

Intro to Parallel and Concurrent Programming

Accumulator Combiner Reducer
Sequential Card Only Framework



[WashU CSE 2301](#)
Prof. [Dennis Cosgrove](#)
#11: Thu, Oct 02, 2025

Office Hours

Bookmarks

Worksheet

- write your name and id number clearly (do it **now**)
- commit your ideas to paper
- discuss and reach consensus at your table

SequentialSolution

```
Map<Student, Integer> map
for course in courses
    for student in course
        sum = map.getOrElse(student, 0)
        sum += course.getCredits()
        map.put(student, sum)

for entry in map.entrySet
    if entry.value < MIN_CREDITS
        send mail (entry.key)
```

MapReduce Solution

```
Mapper<Course, Student, Integer>
```

```
    void map( course )
```

```
        result = new list
```

```
        for student in course.getStudents()
```

```
            result.add( ( student, course.getCredits()) )
```

```
        return result
```

use the same int sum reducer we always use

```
Map<Student, Integer> map = MapReduce.hitIt(courses, mapper,  
reducer)
```

```
for entry in map.entrySet
```

```
    if entry.value < MIN_CREDITS
```

```
        send_mail (entry.key)
```

AAA

*** Who Is Missed?

Why not just slurp it in?

<https://database.lichess.org/>



2013 - January

17.8 MB

121,332

.pgn.zst / .torrent

FilterMapApp runs out of memory for a single month somewhere in 2015

Month

Size

Games

Download

2024 - January

32.4 GB

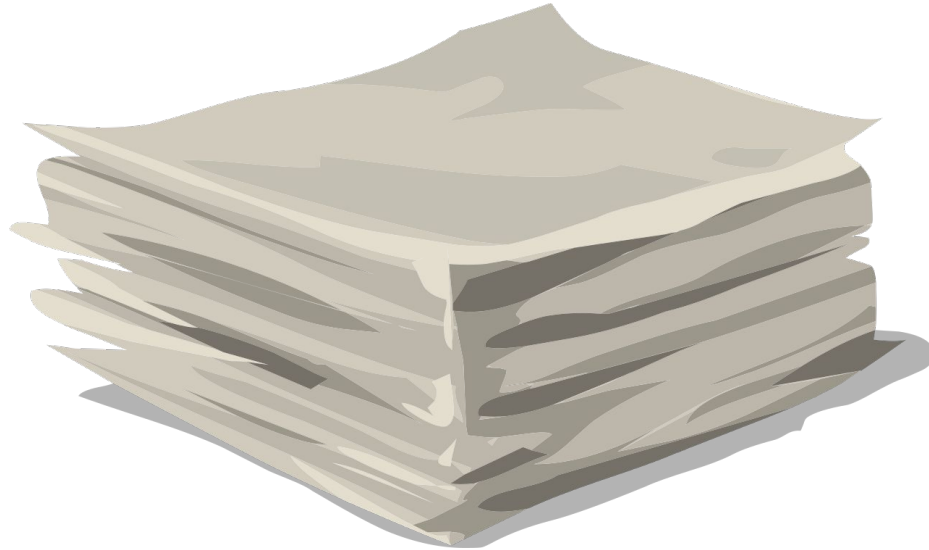
99,001,912

.pgn.zst / .torrent

Reflect, Synthesize, Ask, and Turn In Worksheets

Note: you are responsible for getting the worksheets to the us.

Do not leave a pile on the end of your table and hope.



AccumulatorCombinerReducer<V,A,R>

Card Only Sequential Map Reduce Framework

mapper and accumulatorCombinerReducer exist (use them)

```
private final Mapper<Deck, Suit, Integer> mapper;  
private final AccumulatorCombinerReducer<Integer, List<Integer>, Integer> accumulatorCombinerReducer;  
  
public CardOnlySequentialMapReduceFramework(Mapper<Deck, Suit, Integer> mapper,  
    AccumulatorCombinerReducer<Integer, List<Integer>, Integer> accumulatorCombinerReducer) {  
    this.mapper = mapper;  
    this.accumulatorCombinerReducer = accumulatorCombinerReducer;  
}  
  
@Override  
public Mapper<Deck, Suit, Integer> mapper() {  
    throw new NotImplementedException();  
}  
  
@Override  
public AccumulatorCombinerReducer<Integer, List<Integer>, Integer> accumulatorCombinerReducer() {  
    throw new NotImplementedException();  
}
```

mapAll -> accumulateAll -> reduceAll

```
@Override
public java.util.Map<Suit, Integer> mapReduceAll(Deck[] input) {
    List<List<Map.Entry<Suit, Integer>>> mapAllResult = mapAll(input);
    Map<Suit, List<Integer>> accumulateAllResult = accumulateAll(mapAllResult);
    return reduceAll(accumulateAllResult);
}
```

*** What should mapAll do?

```
private List<List<Map.Entry<Suit, Integer>>> mapAll(Deck[] input) {  
    throw new NotImplementedException();  
}
```

*** What should accumulateAll do?

```
private Map<Suit, List<Integer>> accumulateAll  
    (List<List<Map.Entry<Suit, Integer>>> mapAllResults) {  
    throw new NotImplementedException();  
}
```

*** What should reduceAll do?

```
private Map<Suit, Integer> reduceAll  
    (Map<Suit, List<Integer>> accumulateAllResult) {  
    throw new NotImplementedException();  
}
```

Path To Victory

DefaultEntry



CardMapper

Reducer



Seq Card Framework