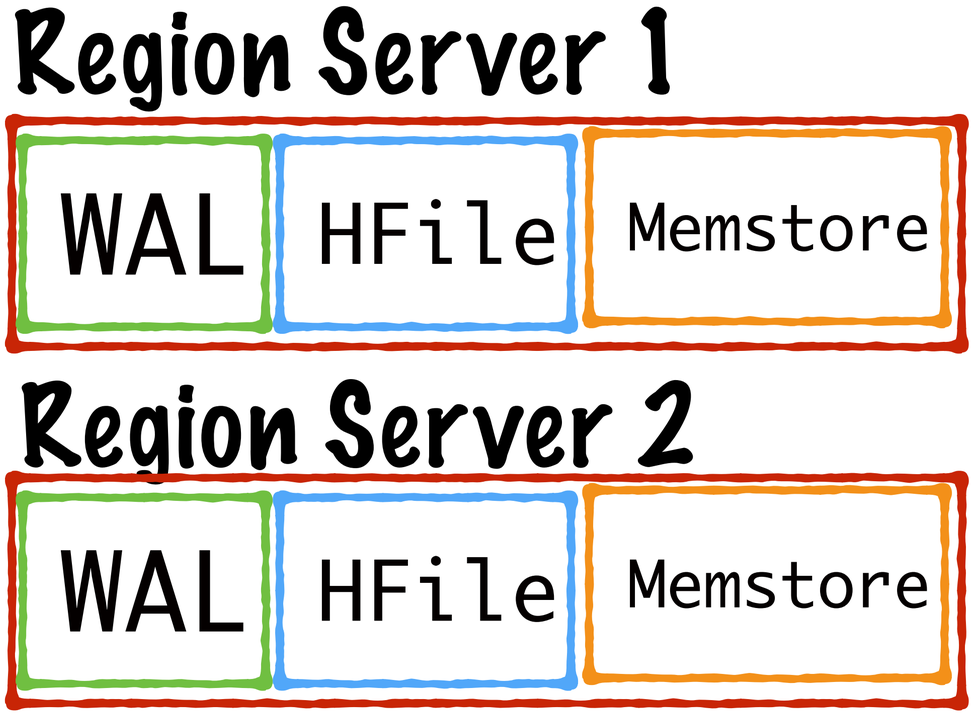
To minimize disk seeks, the region server keeps an index of row key to HFile block in memory

Region Servers



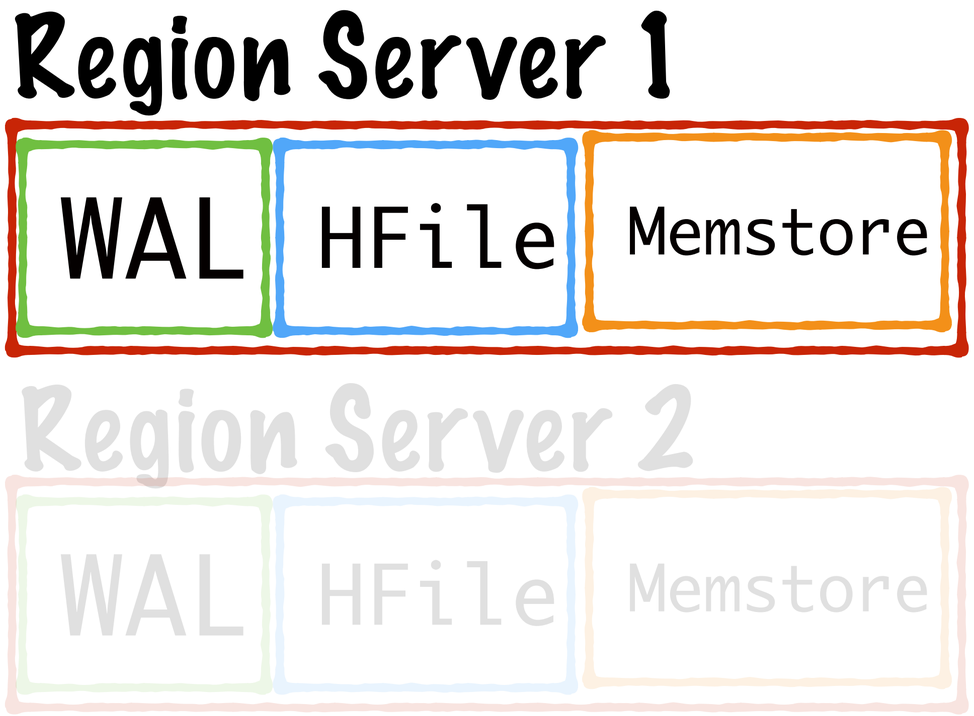
It only performs 1 disk seek for finding a row key

