DiSh: Dynamic Shell-Script Distribution

Tammam Mustafa MIT

Konstantinos Kallas Penn University of Pennsylvania

Pratyush Das Purdue University

Nikos Vasilakis Brown

binpa.sh

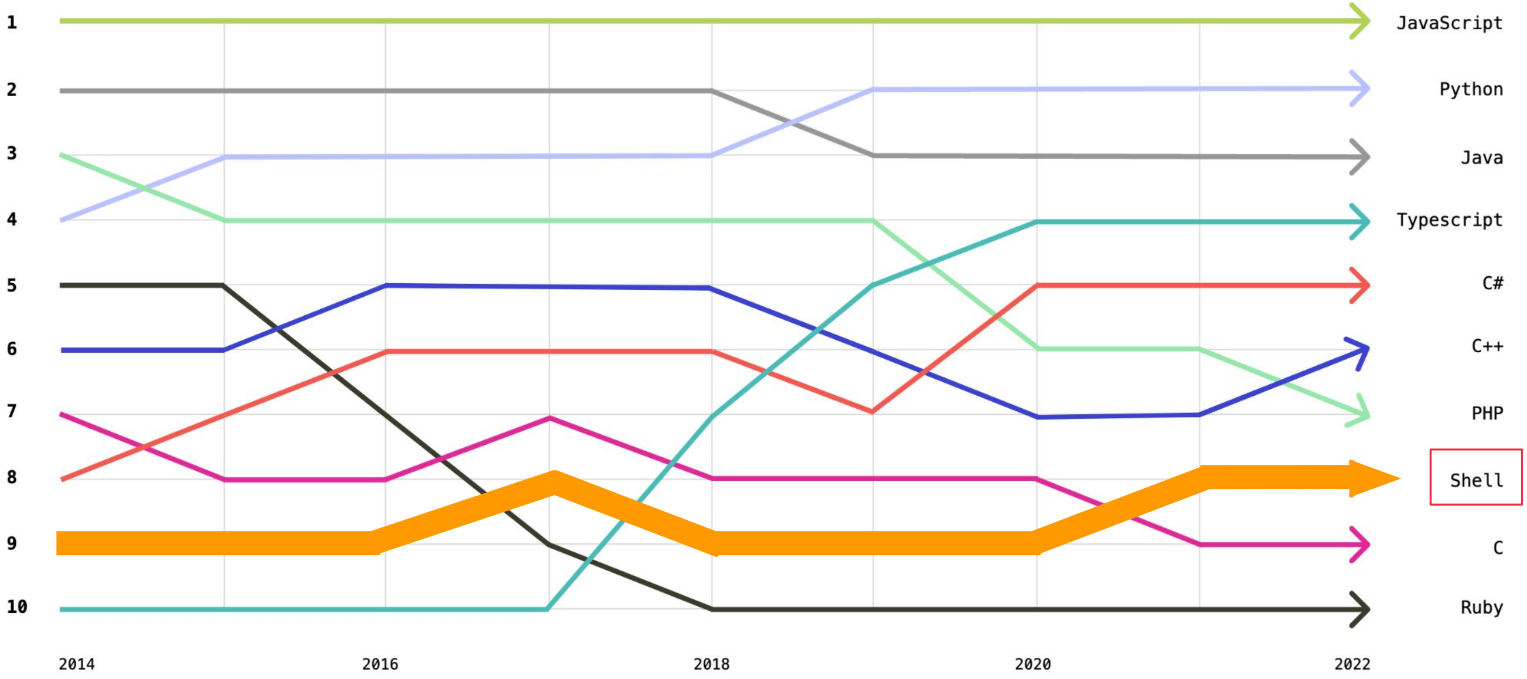
github.com/binpash/dish

THE LINUX FOUNDATION

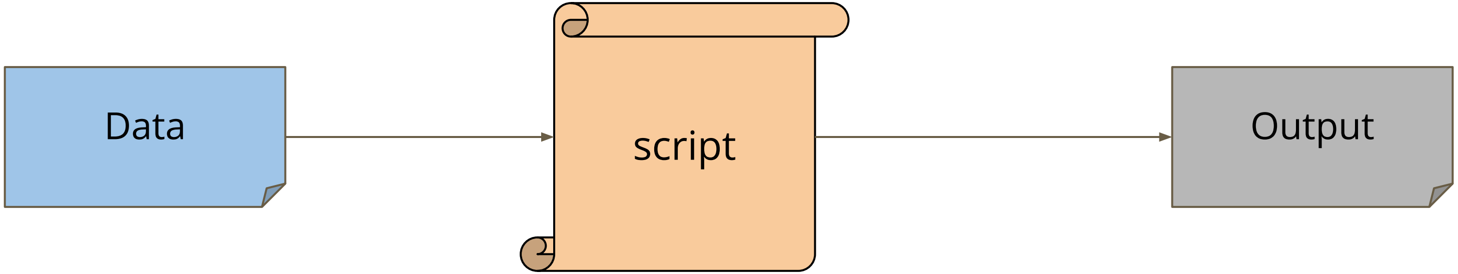
Shells 🐚 are everywhere

Shells 🐚 are everywhere

from the 2022 state of the octoverse: https://octoverse.github.com



A general data processing script



An example: Temperature Analysis

```
0097007070999992015092520244+00000+000000FM-15+707099999V02099
99C000019999999N999999999+02901+00401999999REMMET069MOBOB0

0097007070999992015092520304+00000+000000FM-15+707099999V02099
99V000519999999N999999999+03001+00501999999REMMET069MOBOB0

0097007070999992015092520354+00000+000000FM-15+707099999V02099
99C000019999999N999999999+03001+00501999999REMMET069MOBOB0

0097007070999992015092520404+00000+000000FM-15+707099999V02019
01N000519999999N999999999+03201+00601999999REMMET069MOBOB0

0097007070999992015092520454+00000+000000FM-15+707099999V02099
99C000019999999N999999999+03201+00601999999REMMET069MOBOB0

0097007070999992015092520504+00000+000000FM-15+707099999V02099
99C000019999999N999999999+03201+00501999999REMMET069MOBOB0

0097007070999992015092520554+00000+000000FM-15+707099999V02099
99C000019999999N999999999+03101+00501999999REMMET069MOBOB0

0097007070999992021004+00000+000000FM-15+707099999V02099
```

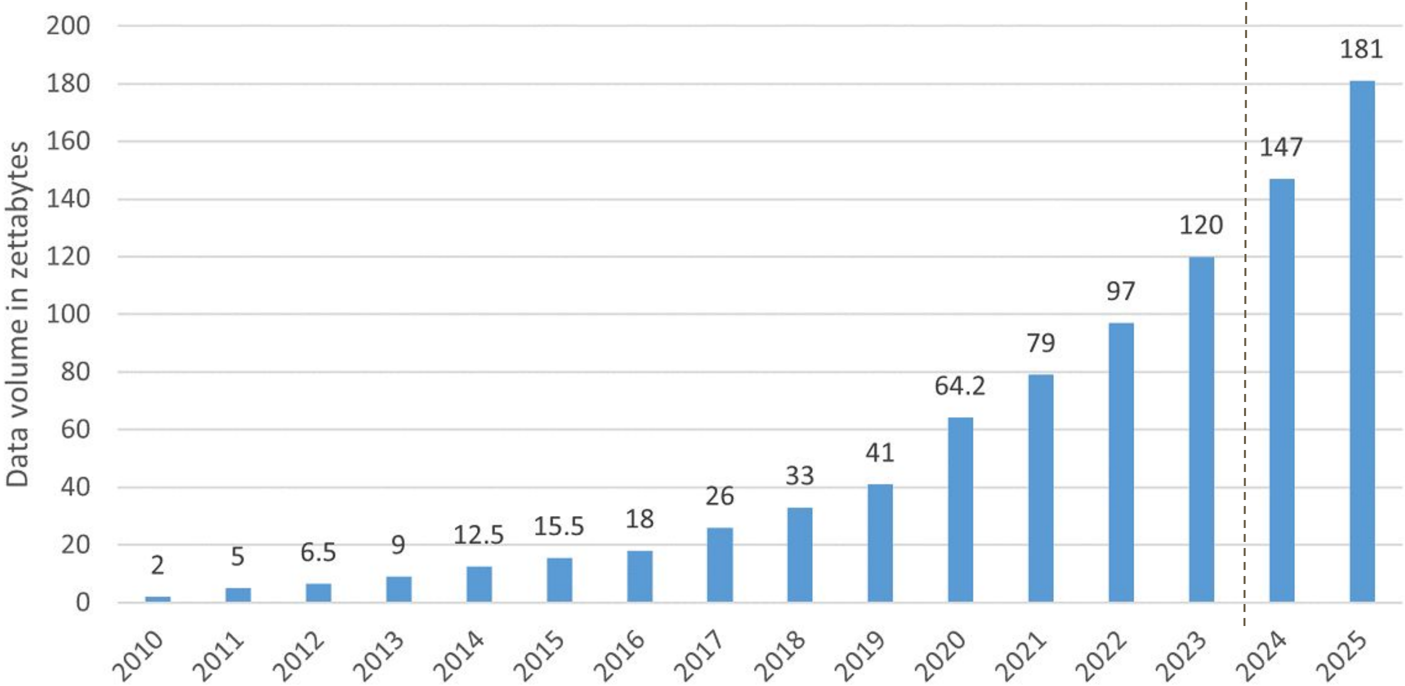
One day

```
TEMPS="temps.txt"

cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn |
head -n1 > max.txt
```

