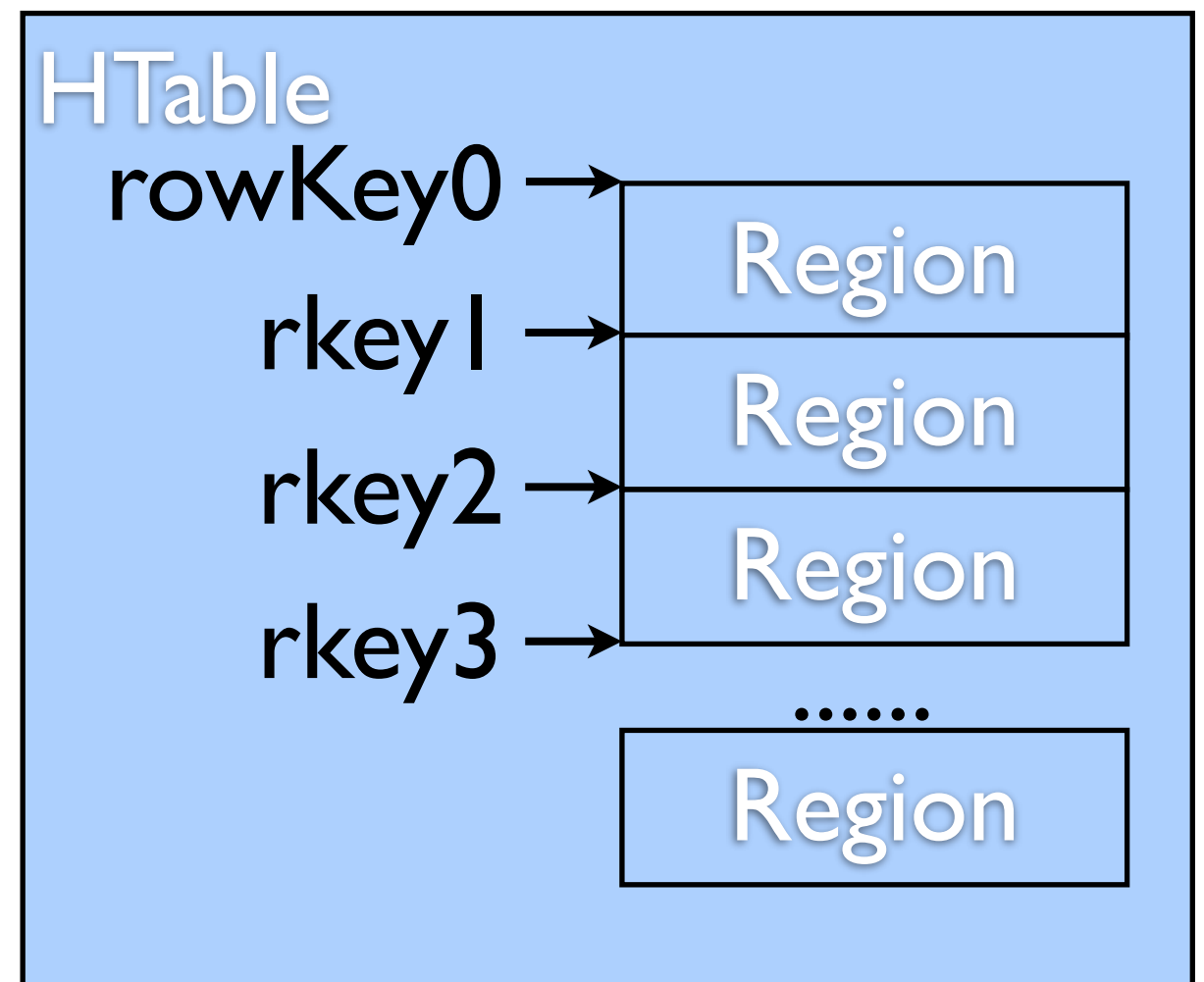


Project Mitosis

Second dimension for HBase Table: PartitionKey

Current Region Layout

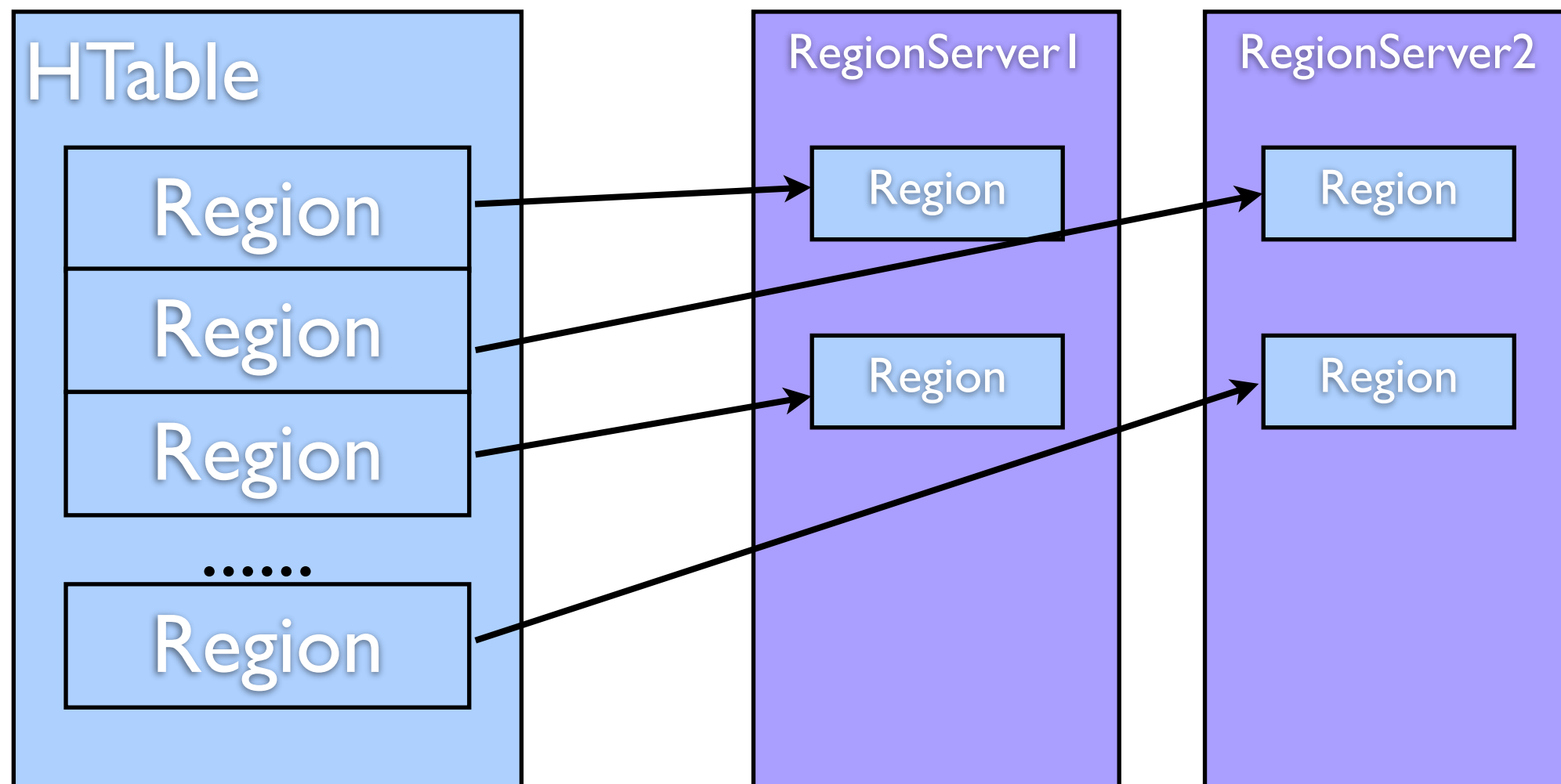
- HTable
 - $=\{\text{Regions}\}$
- Region
 - $=[\text{rkey0}, \text{rkey1})$