Region Servers



When you try to read/insert data

1. The region server containing the row key is identified



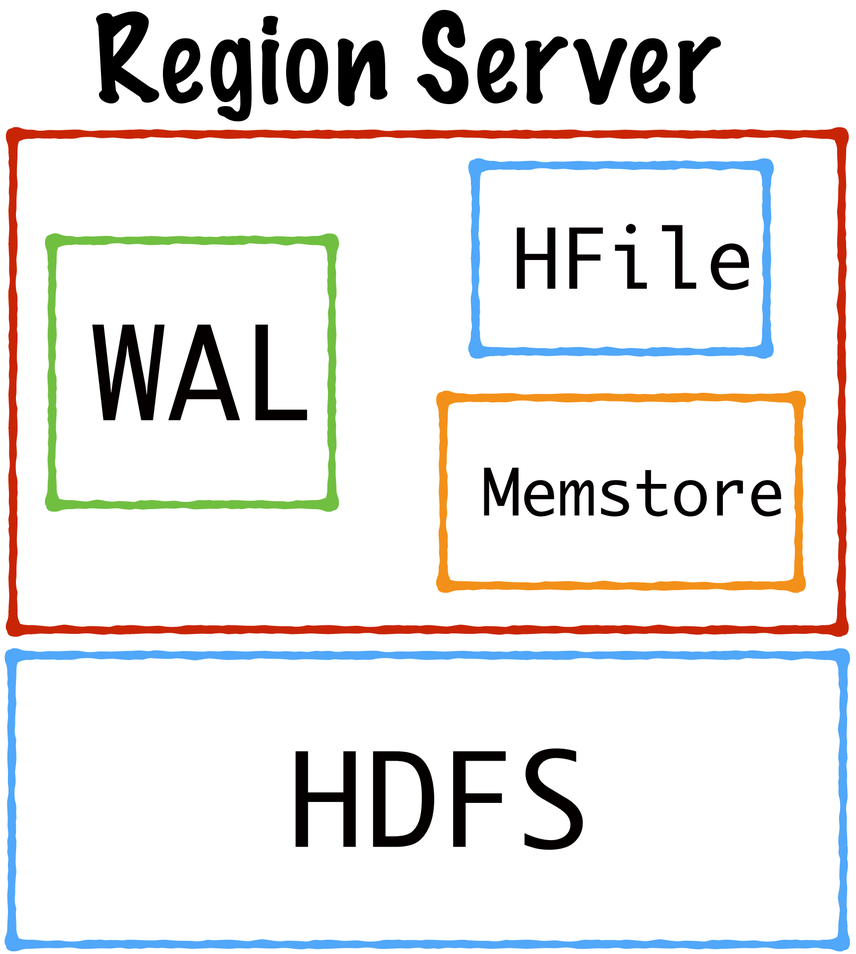
Region Servers

When you try to read/ insert data

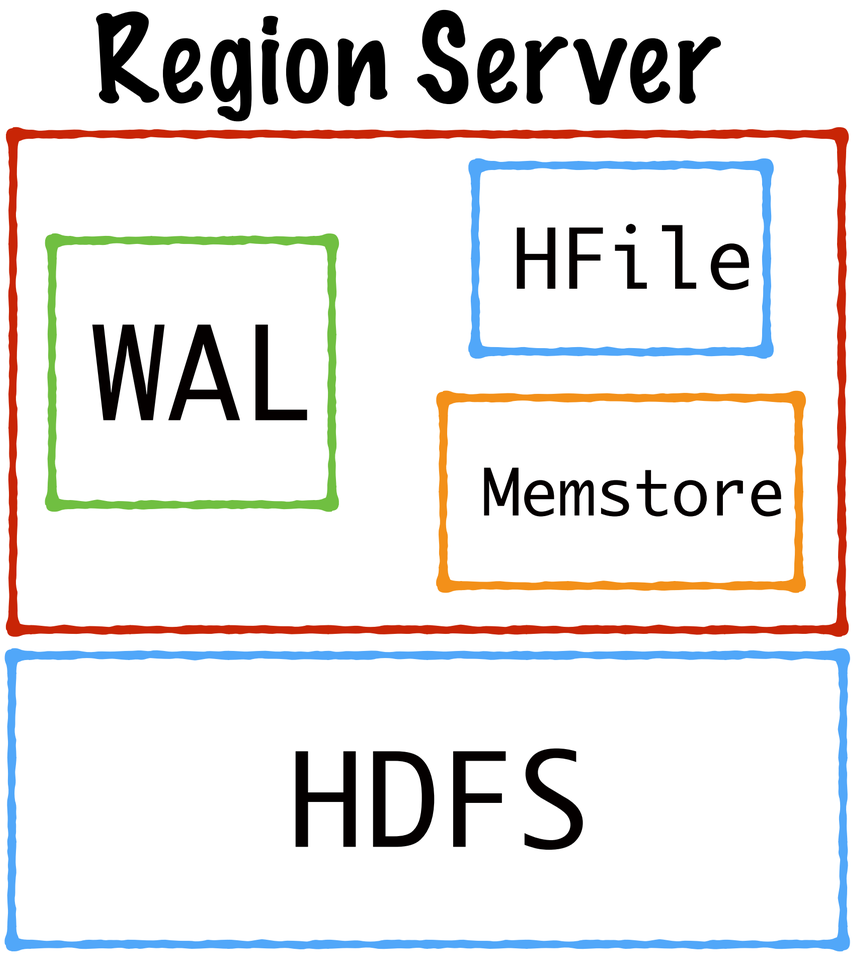