Takes < 1s

Data is exploding

Volume of data created and replicated worldwide (source: IDC)



An example: Temperature Analysis

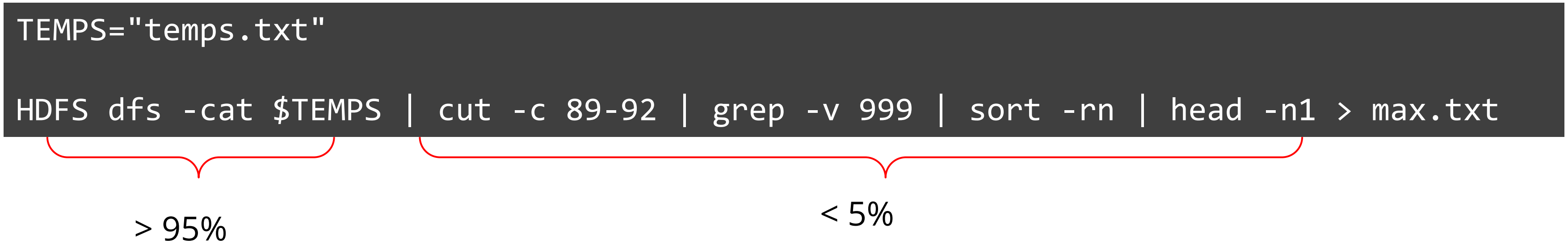
- Data from National Oceanic and Atmospheric Administration (NOAA).
- Stored in HDFS (Hadoop distributed file system).

```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn | head -n1 > max.txt
```

The works but ...

200s

3.6G



```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn | head -n1 > max.txt
```

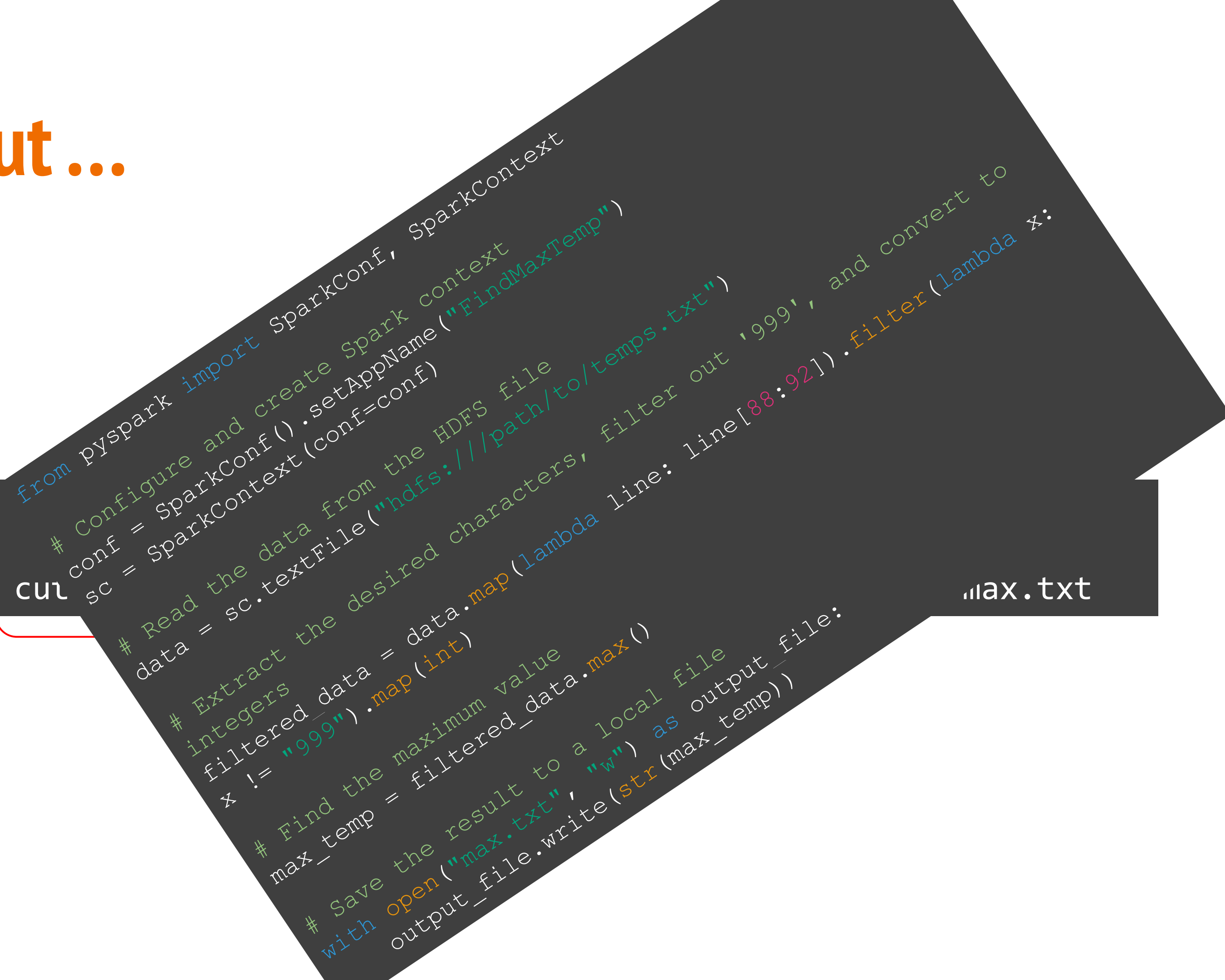
> 95%

< 5%

The works but ...

```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cu                                          ax.txt
```

> 95%

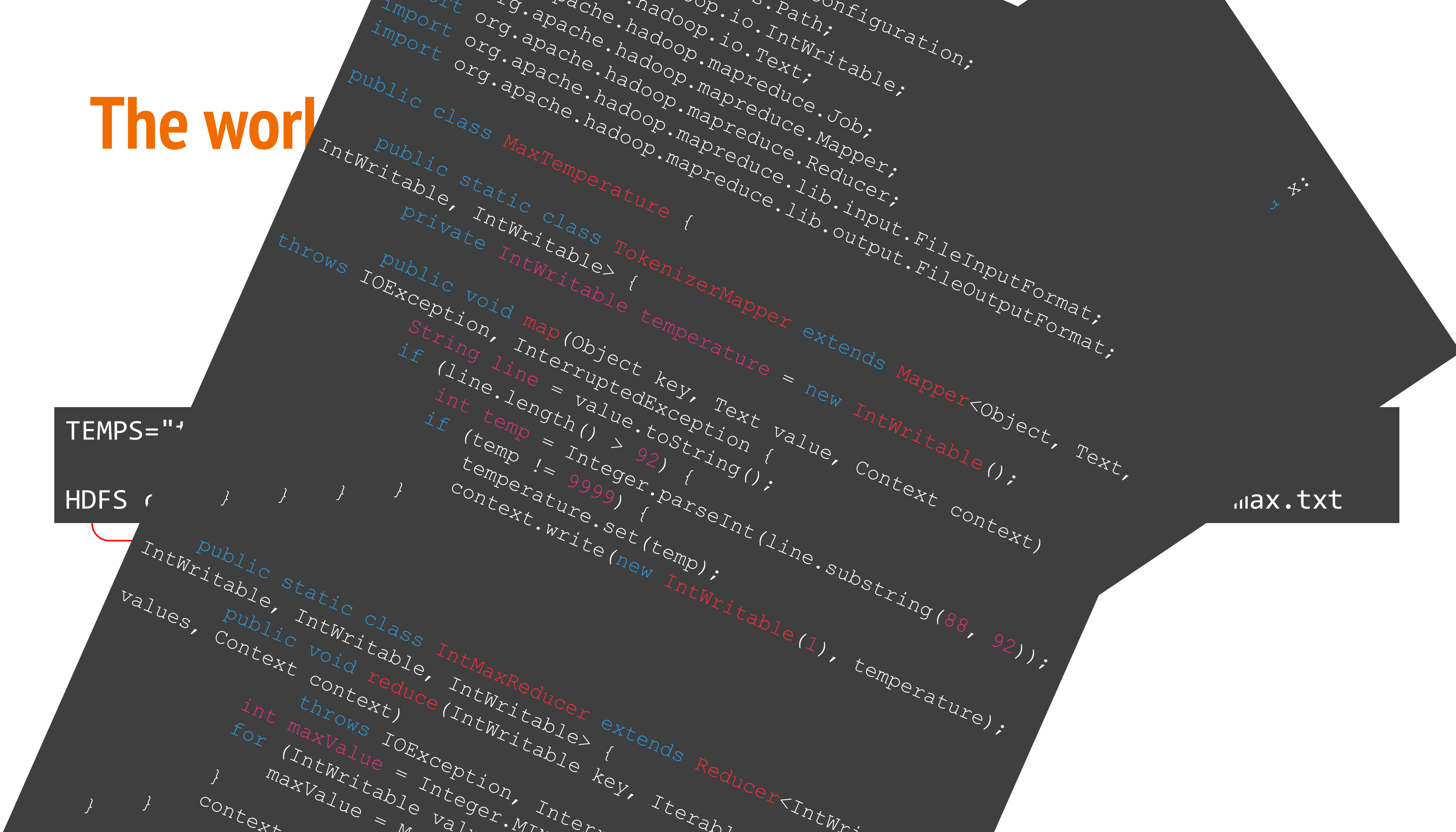


```python
from pyspark import SparkConf, SparkContext

# Configure and create Spark context
conf = SparkConf().setAppName("FindMaxTemp")
sc = SparkContext(conf=conf)

# Read the data from the HDFS file
data = sc.textFile("hdfs:///path/to/temps.txt")

# Extract the desired characters, filter out '999', and convert to
integers
filtered_data = data.map(lambda line: line[88:92]).filter(lambda x:
x != "999").map(int)

# Find the maximum value
max_temp = filtered_data.max()

# Save the result to a local file
with open("max.txt", "w") as output_file:
    output_file.write(str(max_temp))
```