Current Region Layout

- RegionServer

● = *



Current Region Layout

- Benefits
 - do not worry about data distribution
 - ease of system admin
- Drawbacks
 - have no control over data distribution
 - hard to reduce query cost (join)

Current Region Layout

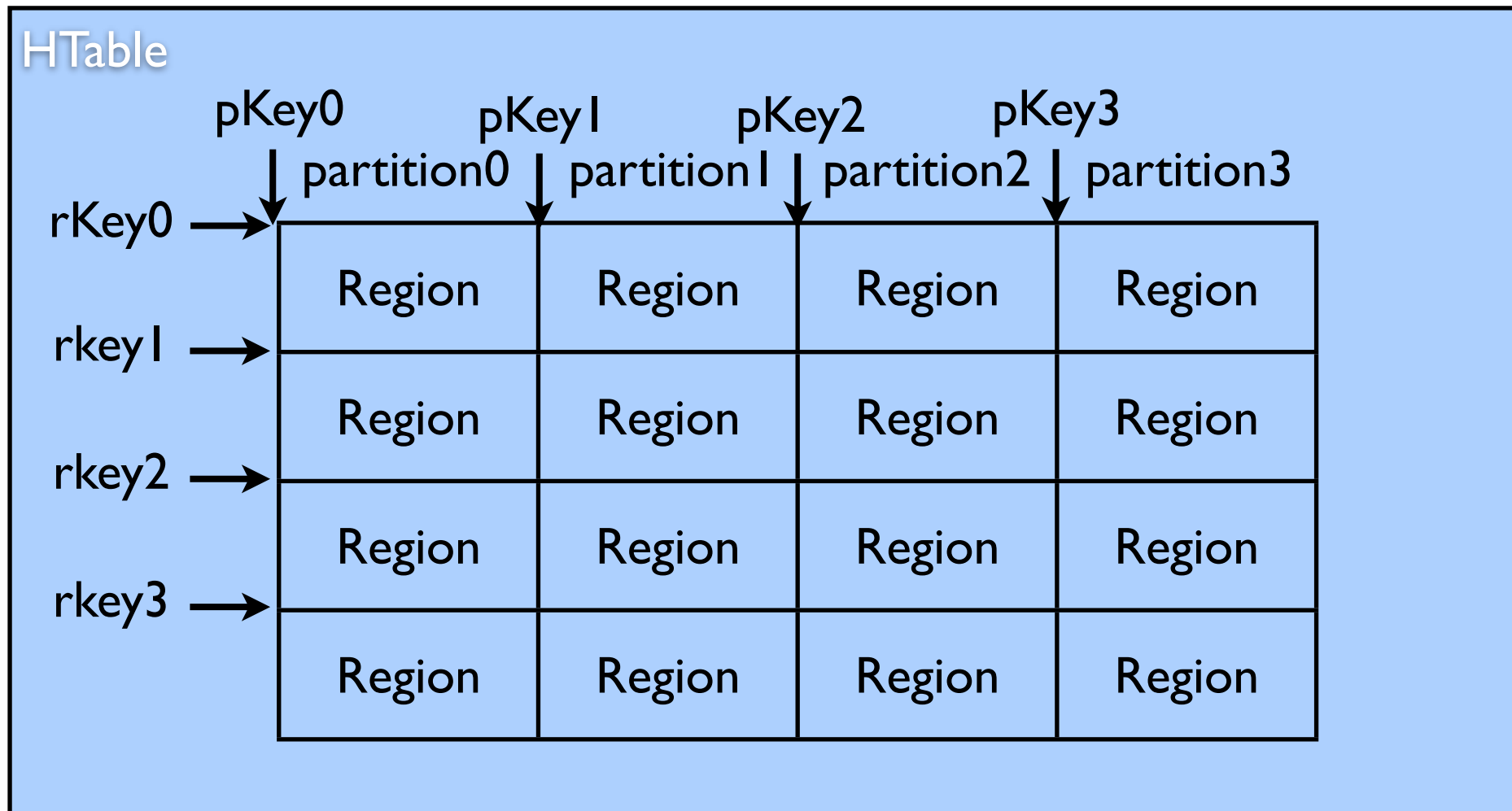
- Drawbacks because
 - rowKey is used for
 - BOTH the key of the most performant query
 - AND the key of definition of Region, thus the definition of data partitioning