1. The region server containing the row key is identified

2. The region server will lookup the Memstore or the HFile and do the needful

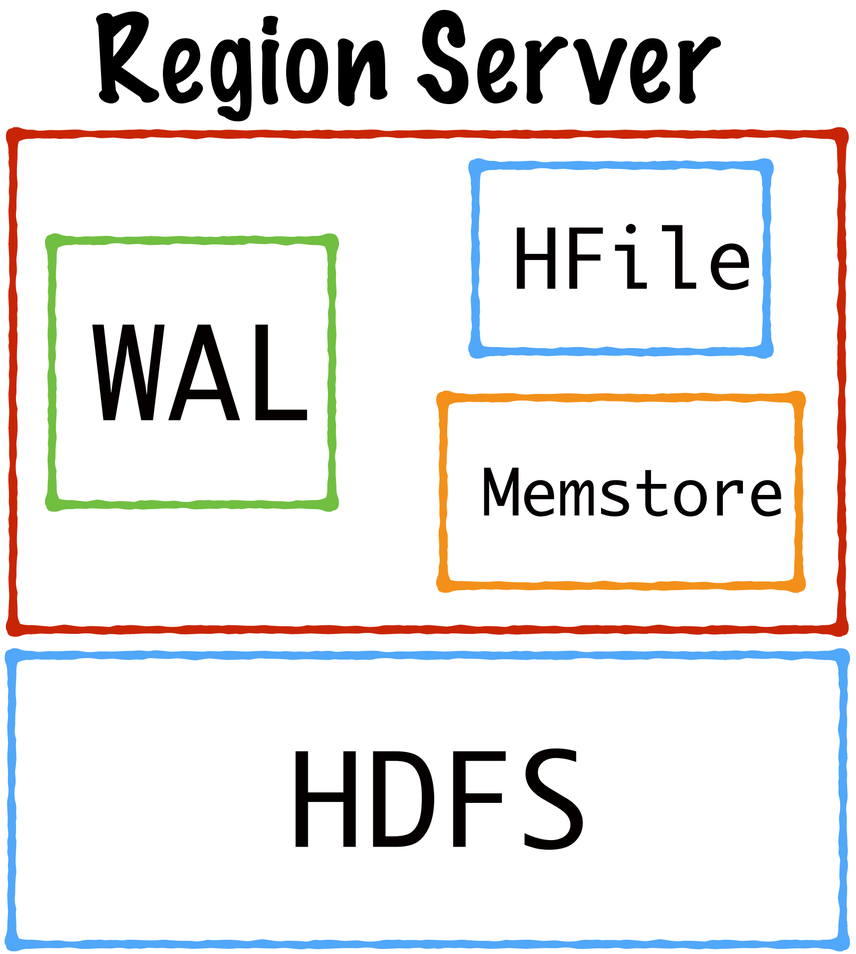
Clients interact directly with a Region server handling the relevant row keys