The worl

```
TEMPS="

HDFS
```

```
max.txt
```

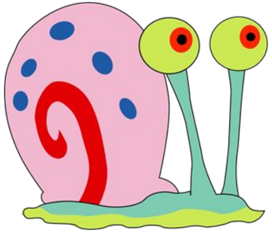


```
...onfiguration;
...s.Path;
...op.io.IntWritable;
...hadoop.io.Text;
...hadoop.mapreduce.Job;
...pache.hadoop.mapreduce.Mapper;
ort ...g.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class MaxTemperature {
  public static class TokenizerMapper extends Mapper<Object, Text,
IntWritable, IntWritable> {
    private IntWritable temperature = new IntWritable();
    public void map(Object key, Text value, Context context)
throws IOException, InterruptedException {
      String line = value.toString();
      if (line.length() > 92) {
        int temp = Integer.parseInt(line.substring(88, 92));
        if (temp != 9999) {
          temperature.set(temp);
          context.write(new IntWritable(1), temperature);
        }
      }
    }
  }
  public static class IntMaxReducer extends Reducer<IntWri
IntWritable, IntWritable, IntWritable> {
    public void reduce(IntWritable key, Iterabl
values, Context context)
      throws IOException, Inter
      int maxValue = Integer.MI
      for (IntWritable val
        maxValue = M
      }
      context
    }
  }
```

```
x:
```

DiSh 🍽️

Turns this ...

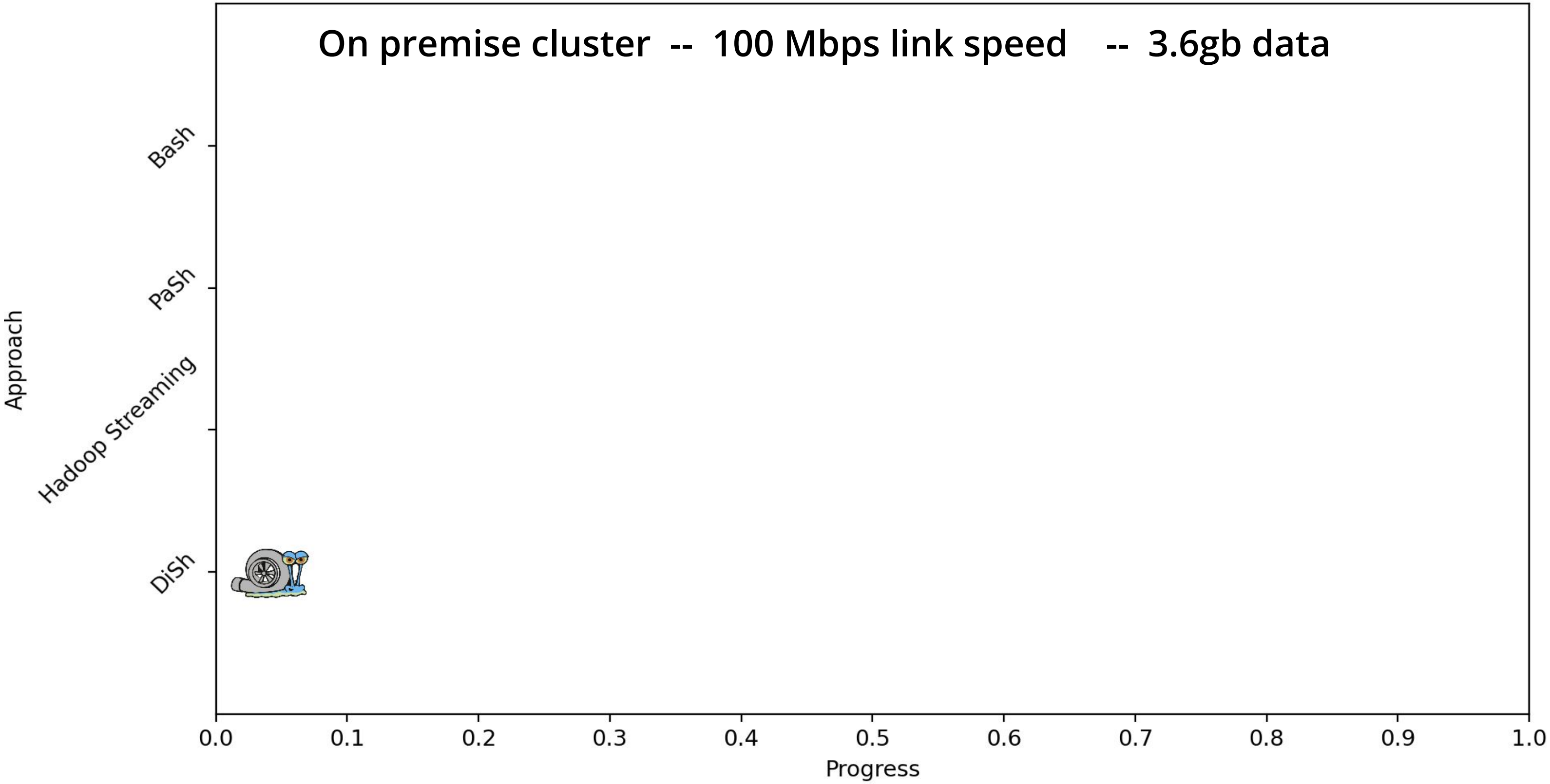


```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn | head -n1 > max.txt
```

Into this ...

```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn | head -n1 > max.txt
```

