YugabyteDB Sharding Strategies

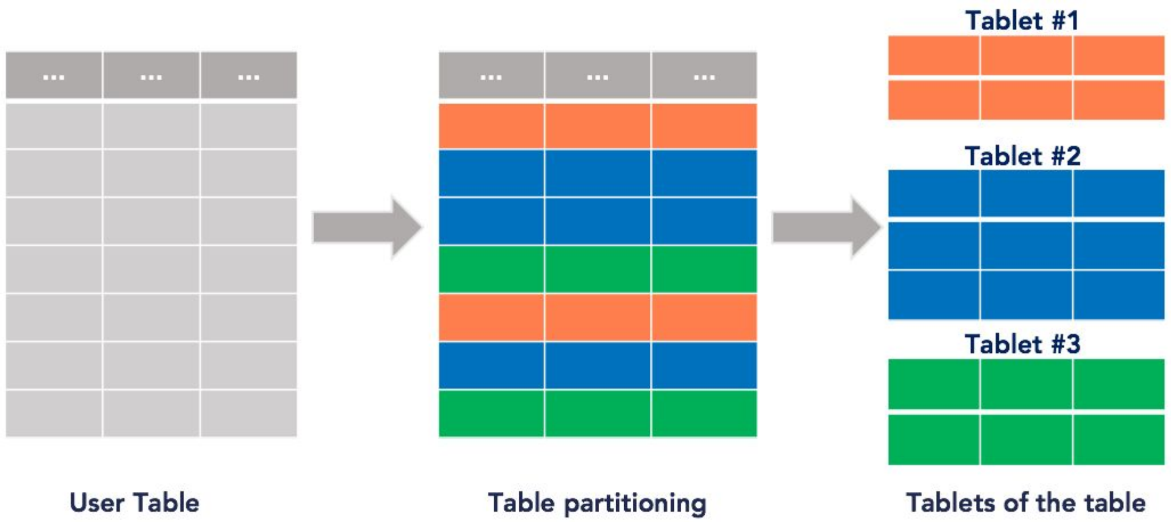
Friday, Jan/21/2022

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

yugabyteDB

What is sharding?

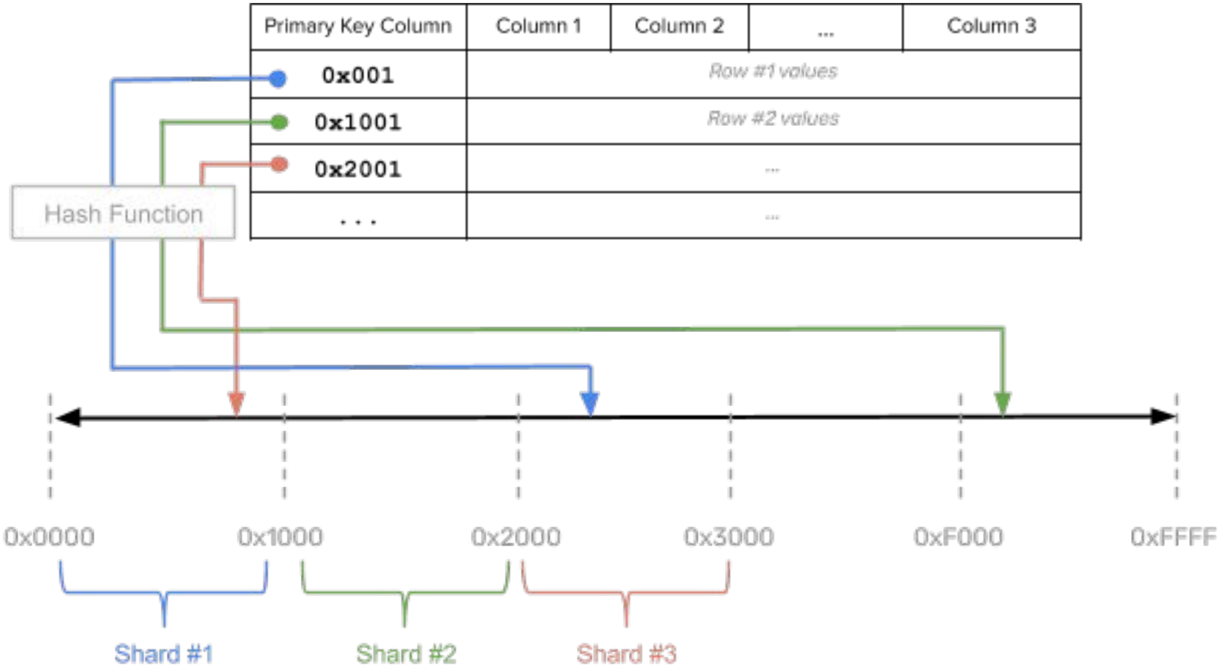
- Breaks up large user tables into smaller pieces, also called shards / tablets
- Spreads shards across database nodes, to distribute the load
- Each row in the user table gets mapped to exactly 1 shard
- Sharding in YugabyteDB is automatic
- Two flavors: Hash & Range



