5. Parallel Aggregation + MapReduce

GPT

Definition

Concurrency patterns.
Divide problems into smaller tasks and process these simultanously for a final aggragated result.
Requires independency.
Compliment each other.

Parallel aggregation

How to use it?

`Parallel.ForEach()` and `Parallel.For()` splits the workload in the for-loop into multiple threads and connects the result.

Example

```
Foreach(word in book)
{
     if (word.count > 6)
          bigWords++
}
```

this is slow → beneath is faster

```
Parallel.ForEach(book, word =>
{
        //lock(lockObj)
        //{
```

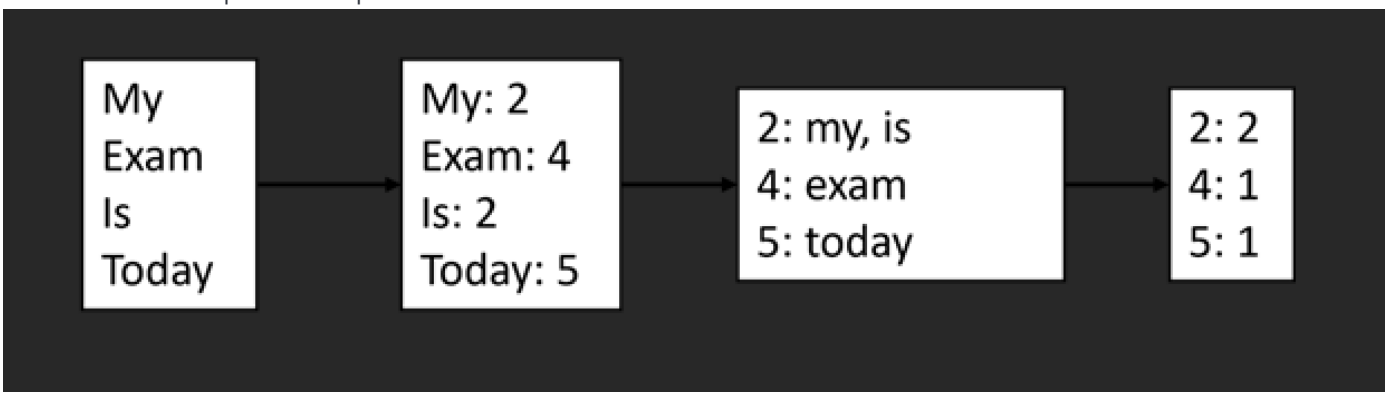
```
            if (word.count > 6)
            bigWords++
        //}
}
```

MapReduce

Used to find simple answer on big data sets. Can split data in specific groups.

Example

Distribute -> Map -> Group -> Reduce



SOLID

SRP

Comparison

Futures and pipelines