Performance Comparison



On premise cluster -- 100 Mbps link speed -- 3.6gb data

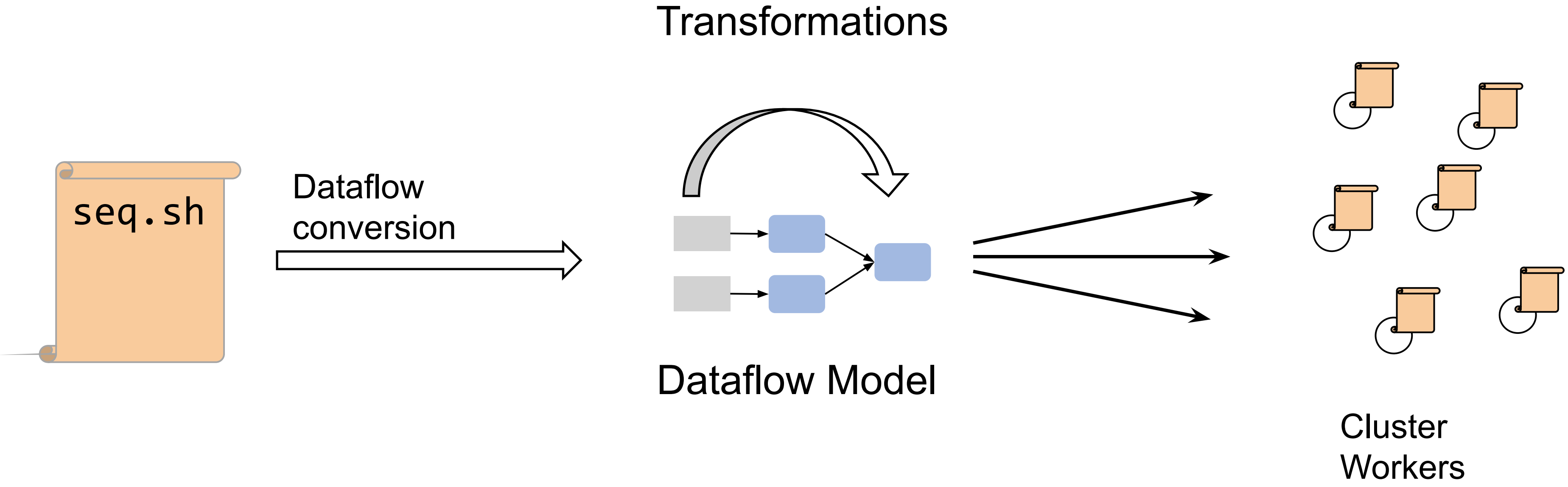
200s

190s

90s

7s

DiSh



seq.sh

Dataflow conversion

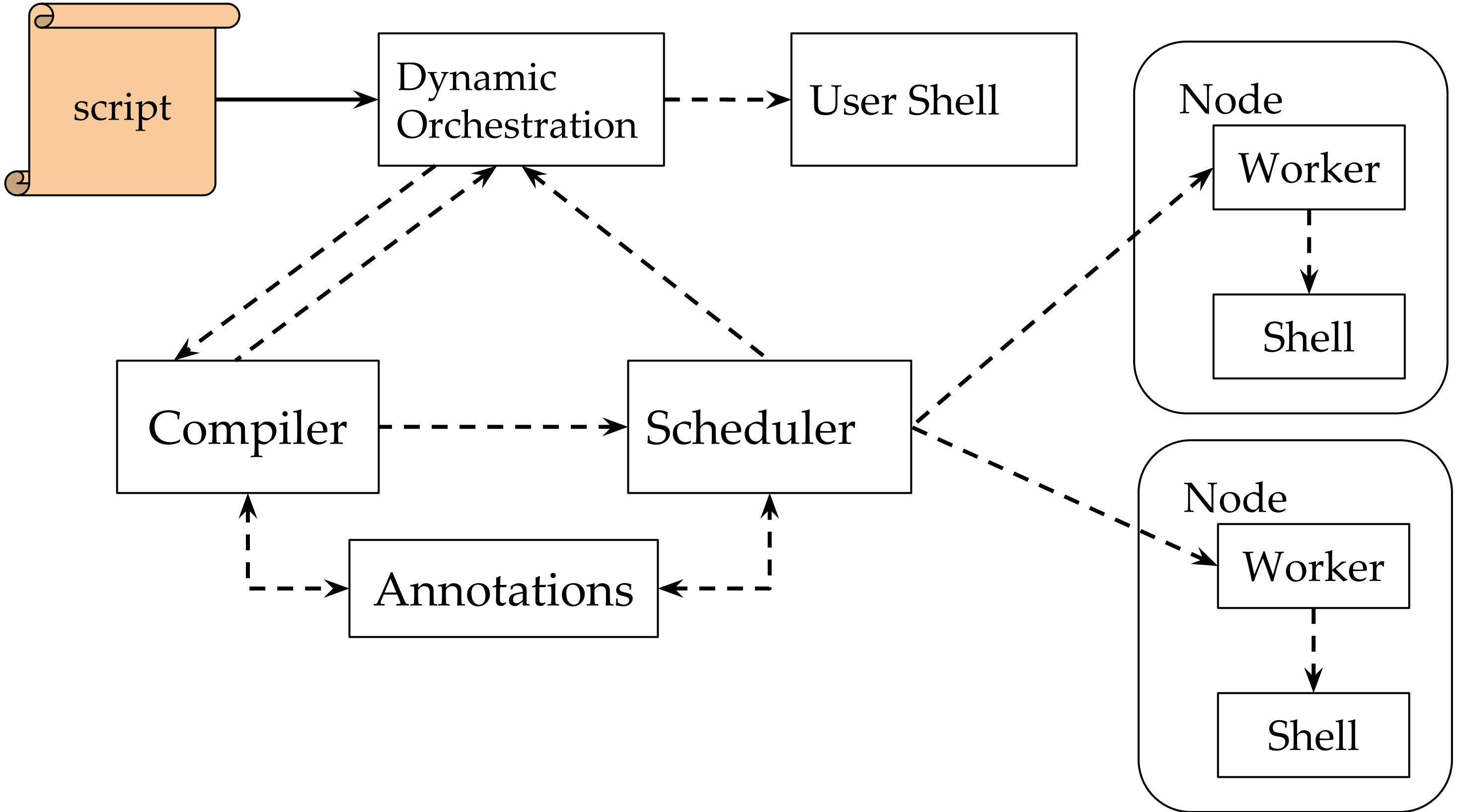
Transformations

Dataflow Model

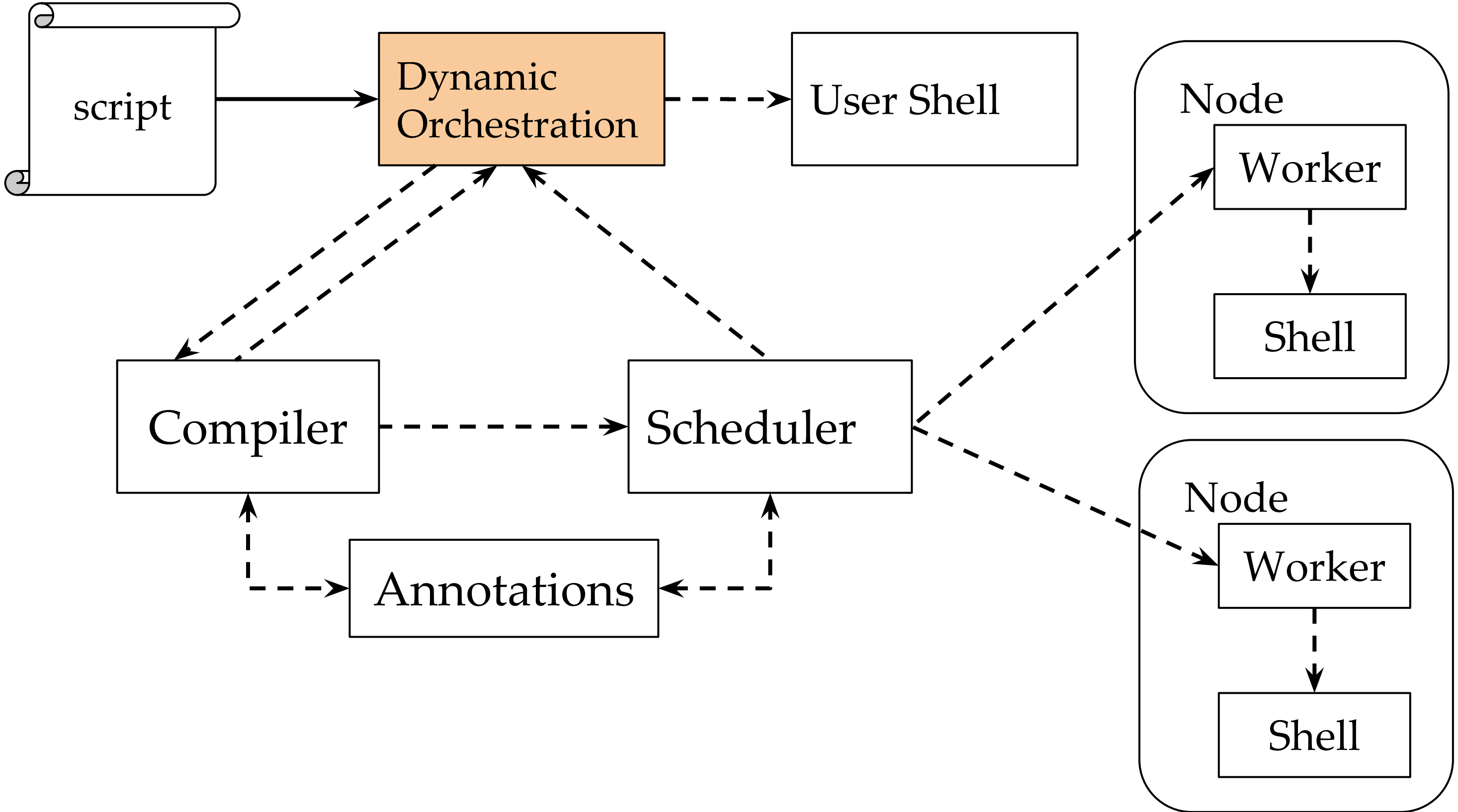
Cluster Workers

No tight coupling: Could work on top of any shell!

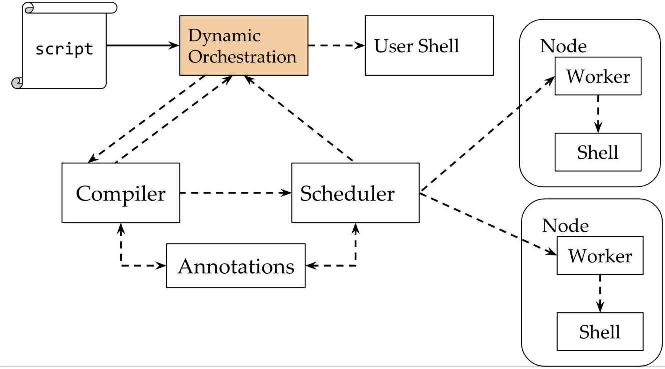
DiSh Overview

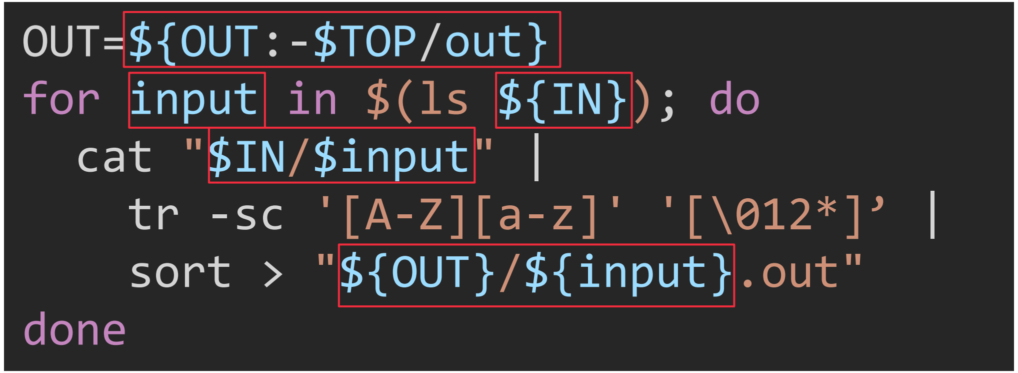


DiSh Overview



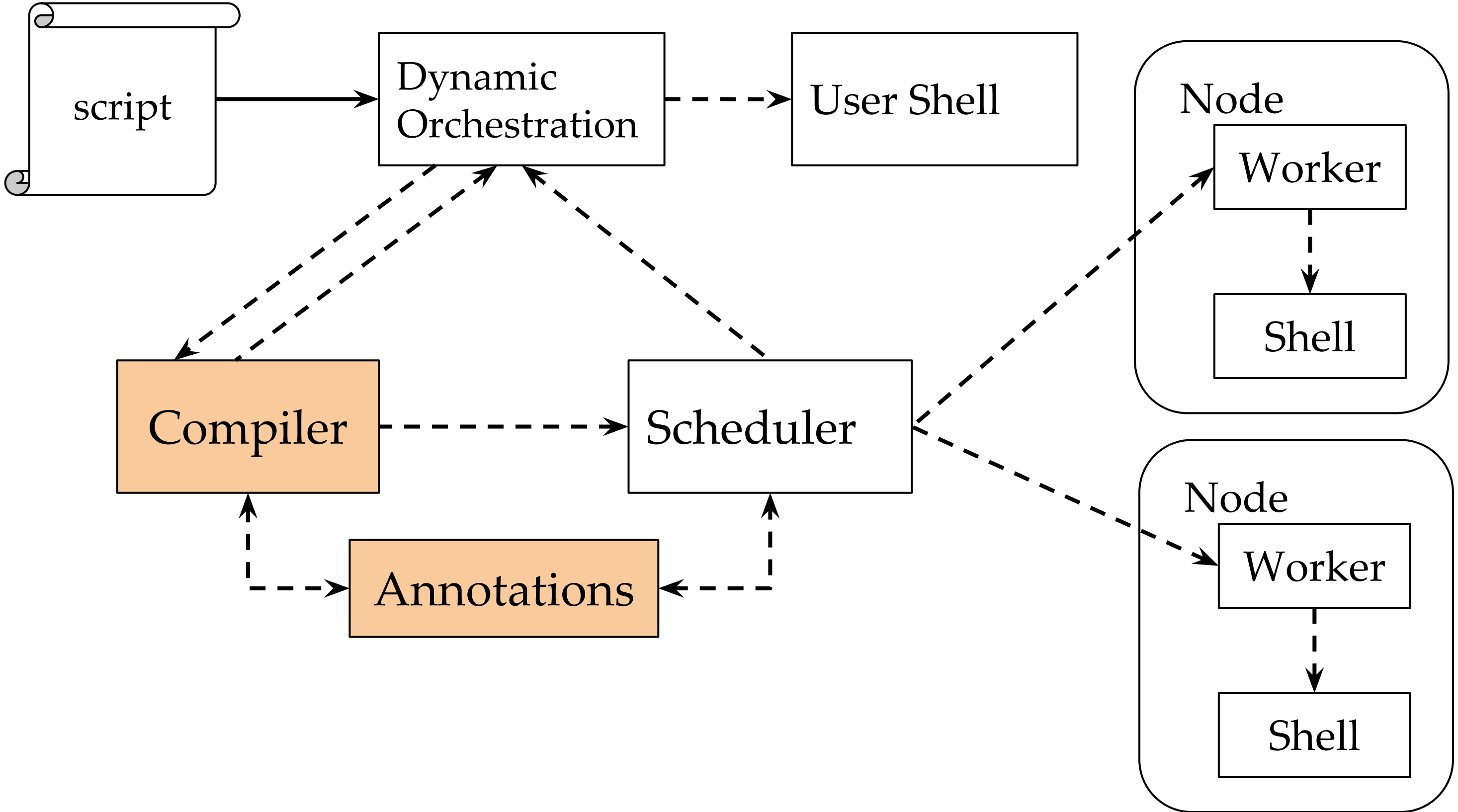
The shell is extremely Dynamic



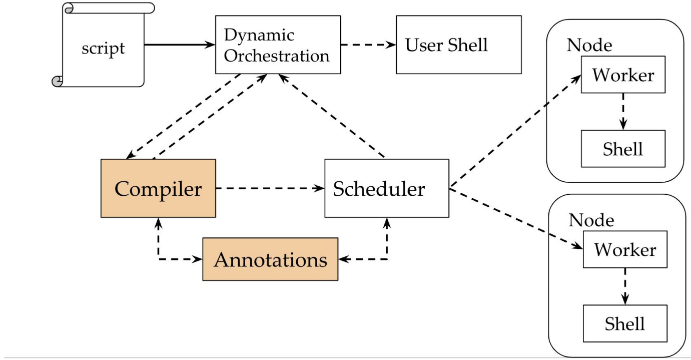


```
OUT=${OUT:-$TOP/out}
for input in $(ls ${IN}); do
  cat "$IN/$input" |
    tr -sc '[A-Z][a-z]' '[\012*]’ |
    sort > "${OUT}/${input}.out"
done
```

DiSh Overview

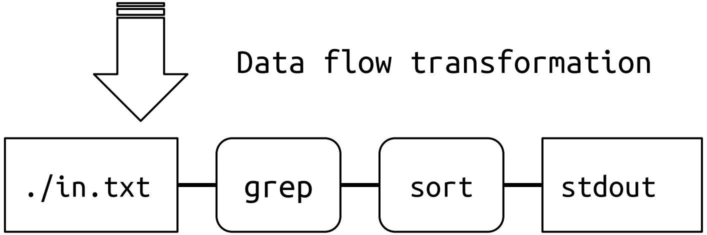


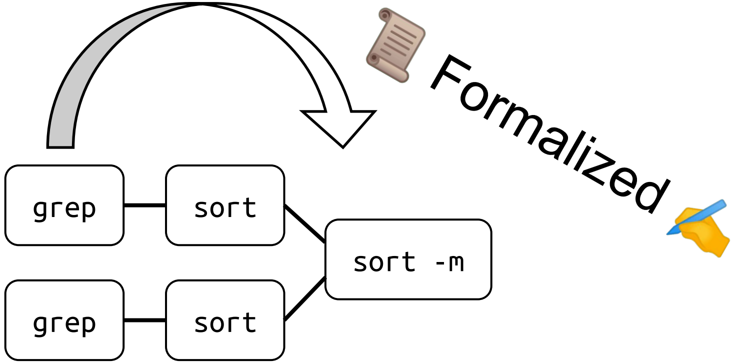
Dataflow Transformations



Order Aware Dataflow Model

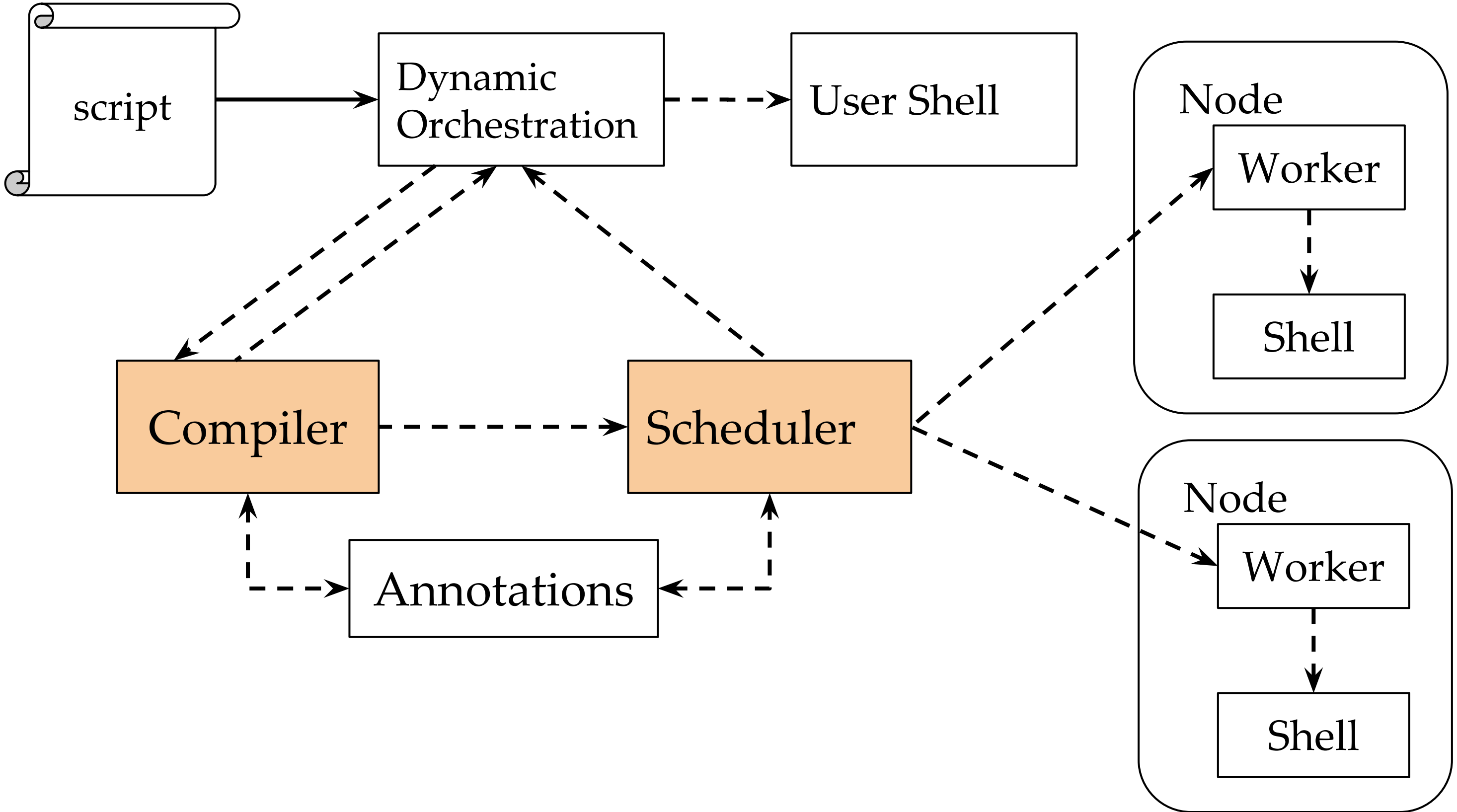
```
grep hello ./in.txt | sort
```



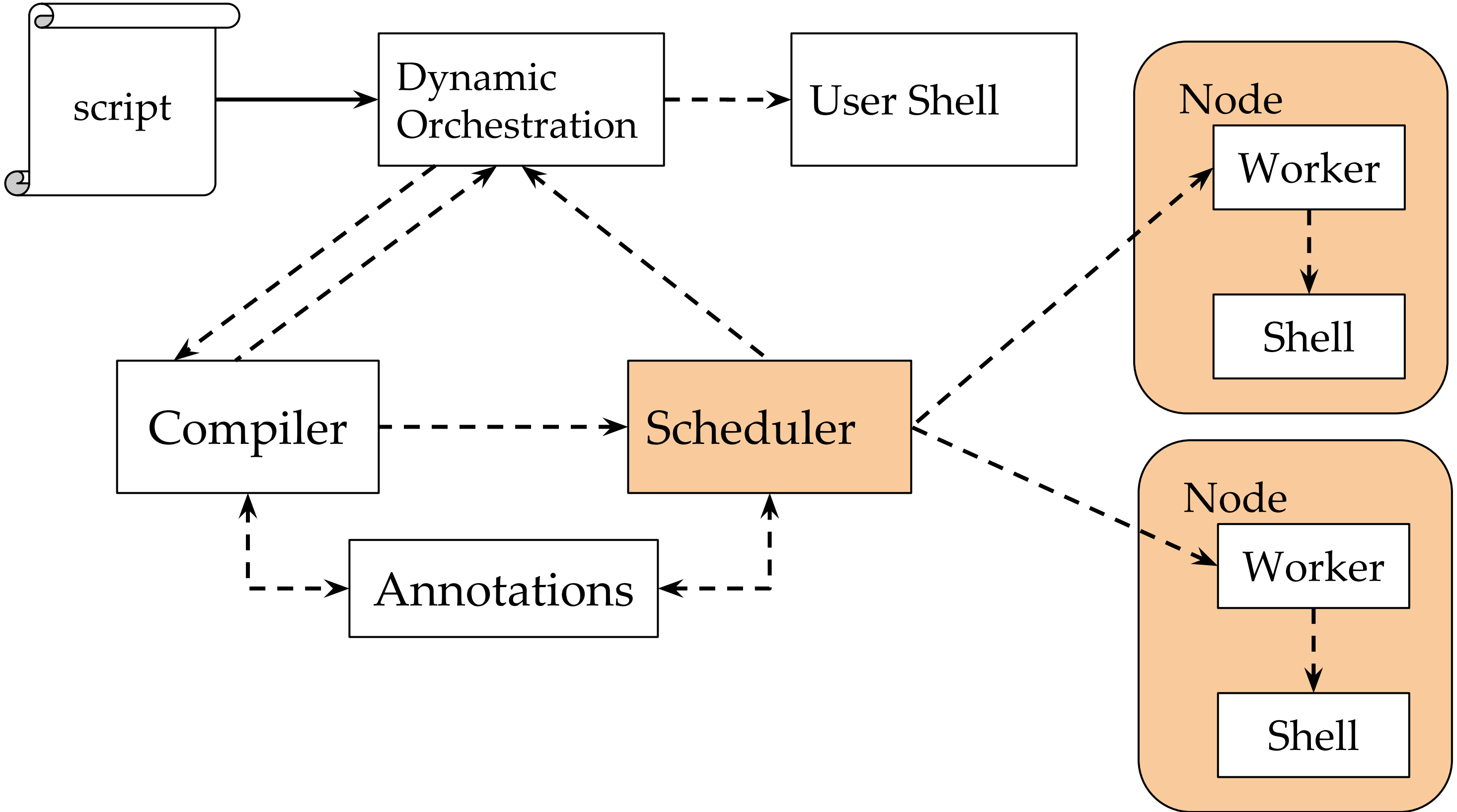


An Order-aware Dataflow Model for Parallel Unix Pipelines
Shivam Handa*, Konstantinos Kallas*, Nikos Vasilakis*, Martin Rinard. 26th ACM SIGPLAN International Conference on Functional Programming (ICFP21)