Hash Sharding

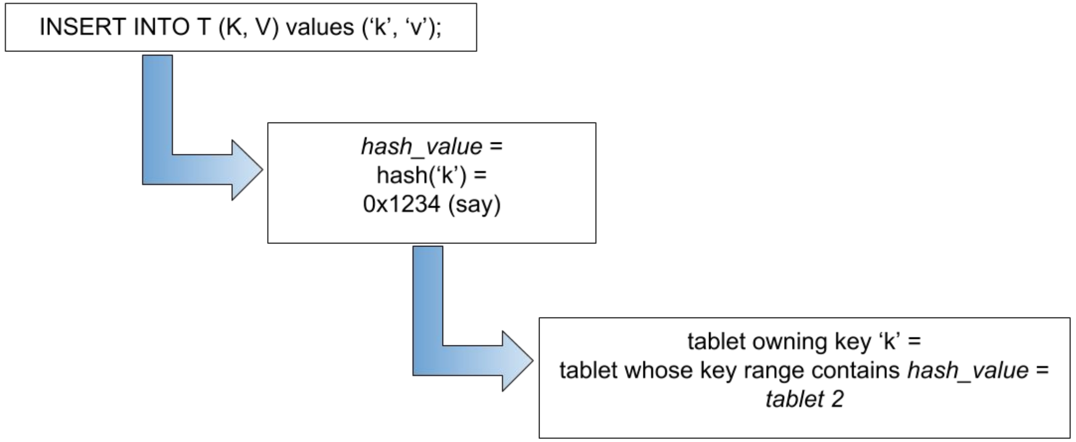
Consistent Hash Sharding: definition

- Random uniform data distribution across shards
- Hash function applied to primary key column



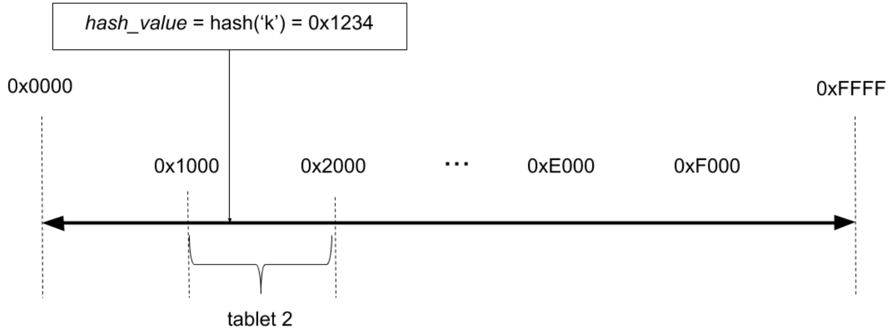
Primary Key Column	Column 1	Column 2	...	Column 3
0x001	Row #1 values			
0x1001	Row #2 values			
0x2001	...			
. . .	...			

Consistent Hash Sharding: routing


INSERT INTO T (K, V) values (‘k’, ‘v’);

hash_value =
hash(‘k’) =
0x1234 (say)

tablet owning key ‘k’ =
tablet whose key range contains *hash_value* =
tablet 2



Consistent Hash Sharding: syntax

```
CREATE TABLE orders (
    order_id int NOT NULL,
    physical_address text,
    email_address text,
    PRIMARY KEY (order_id  HASH)
);
```

```
CREATE TABLE order_details (
    order_id smallint NOT NULL,
    product_id smallint NOT NULL,
    unit_price real NOT NULL,
    quantity smallint NOT NULL,
    PRIMARY KEY (order_id  HASH, product_id ASC),
    FOREIGN KEY (product_id) REFERENCES products,
    FOREIGN KEY (order_id) REFERENCES orders
);
```

Consistent Hash Sharding: initial number of splits

- Default start with N * #tservers number of shards
- Can override explicitly on create with SPLIT INTO syntax