Redefinition of Region

- Introduce: **PartitionKey**
 - $HTable = \{Regions\}$
 - $Region = ([rKey0, rKey1), [pKey0, pKey1))$
 - Partition: $[pKey0, pKey1)$
 - $RegionServer = \{Partitions\}$

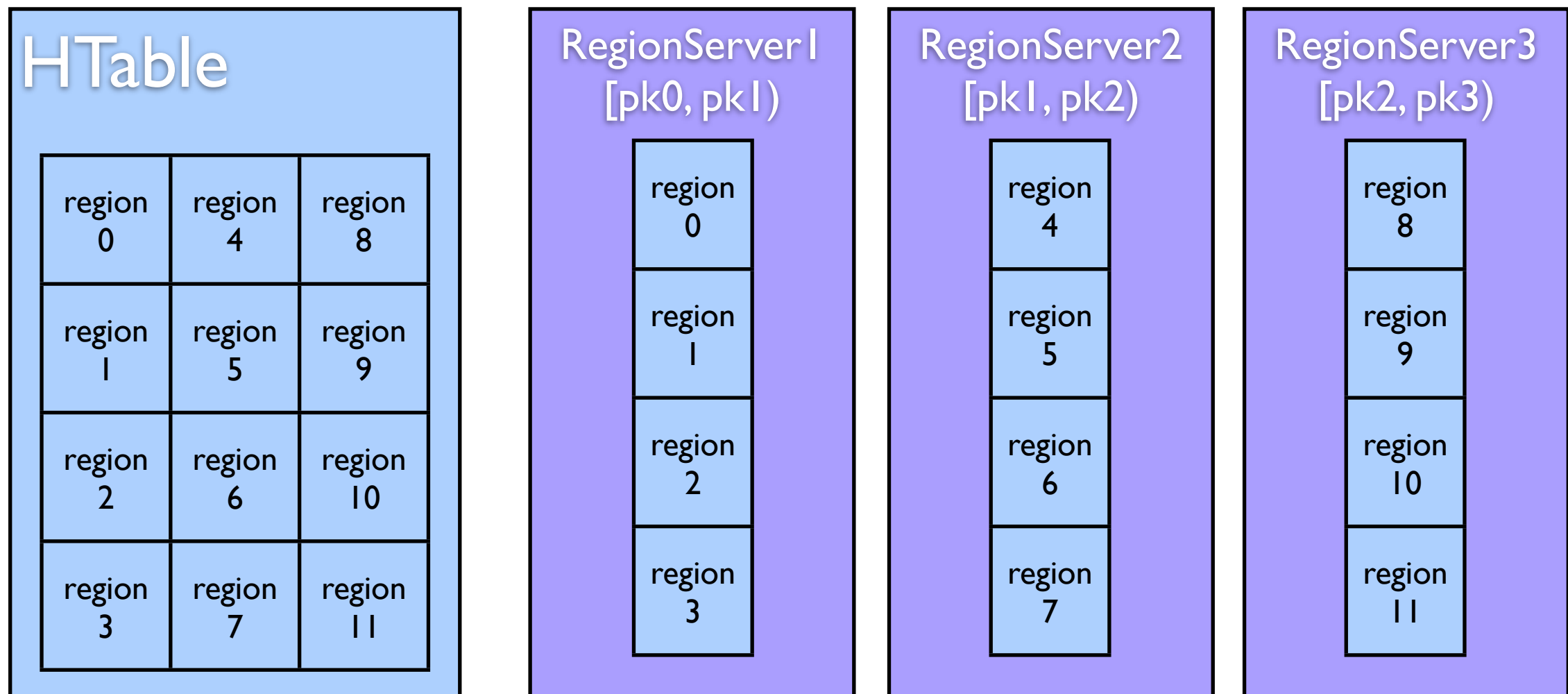
The New Region

- 2 dimensional data space of HTable



The New Region

- RegionServer example
- 3 RegionServers with 1 partition each



The New Region

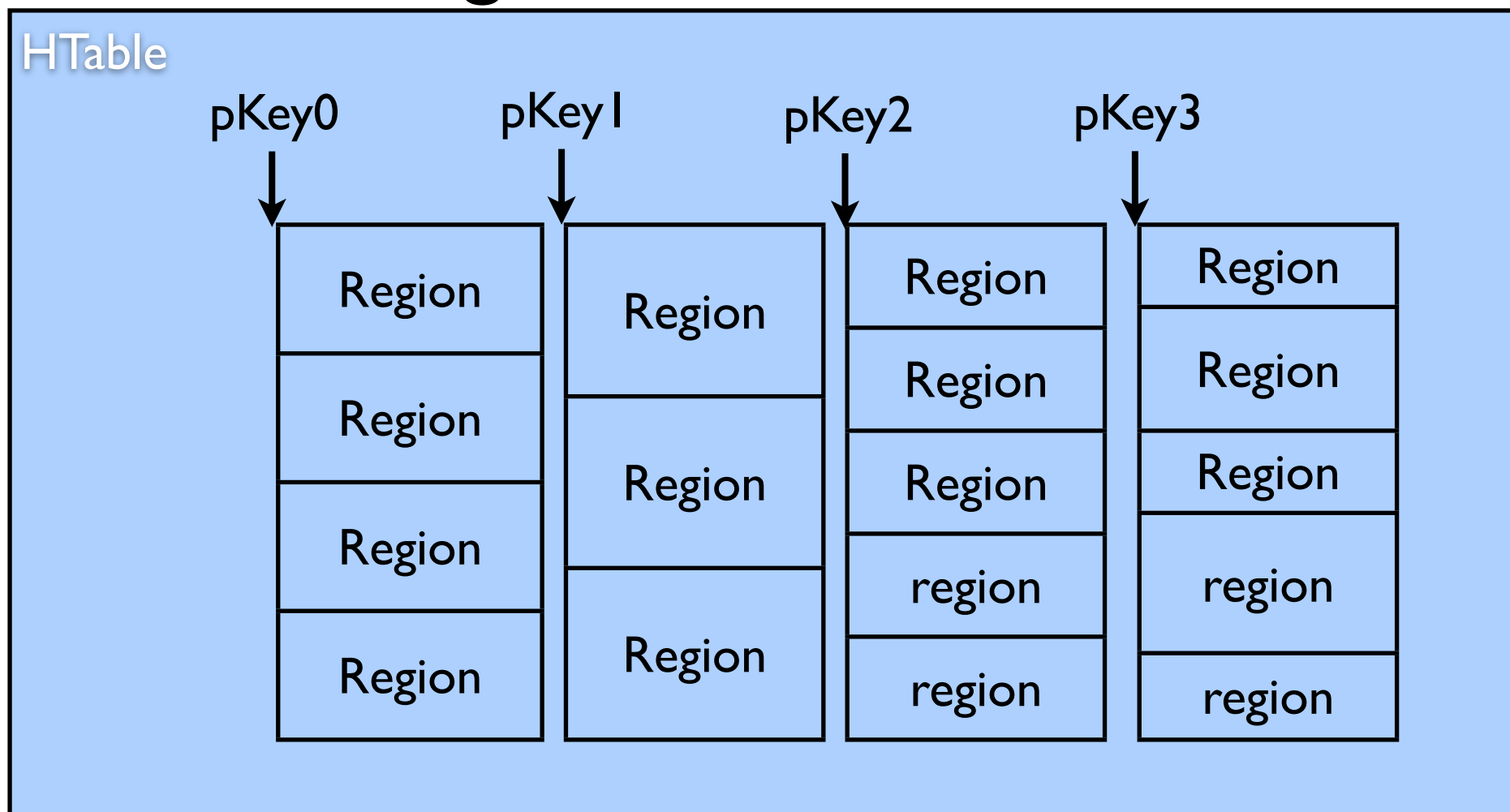
- All regions within a partition belong to the same RegionServer.
- partitions of a RegionServer do not overlay with partitions of another RS.

