Programming MapReduce in Mathematica

Paul-Jean Letourneau
Data Scientist, Wolfram Research

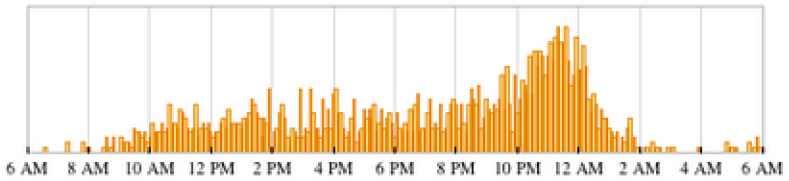
Commercial Users of Functional Programming
Sept 22, 2013

per

These time series plots show my emailing behavior on timescales of years, but we can also look at the distribution of emails sent by time of day. Here's the daily distribution for my sent mail:

In[20]:=
```
dailydistribution[sentdates]
```

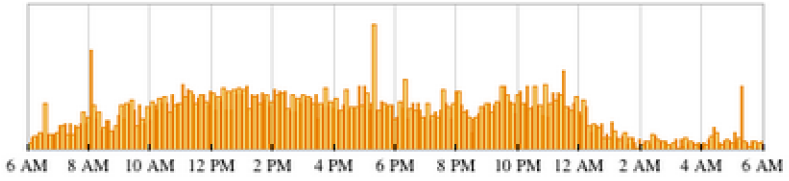
Out[20]=



It looks like I send the majority of emails between 10pm and midnight, which makes sense because I mainly use Gmail for personal stuff in the evenings. The daily distribution of incoming mail is a lot flatter:

In[21]:=
```
dailydistribution[incomingdates]
```

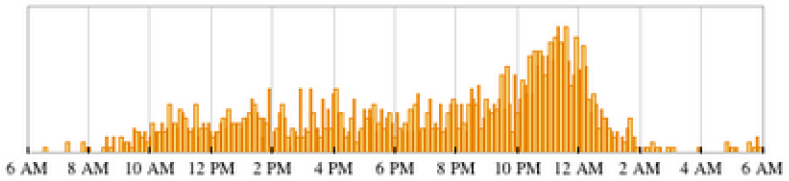
Out[21]=



These time series plots show my emailing behavior on timescales of years, but we can also look at the distribution of emails sent by time of day. Here's the daily distribution for my sent mail:

In[20]:=
```
dailydistribution[sentdates]
```

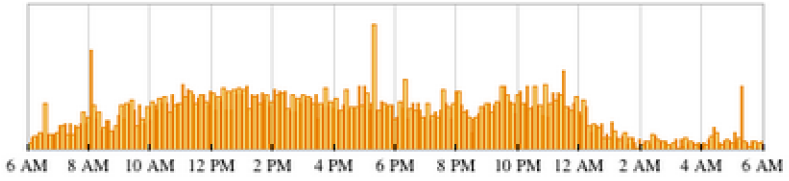
Out[20]=



It looks like I send the majority of emails between 10pm and midnight, which makes sense because I mainly use Gmail for personal stuff in the evenings. The daily distribution of incoming mail is a lot flatter:

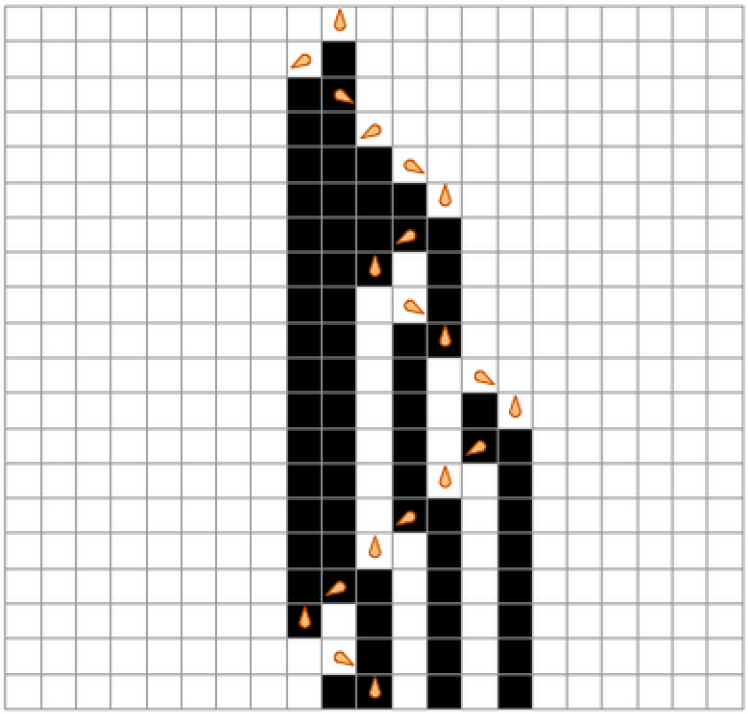
In[21]:=
```
dailydistribution[incomingdates]
```