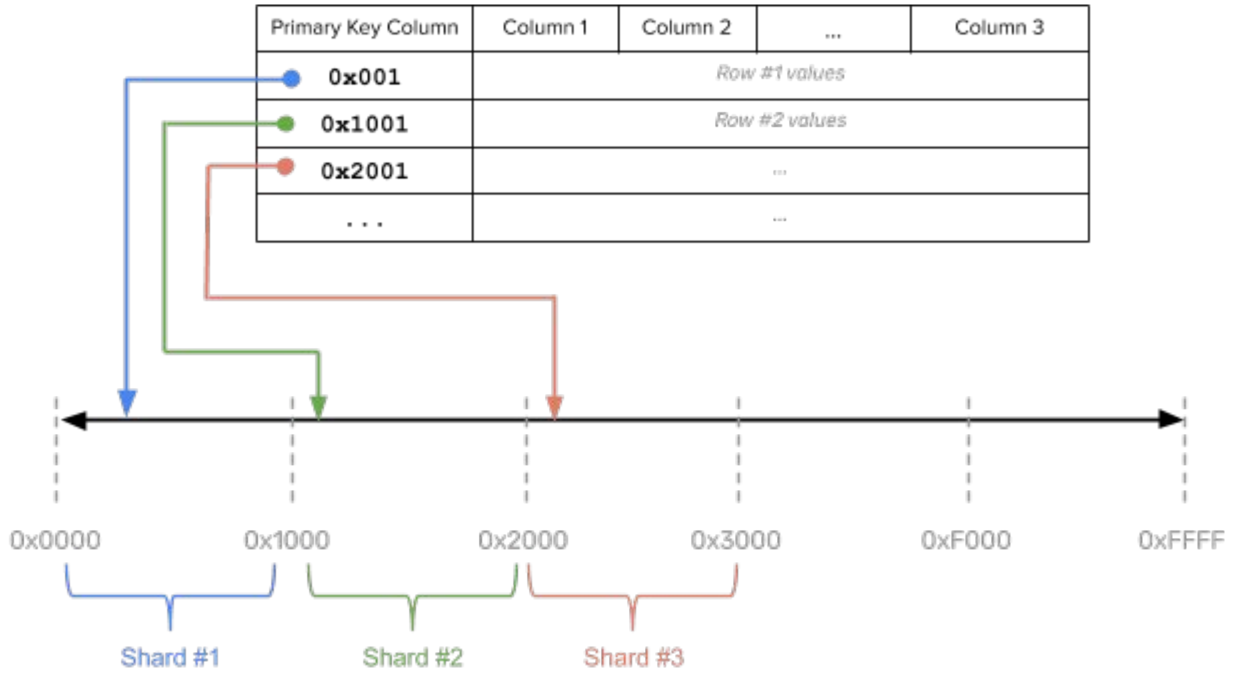
```
CREATE TABLE orders (
    order_id int NOT NULL,
    physical_address text,
    email_address text,
    PRIMARY KEY (order_id HASH)
);
```

```
CREATE TABLE orders2 (
    order_id int NOT NULL,
    physical_address text,
    email_address text,
    PRIMARY KEY (order_id HASH)
) SPLIT INTO 16 TABLETS;
```

Range Sharding

Range Sharding: definition

- Data is split into contiguous ranges, respecting the sort order of user data
- Hard to split upfront into several shards, as there's no knowledge of the input data distribution



Primary Key Column	Column 1	Column 2	...	Column 3
0x001	Row #1 values			
0x1001	Row #2 values			
0x2001	...			
. . .	...			

Range Sharding: syntax

```
CREATE TABLE order_details_range (
    order_id smallint NOT NULL,
    product_id smallint NOT NULL,
    unit_price real NOT NULL,
    quantity smallint NOT NULL,
    PRIMARY KEY (order_id  ASC, product_id  ASC),
    FOREIGN KEY (product_id) REFERENCES products,
    FOREIGN KEY (order_id) REFERENCES orders
);
```

Range Sharding: customizing initial split points

Starts off with one tablet, but dynamically splits over time

```
CREATE TABLE order_details_range (
    order_id smallint NOT NULL,
    product_id smallint NOT NULL,
    unit_price real NOT NULL,
    quantity smallint NOT NULL,
    PRIMARY KEY (order_id ASC, product_id ASC),
    FOREIGN KEY (product_id) REFERENCES
products,
    FOREIGN KEY (order_id) REFERENCES orders
);
```

Can override initial split points on create, if desired, eg: clear knowledge about key distribution

```
CREATE TABLE order_details_range (
    order_id smallint NOT NULL,
    product_id smallint NOT NULL,
    unit_price real NOT NULL,
    quantity smallint NOT NULL,
    PRIMARY KEY (order_id ASC, product_id ASC),
    FOREIGN KEY (product_id) REFERENCES
products,
    FOREIGN KEY (order_id) REFERENCES orders
) SPLIT AT VALUES ((1000), (2000));
```

Sharding type: comparison

	Consistent Hash Sharding	Range Sharding
Supports pre-splitting (to prevent database warming problem)?	Yes	No*
Can efficiently perform range scans on large datasets?	No	Yes
Prevents hotspots in the database (hence works well for massive scale)?	Yes	No

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB