Parallel Boxes & Jam

Jose Rodriguez
Rotem David
Robert Tolda
Hahn Chong
Fred Clark Jr.

PB & J Introduction

- Language for Distributed Computing (2+ processors)
- Useful for Parallelism

  •Amdahl’s Law : speedup = 1/(1-P)

  where P is fraction of code that can be run in parallel

  •speedup = 1/((P/N)+S)

  where N is number of processors and S is the serial fraction
- Useful for Fault Tolerant Execution – Critical Applications

Team Progress

Feature vs Module

- Feature
- Implement in all files
- Create Tests
- Merge into Master (GitHub)

PB & J Library

- Data Types
  - Long, Double, Boolean, String
  - Map, Array
- 2 Functions make this language unique
  - Spread
  - Jam

Arrays and Maps

- For maximizing the potential to solve distributed problems.
- Useful for distributing jobs and retrieving their results.

Jam and Spread

```
master(map slaves, array args){ ... Runtime argument.
        ...

        array result <- jam: spread: factor(@searchStarts, n, |slaves|);
        print("Result: " ~ result);
}

array factor(array starts, long n, long slaves) {
        ...
}
```

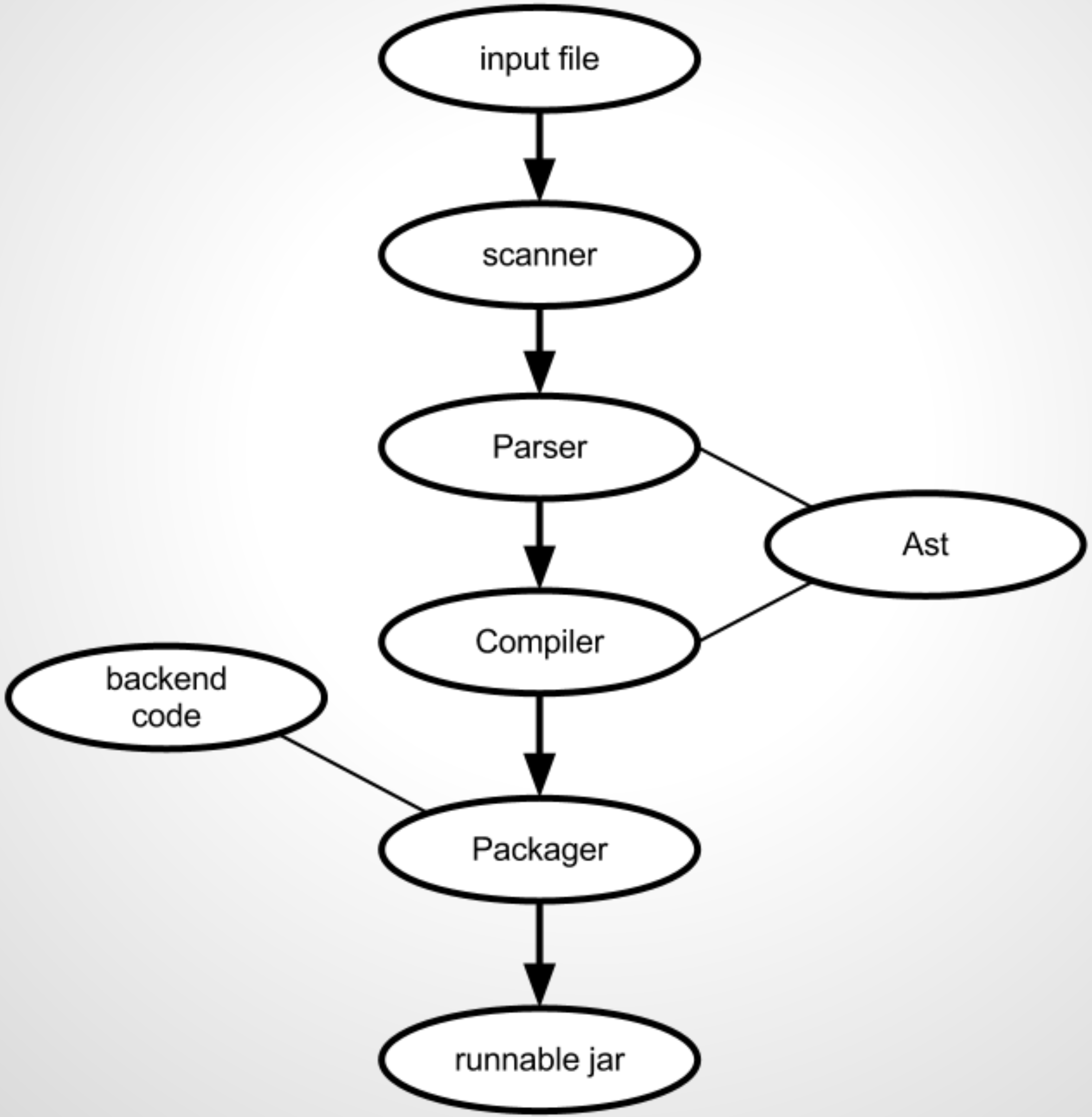
Prime Factorization

```
master(map slaves, array args){ ... Runtime argument.
    long n <- args[0];
    array searchStarts;

... get the place for each slave to start
    long iterations <- (n / |slaves|) - 1; ... size of slaves
    long start <- 1;
    for(long m <- 0; m < |slaves|; m <- m + 1) {
        searchStarts[m] <- start + (m * 2);
    }

... spread the starting points to the slaves
    array result <- jam: spread: factor(@searchStarts, n, |slaves|);
    print("Result: " ~ result);
}
```

Prime Factorization

```
array factor(array starts, long n, long slaves) {
        long start <- starts[0];
        array factors;
        for(long i <- start; i <= n / 2; i <- i + (2 * slaves)) {
                print("Trying " ~ i);
                if( i > 1 && n % i = 0) { ...it is not prime
                        factors[|factors|] <- i;
                }
        }
        if(|factors| > 0) {
                -> factors;
        }
        -> null;
}
```

PB & J Code Generation



PB&J Execution

- Running on Slaves
  - java -jar PBJ.jar -slave [port]
- Running on Master
  - java -jar PBJ.jar slave_ip[:port];ip2;ip3... args

Lessons & Advice

- Make sure everyone knows how to use version control
- Meet regularly and work consistently
- Don’t try to put too much in your language