The New Region: P.S.

- In fact, we care more about data distribution across RegionServers.
- we don't really care about data distribution within a RegionServer.
- So...

The New Region: P.S.

- rowKey boundaries of different partitions do not align.



TODO list: changes

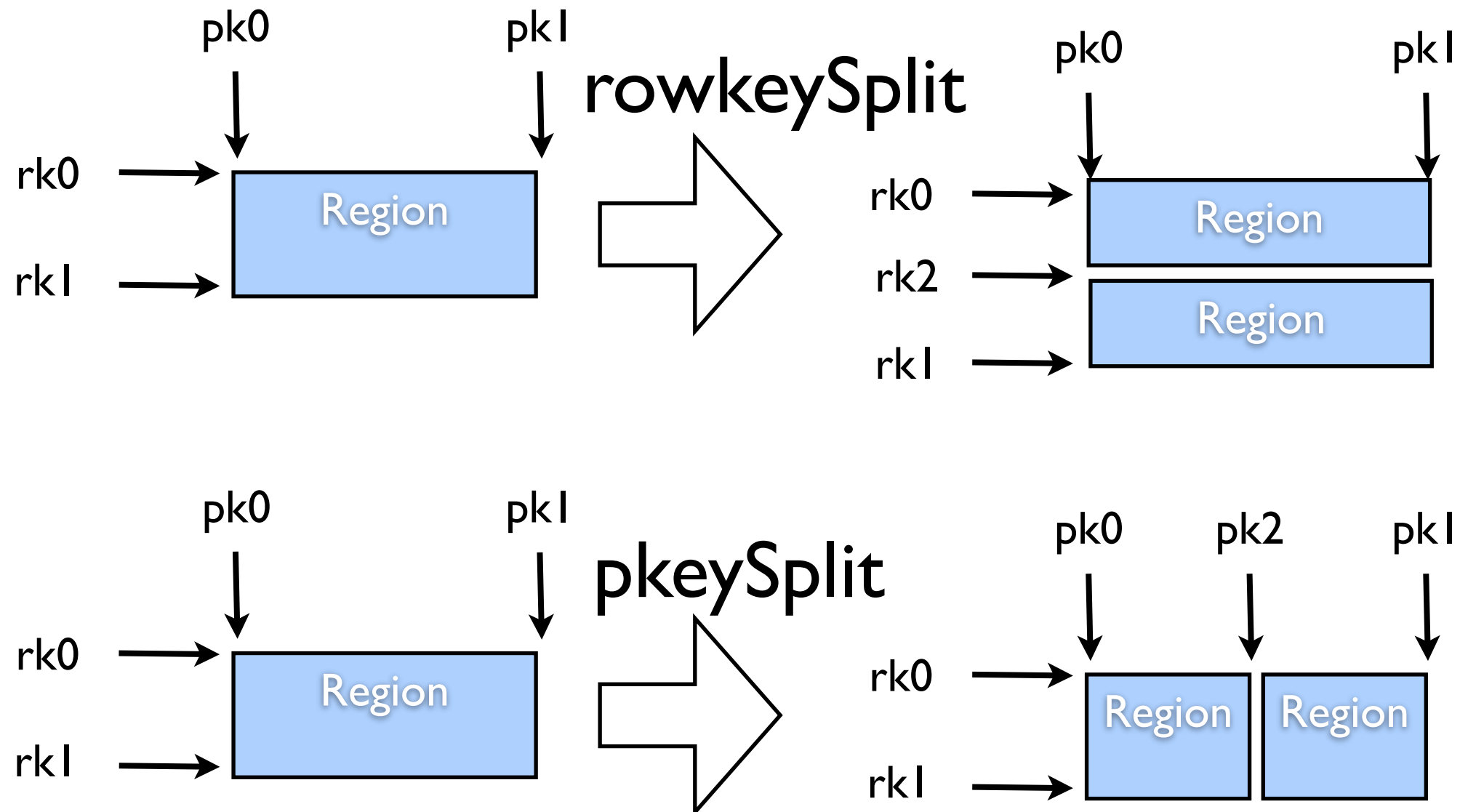
- Region
 - additional meta about pKey
- RegionServer
 - additional meta about pKey
- HMaster
 - pKey-aware of Region-RS assignment
- new procedure: **PartitionSplit**
- changes in read/write op of HBase

Partition Split

- Why a region splits? mainly 2 reasons:
 - (a) we need smaller region for faster op.
 - (b) we need to distribute data to more node .
- with PartitionKey, there are 2 types of split:
 - a region split along rowKey (for (a))
 - a region split along pKey (for (b))

Partition Split

- rowKey Split vs pKey Split

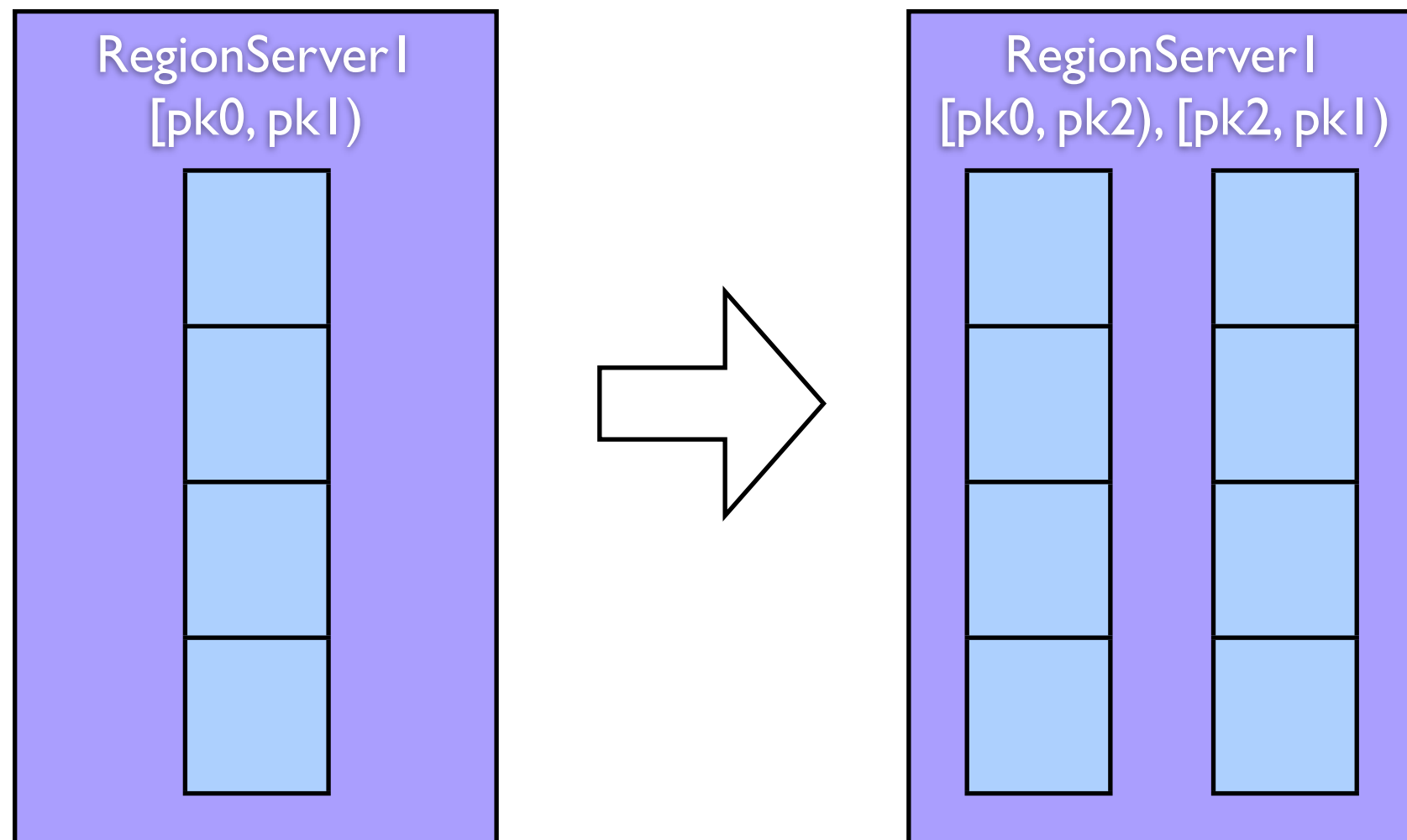


Partition Split

- Definition: a partition splits into 2
 - $[pK0, pK1) \rightarrow [pk0, pk2), [pk2, pk1)$
- every Region in the partition split into 2 along pKey

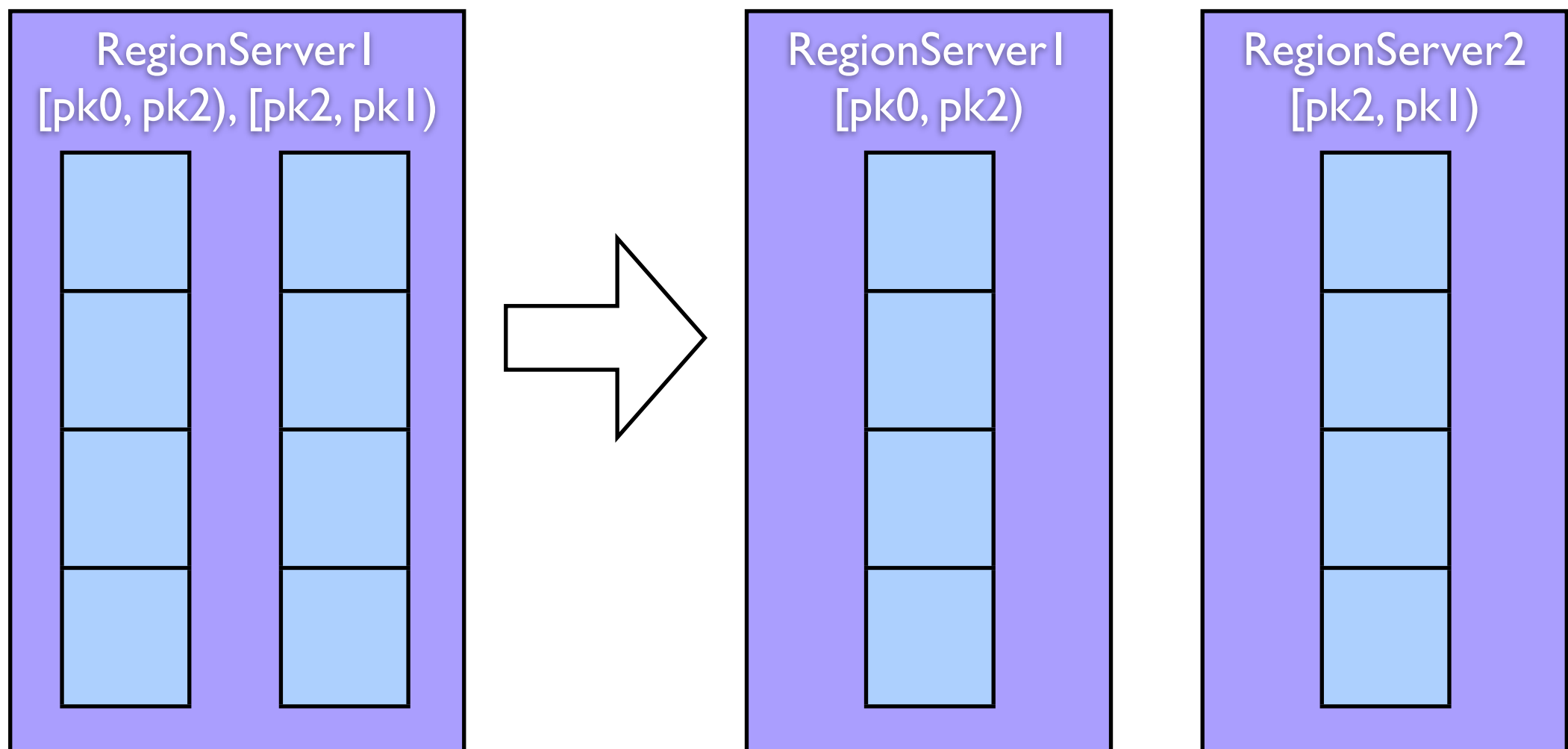
Partition Split

- every Region in the partition split into 2



Partition Split

- Why partition split? because we need more node



Partition Split

- More questions
 - when ordinary split? when partition split?
 - what happend when adding node?

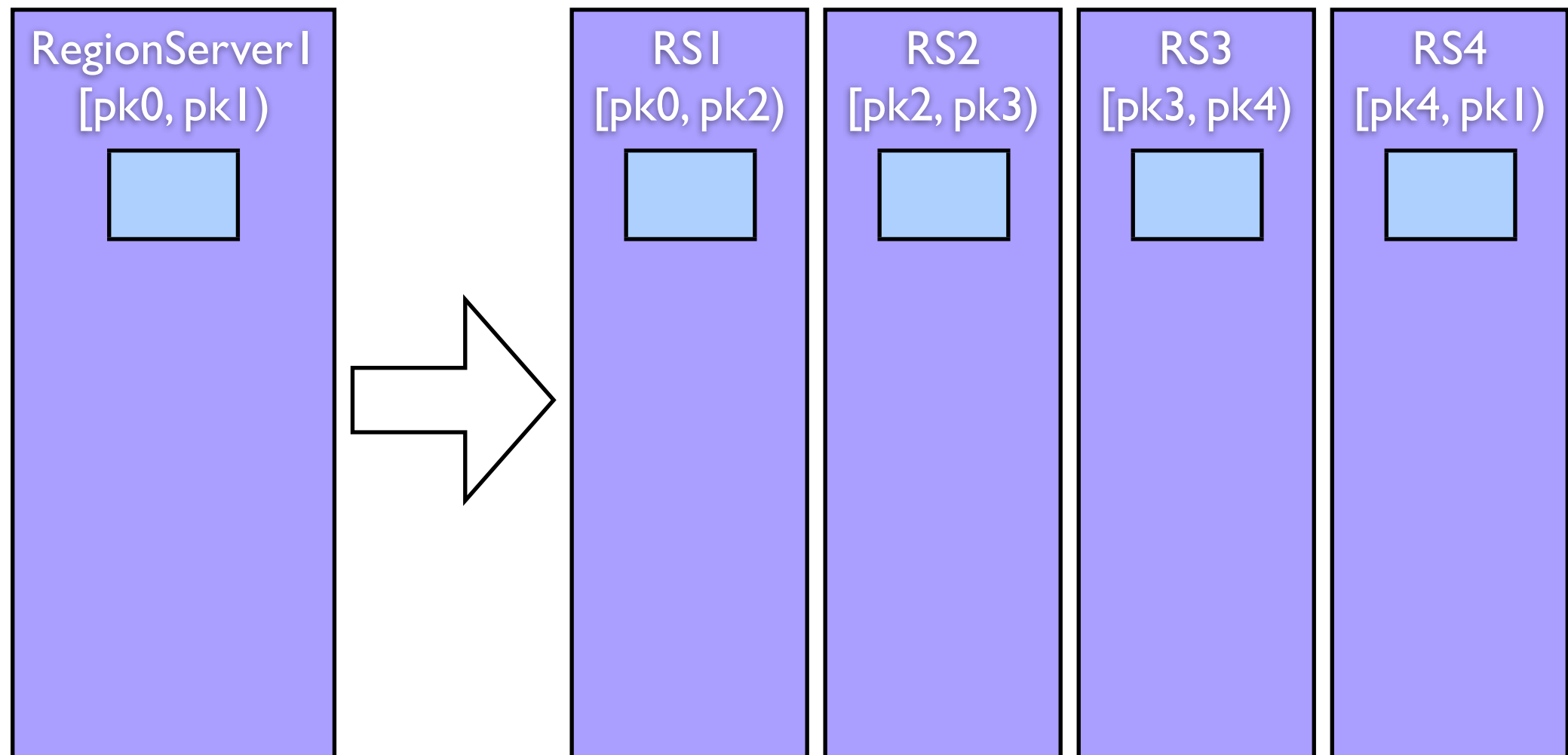
Split Policy

- rowKey split: same as now
- partition split: means we need more node for this HTable
 - regions size
 - query load

