


# Memo: Data Intensive Application

---

---

---

---

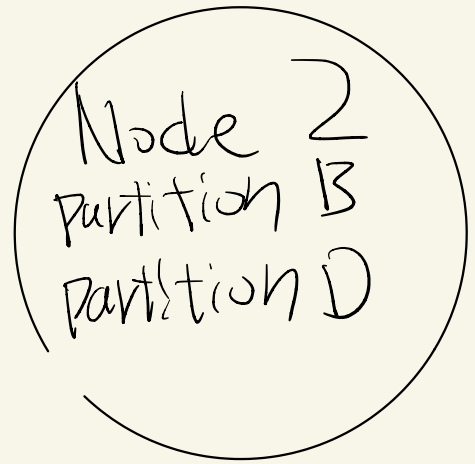
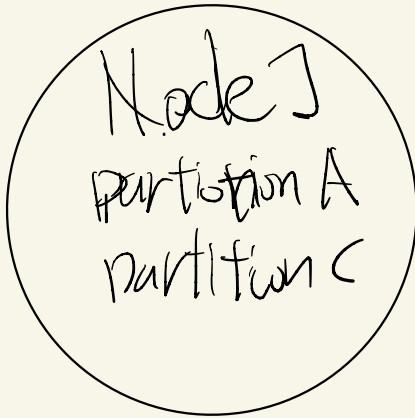


# Chapter 6: Partitioning (aka sharding)

A partition: small database of its own  
each piece of data belongs to one partition

Advantage of partitioning: Scalability

↑  
different partitions can be  
placed on different nodes



skewed partition: some partitions have more data or queries than others

A hot spot: a partition with disproportionately high load

## \* Partitioning of key-value data

### • Partitioning by key Range

- assign a continuous range of keys to each partition

#### Pros

- range scans are easy
- keys can be sorted in each partition

#### Cons

- hot spots may occur

## \* Partitioning by Hash of key

- assign a range of hashes to each partition

#### Pros

- keys can be distributed among the partitions

#### Cons

- range queries are inefficient

## □ Partitioning Secondary Indexes

### \* Partitioning by Document (aka local index)

- each secondary index is partitioned to the same one as the primary index

✓ may occur

Scatter / gather

- must send a query to all partitions
- = read queries are expensive

### \* Partitioning by Term (aka global index)

(term-partitioned)

- each secondary index is partitioned on the same node regardless of the node where its primary index is located

Pros — range scan is efficient

Cons

- writes are expensive

↑  
a write to a single document may affect multiple partitions of the index