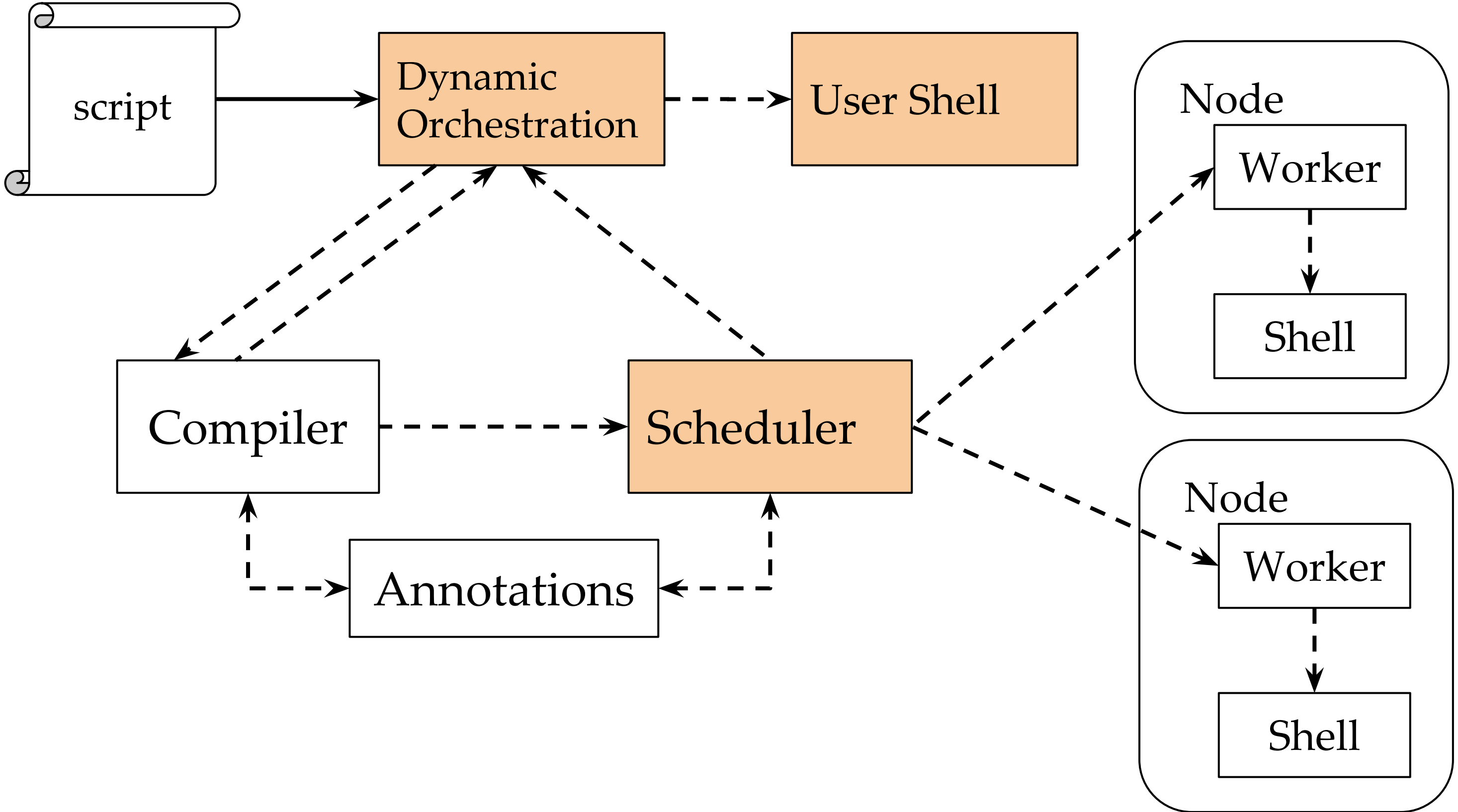
DiSh Overview



DiSh Overview



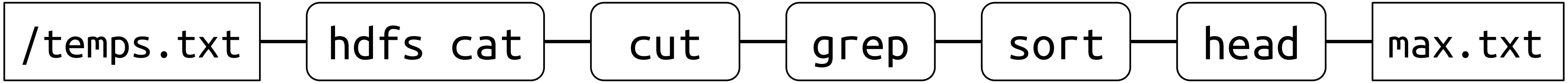
DiSh Overview



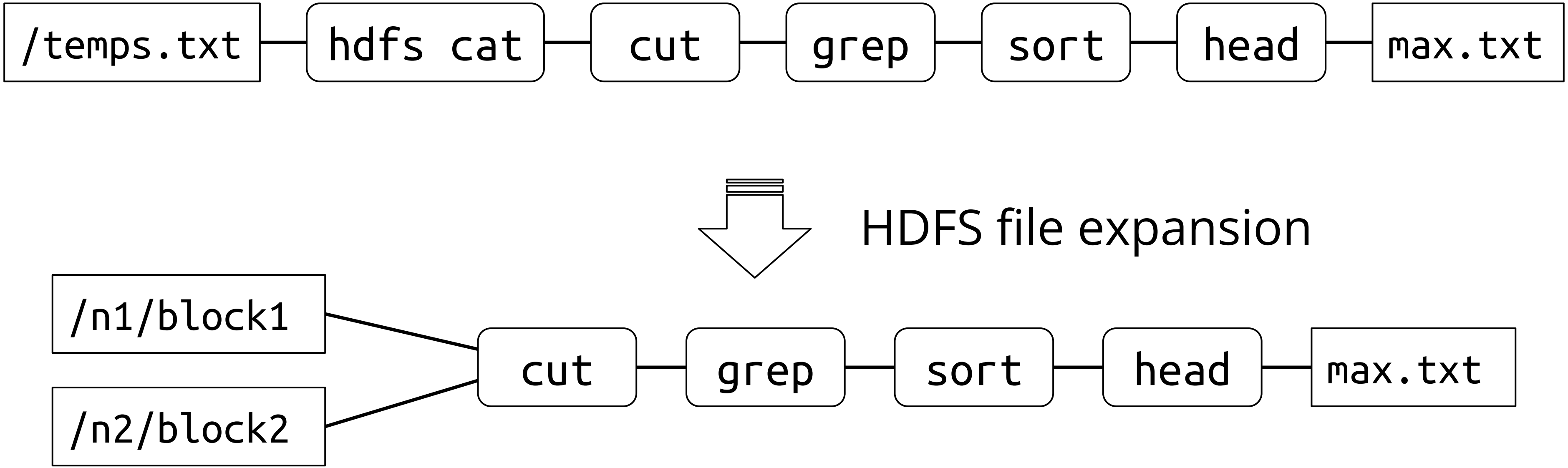
DISH in Action - Temperature Analysis

```
TEMPS="temps.txt"

HDFS dfs -cat $TEMPS | cut -c 89-92 | grep -v 999 | sort -rn | head -n1 > max.txt
```