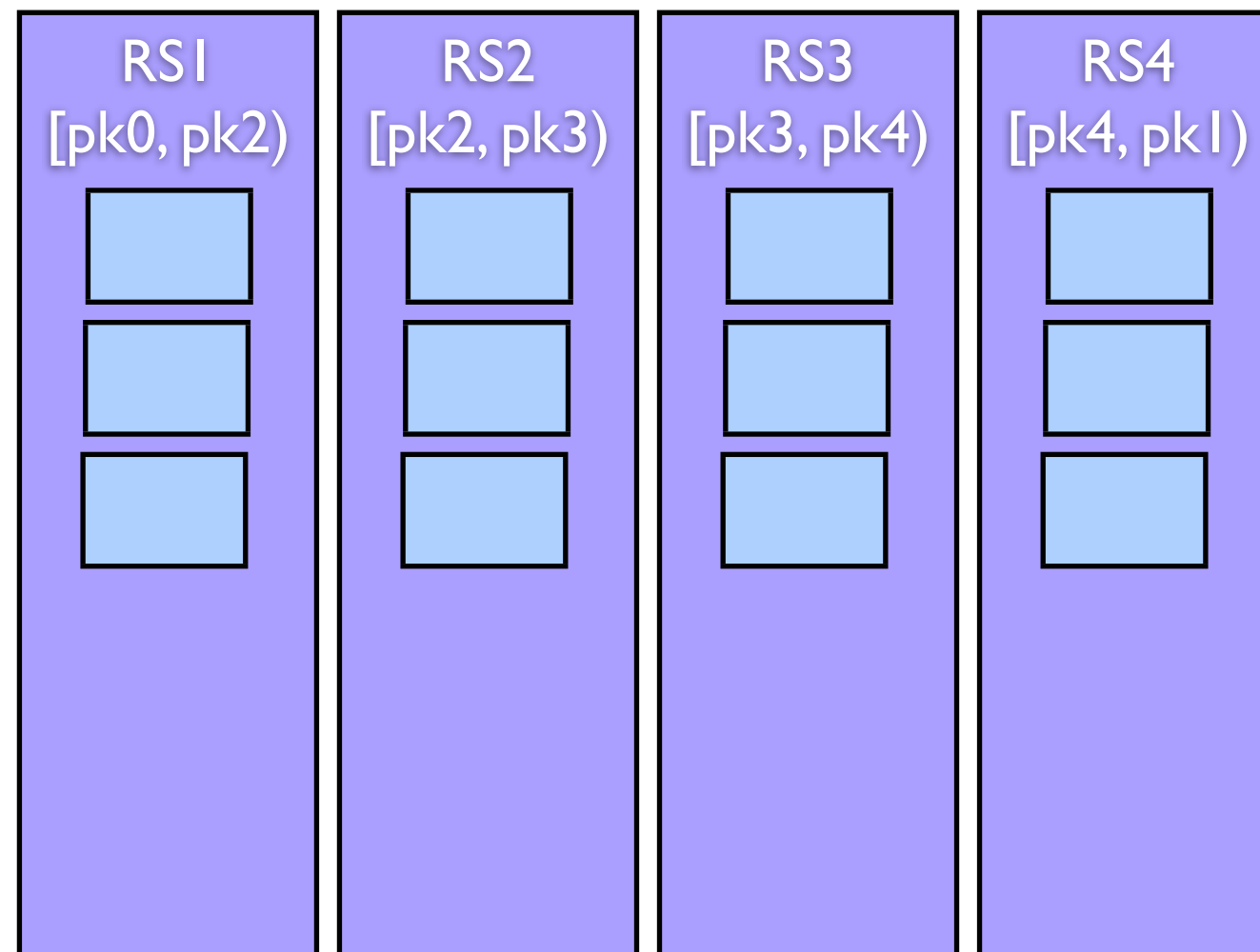
Split Policy Proposal

- partition split first
- with per table maximum partition limit
 - region would first pKey split to Pmax partitions, then rowKey split within each partition