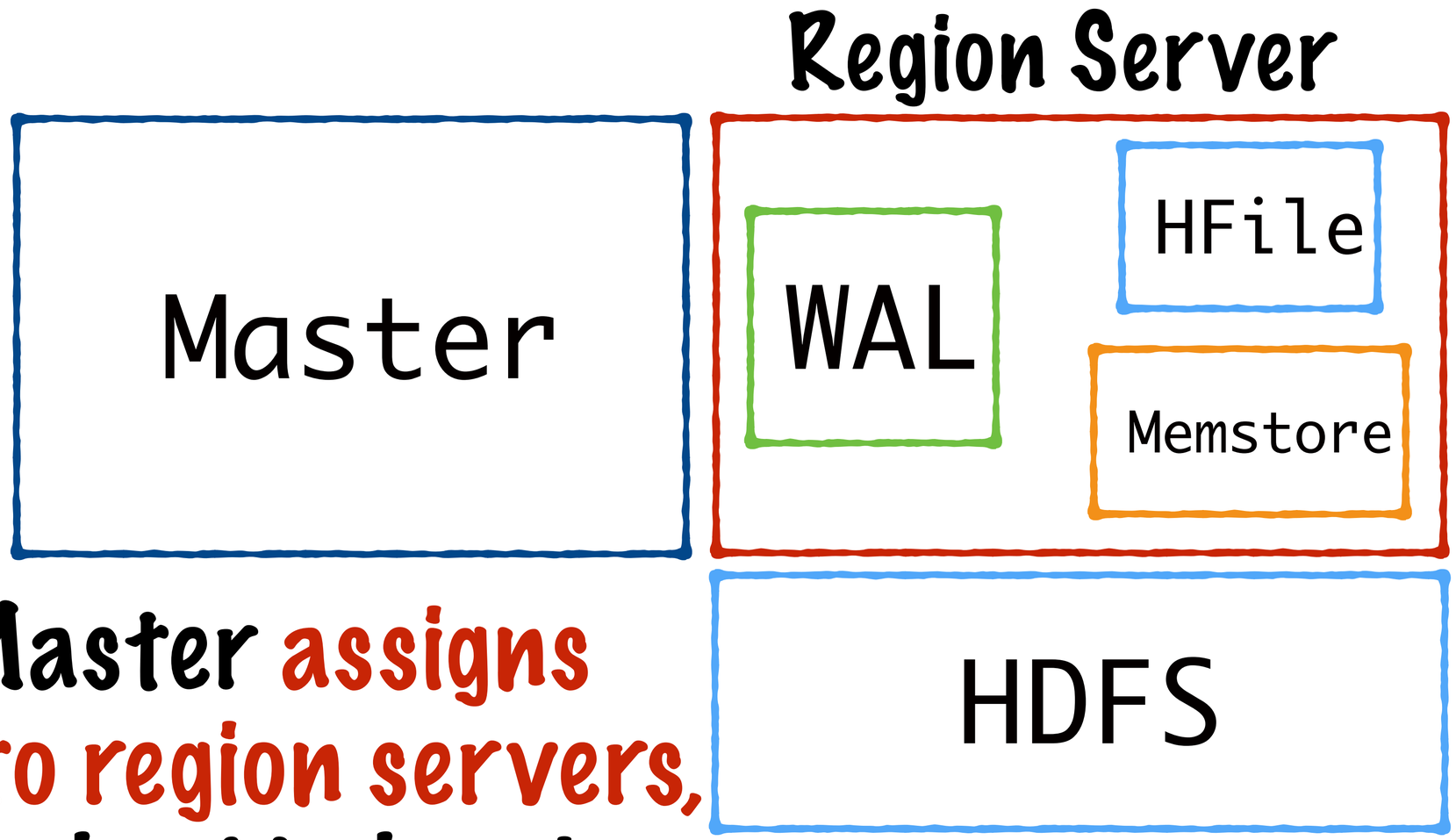


They need to know which region server their row key is being handled by

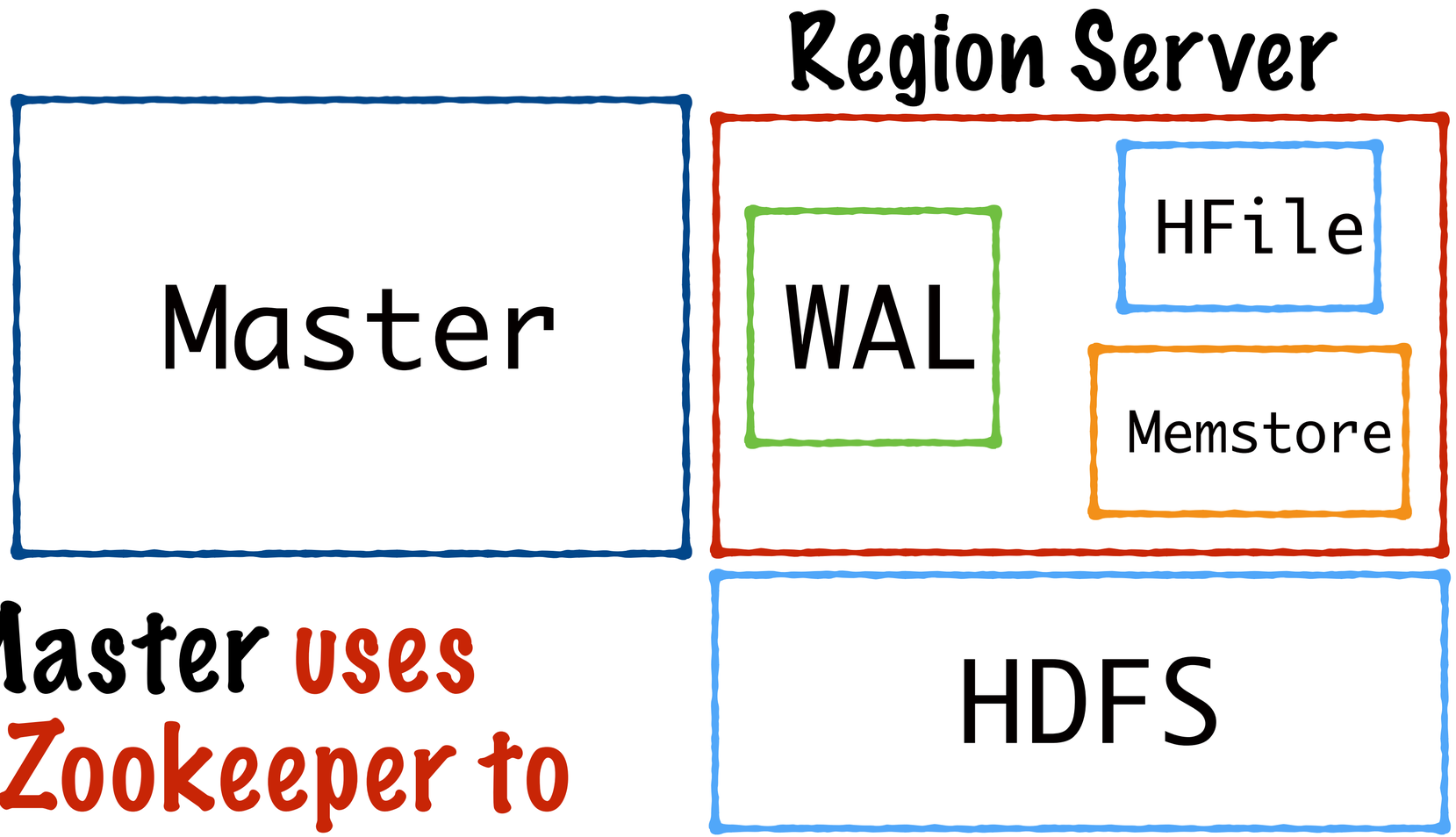