Data flow transformation



DISH in Action - Temperature Analysis



/temps.txt — hdfs cat — cut — grep — sort — head — max.txt

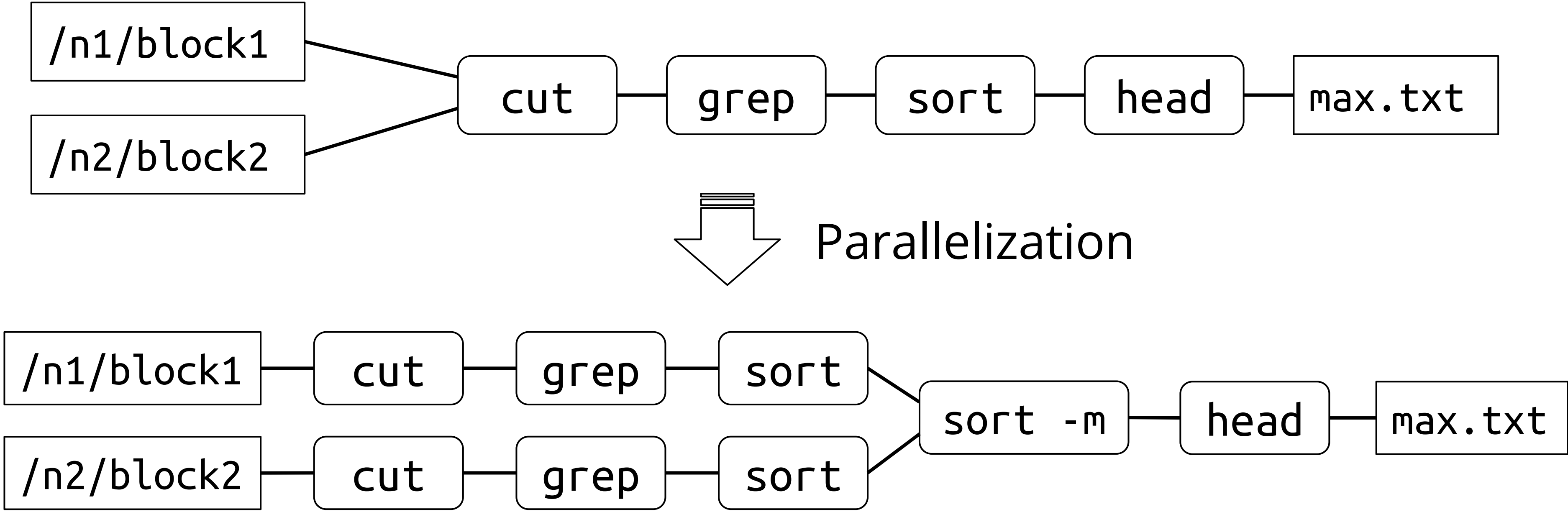
HDFS file expansion

/n1/block1

/n2/block2

cut — grep — sort — head — max.txt

DISH in Action - Temperature Analysis



/n1/block1

/n2/block2

cut

grep

sort

head

max.txt

Parallelization

/n1/block1

cut

grep

sort

/n2/block2

cut

grep

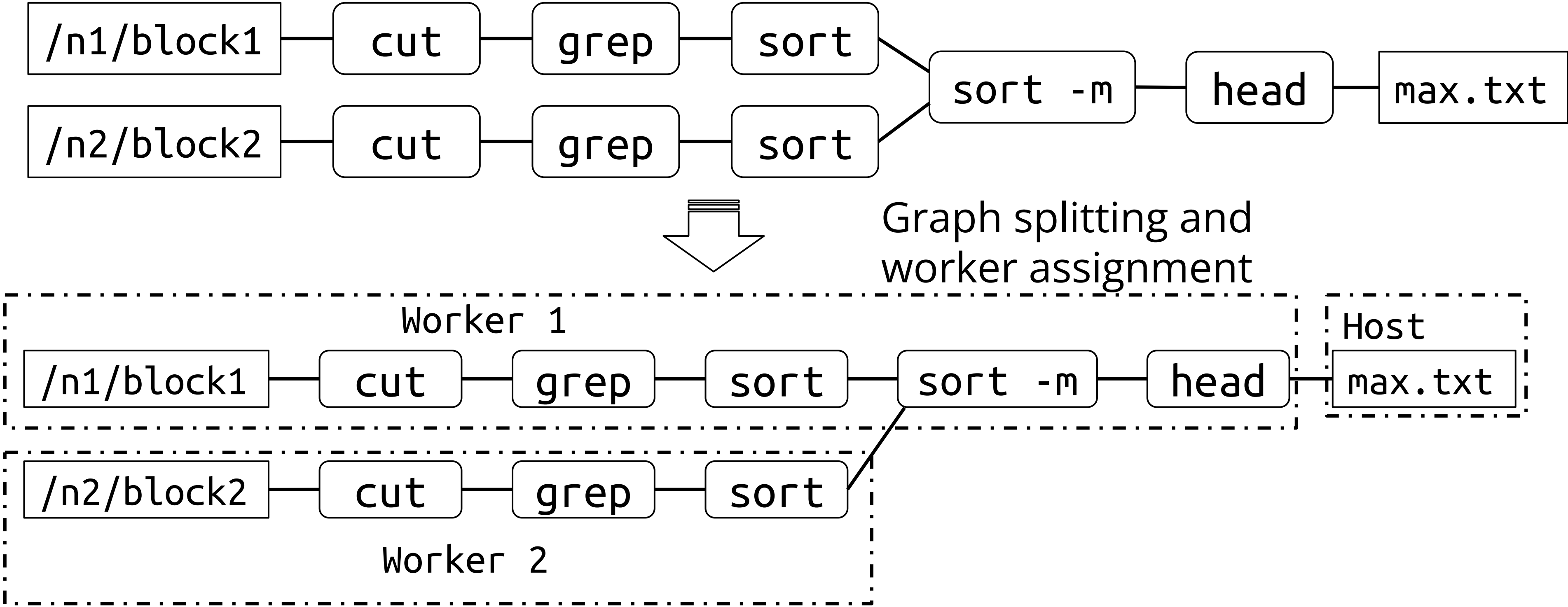
sort

sort -m

head

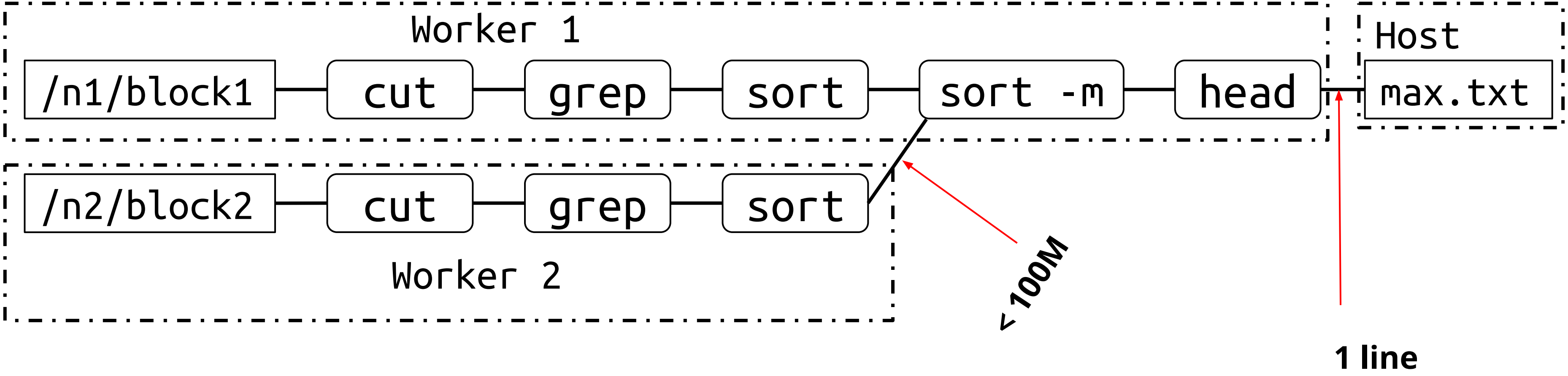
max.txt

DISH in Action - Temperature Analysis



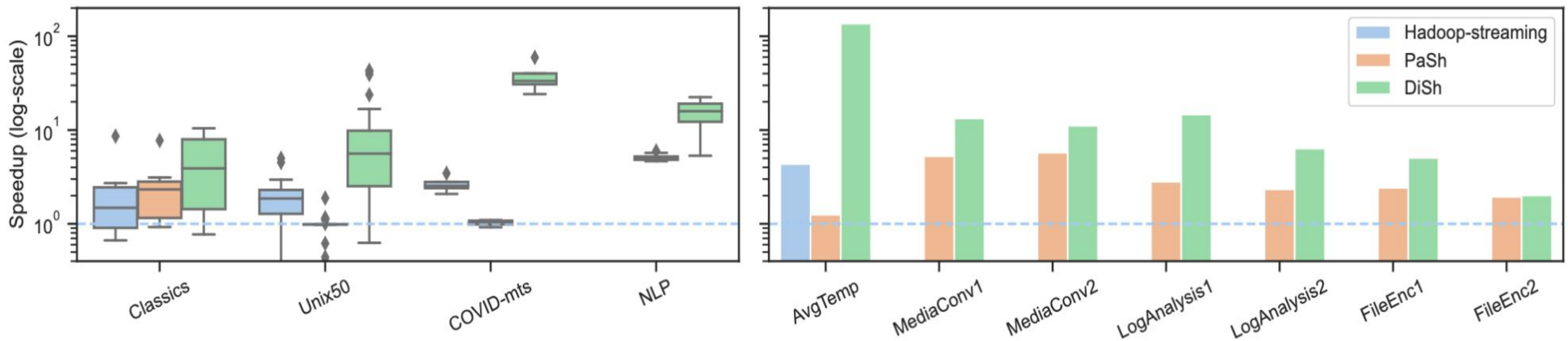
Graph splitting and worker assignment

Data movement



Performance Results 🚀

Higher is better



- Hadoop Streaming: 7.2x avg speedup
- DiSh: 13.6x avg speedup

Not only fast but also correct!

Out of the 408 tests:

- DiSh and Bash only differ in 2 tests.
- Both return with an error, though different code

The POSIX test suite

	Bash and X differ
dash	20
ksh	22
mksh	29
yash	20

What DiSh doesn’t do

DiSh..

DiSh 🍽️ Offers

- A system that automatically distributes shell scripts
- Significant performance gains without developer effort
- Fully compatible with existing shells scripts