Split Policy Proposal



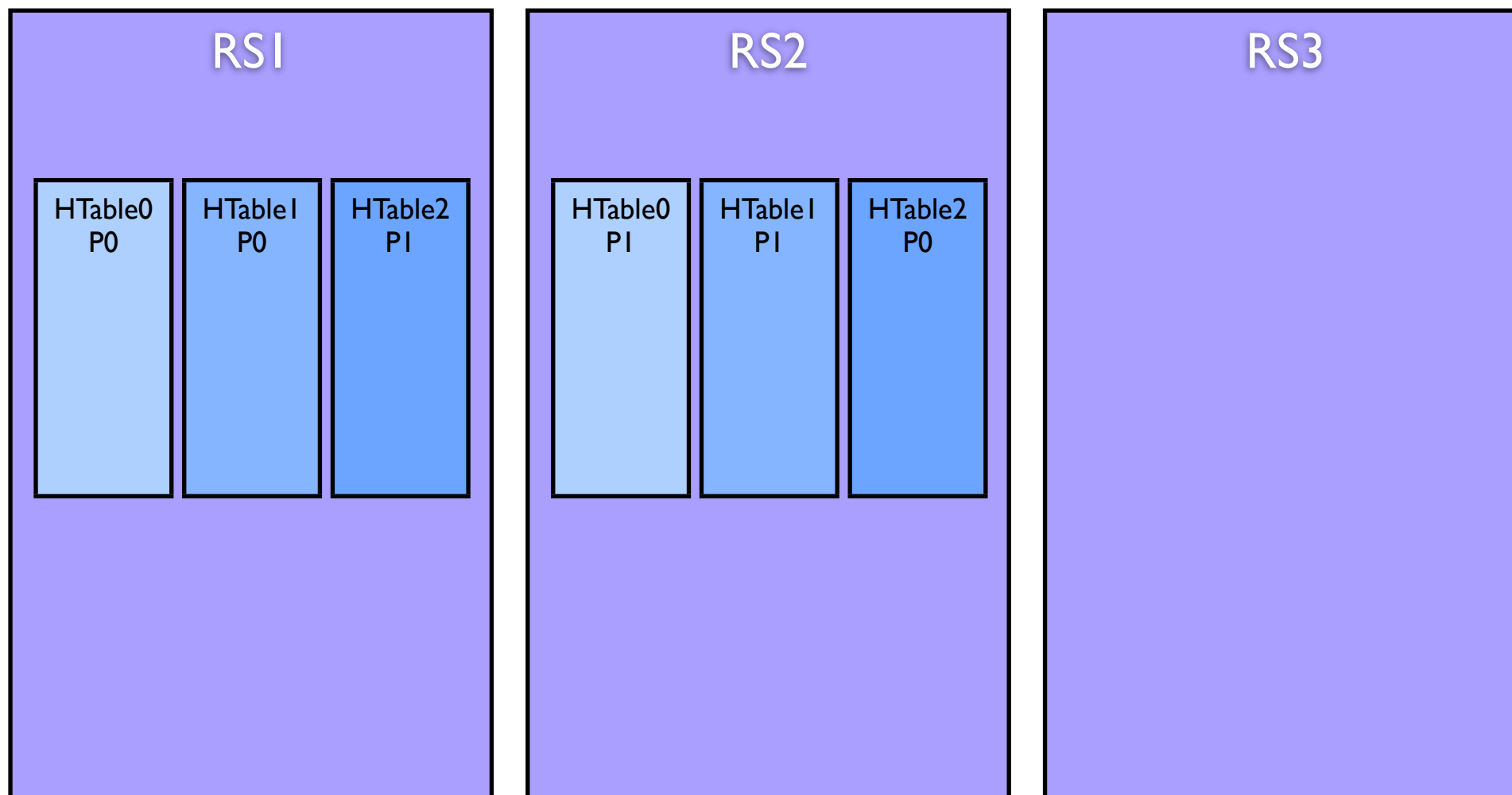
Split Policy Proposal



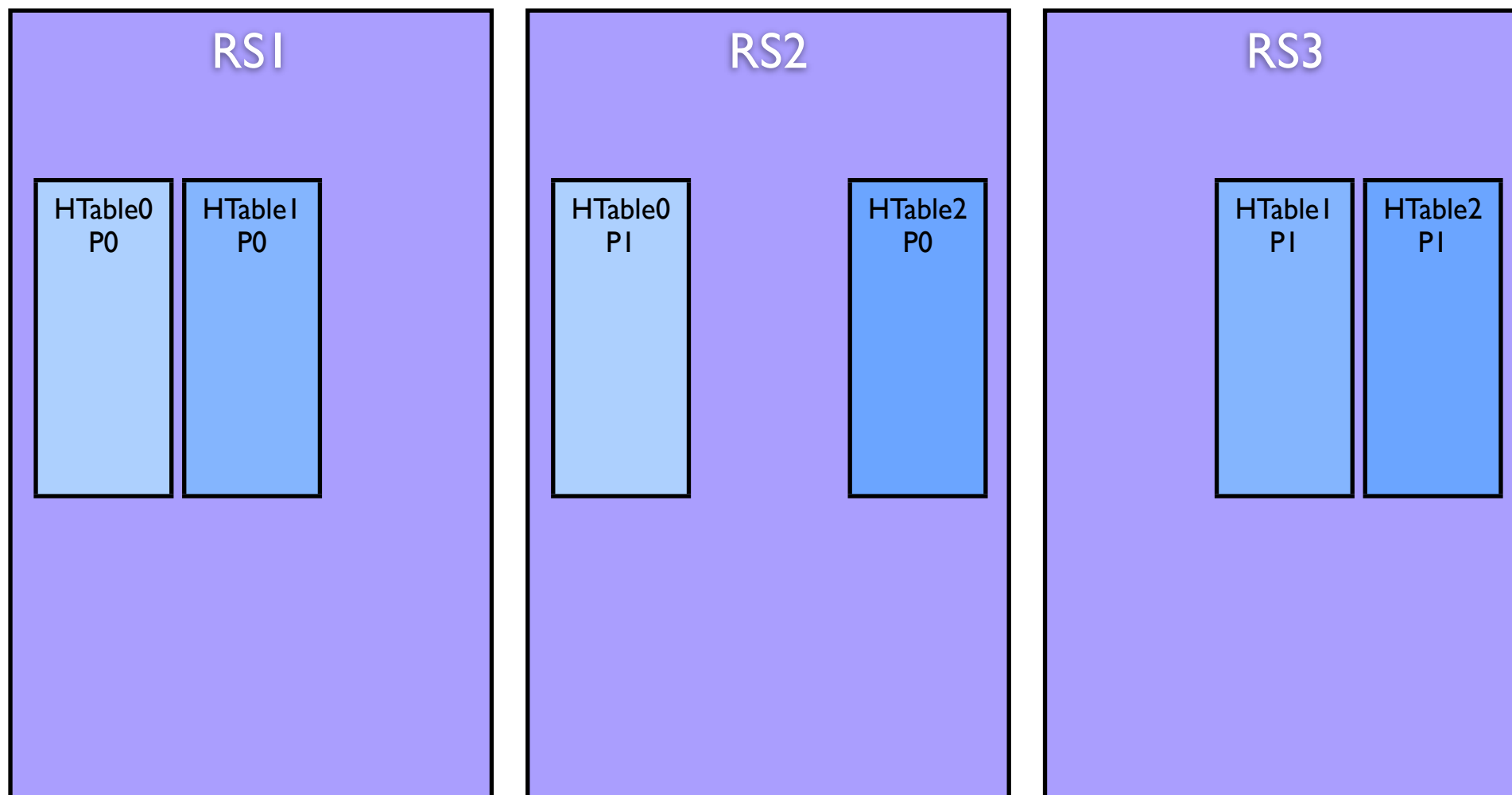
Adding Node Proposal

- Same as HBase
 - do not trigger split;
 - reassign partitions, like HBase reassign regions

Adding Node Proposal



Adding Node Proposal



Adding Node Proposal

- If a partition need to plit, but not enough node, adding node would trigger partition split.

Questions

- write into HBase?
- scan from HBase?
- split along partition key?
- split while writing?
- split while reading?
- split failover?
- Phase 2?

Phase 2:TableGroup

- Define 2 HTable with same Partition Def
 - 2 HTable with same partition info
 - 2 HTable partition-split simultaneously