HBase uses a Master server to manage Regions and RegionServers

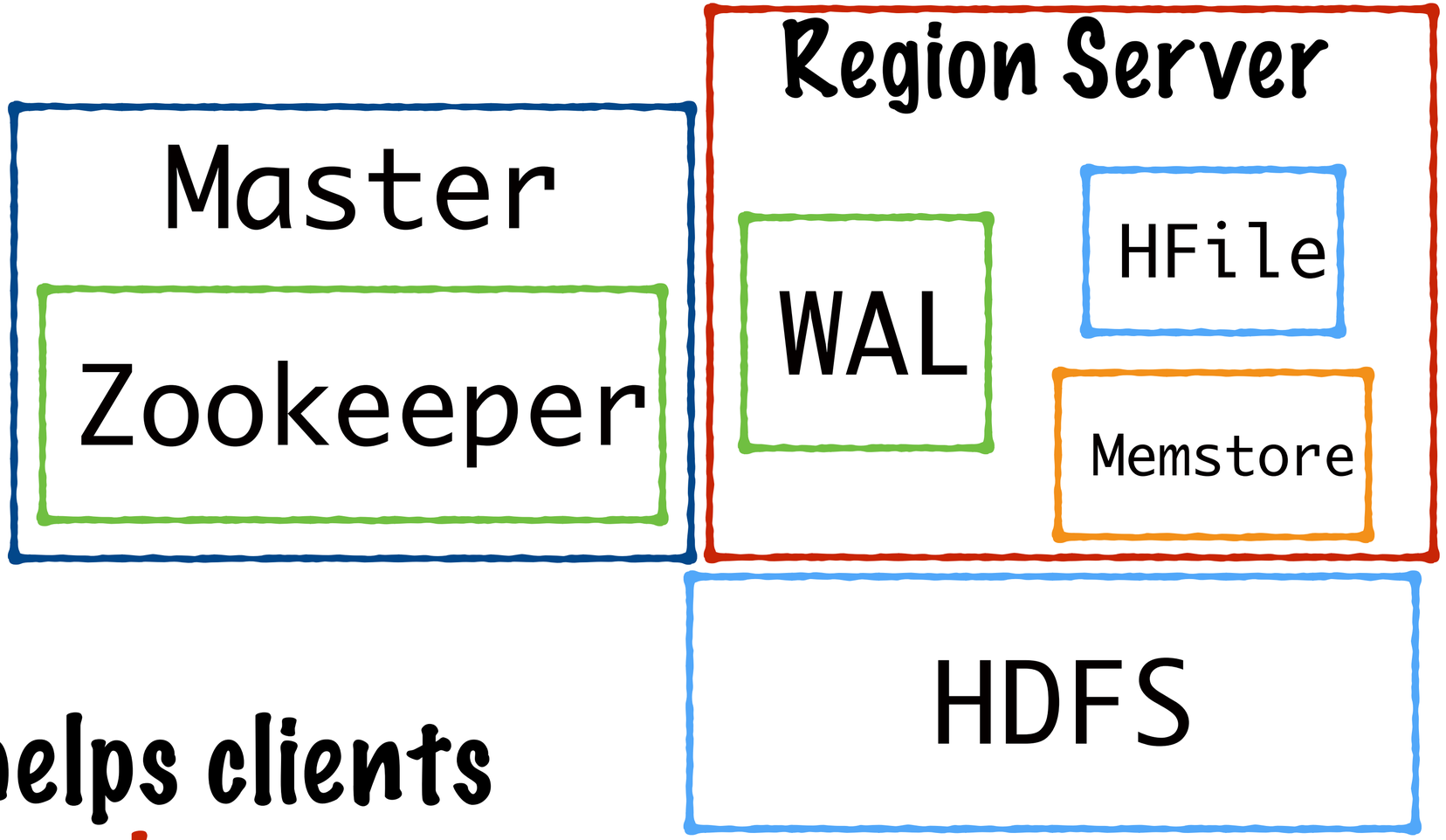




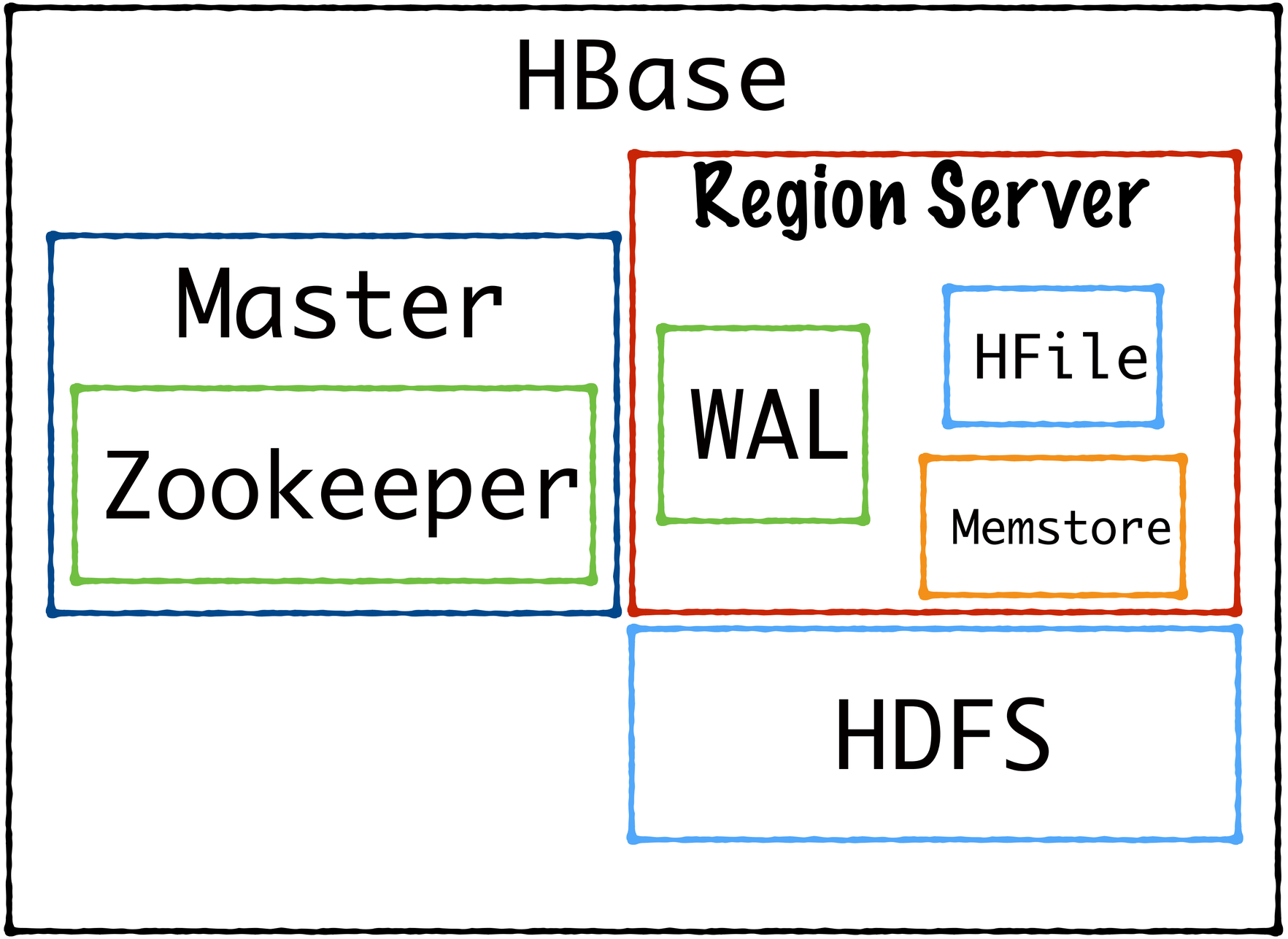
The Master assigns regions to region servers, manages load balancing etc



The Master uses Apache Zookeeper to help assign regions to region servers



Zookeeper helps clients lookup the relevant region server for a specific row id


HBase
Region Server
Master
Zookeeper
WAL
HFile
Memstore
HDFS