

Designing a concurrent hash table

Concurrent Algorithms 2014
Vasileios Trigonakis

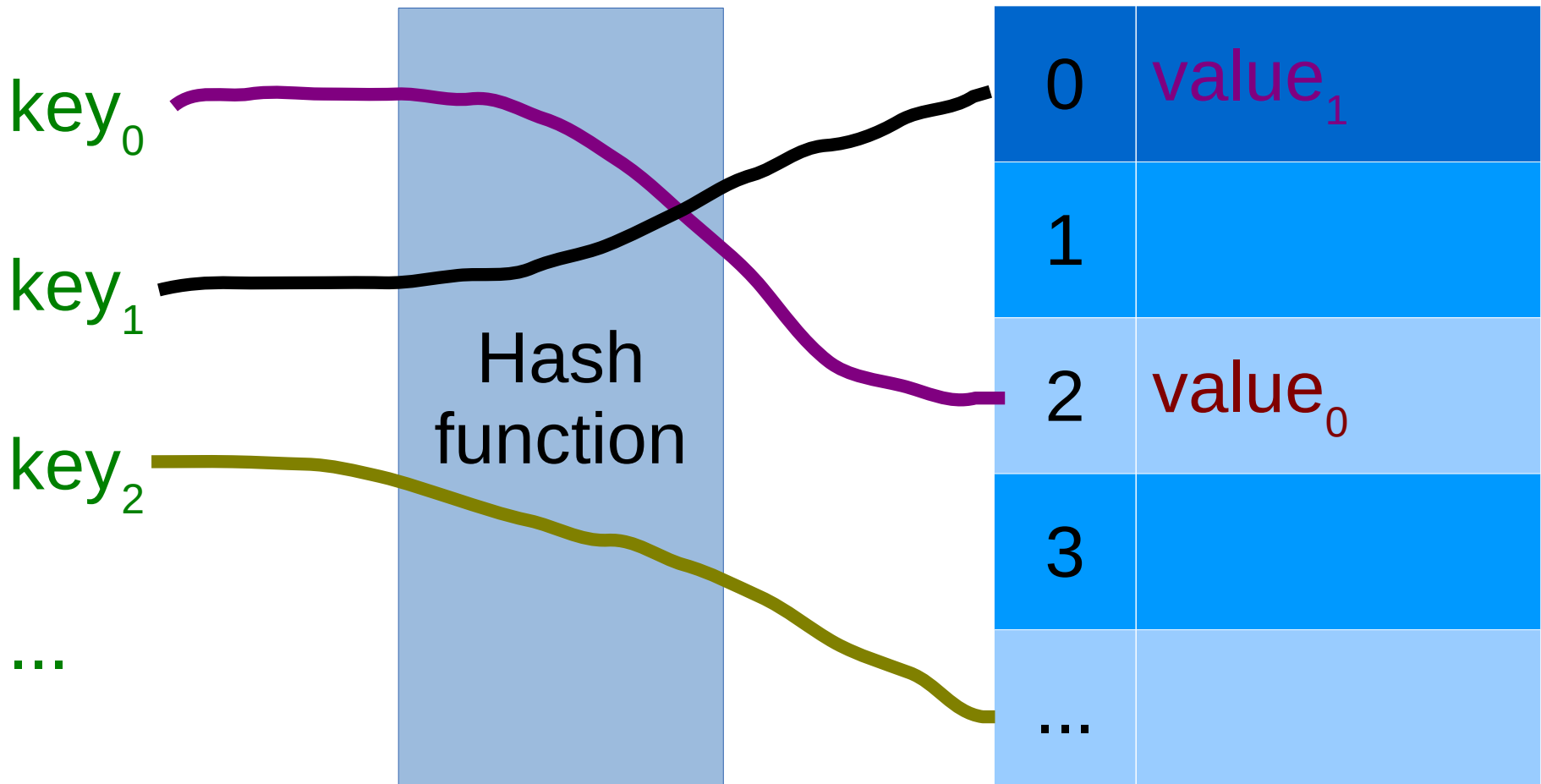
Hash table

- Data structure
 - map **keys** → **values**
- Organized as a set of buckets
 - that hold **key/value** mappings

Bucket	Key	Value
0	k0	v0
1	k1	v1
...

Hash table – Operation

Hash a given **key** → get the bucket to look in



Hash table – (Main) Interface

- put(key, val) → insert the key/val mapping
 - return false if key already exists
- rem(key) → removes the key/val mapping
 - return false if key not there
- get(key) → gets the val corresponding to key
 - return NULL if key not there

Concurrent Hash Table

- Multiple concurrent accesses
 - any number of processes
- Should provide **linearizable** operations
 - put, rem, get
 - ... should appear as atomic

Let's design and implement one :-)