Out[21]=

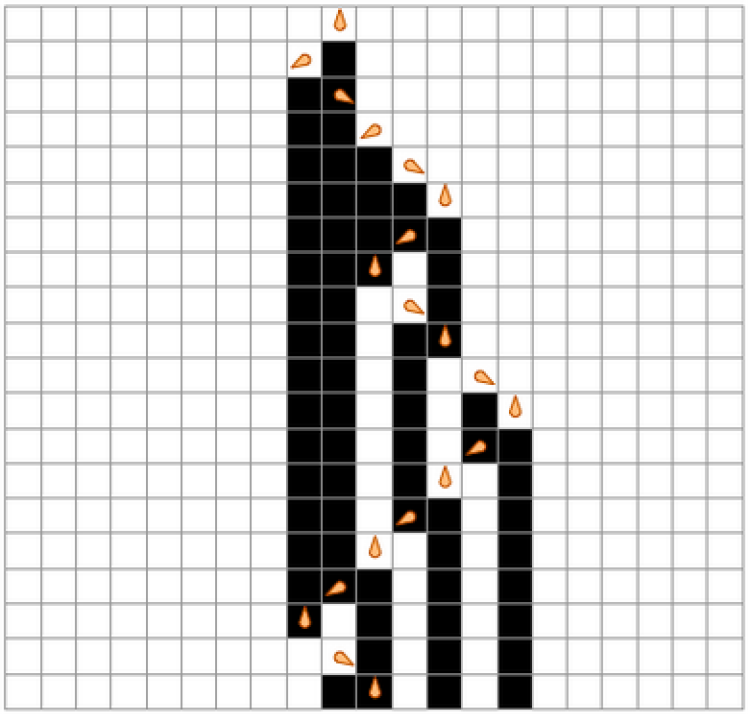


exp

Out[26]=



Out[26]=



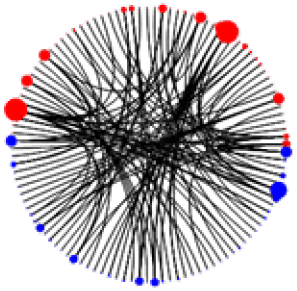
bioinformatics

NEW IN Wolfram Mathematica 7

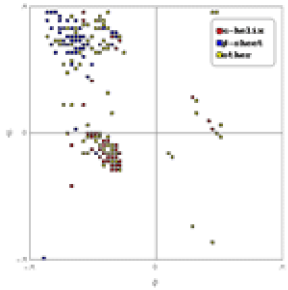
More about *Mathematica* »

< previous | all | next >

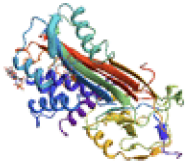
Integrated Genomic & Protein Data



Invent New Gene Network Visualizations



Reproduce Classic Protein Structure Plots



Visualize Protein Folds Using Built-in Protein Structure Data

genomics

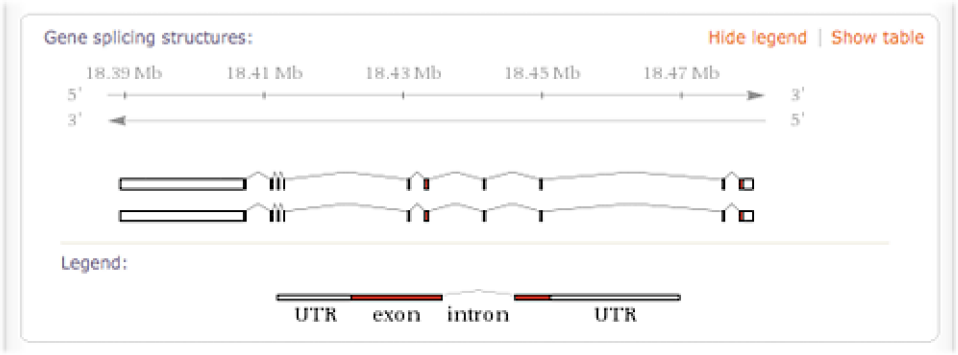
DID YOU KNOW THAT WOLFRAM|ALPHA KNOWS YOUR DNA?

March 10, 2010

8 Comments Tweet 0 Share 0 +1 0 Share 174

Posted by
Paul-Jean Letourneau

The "Gene splicing structures" pod shows the various patterns of introns and exons in the gene's DNA sequence:



distributed computation

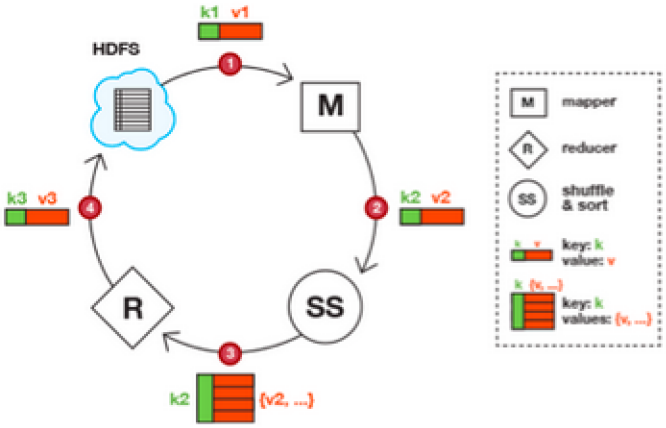
WOLFRAM BLOG

Mathematica Gets Big Data with HadoopLink

July 31, 2013 — Paul-Jean Letourneau, Senior Data Scientist, Wolfram Research

2 Comments Tweet 247 Share 476 ShareThis 907

Here's a diagram that shows how the mapper and reducer exchange key-value pairs:



overview

- core principles of *Mathematica*
- examples
- programming MapReduce with *Mathematica*

the fundamental principles

1. everything is an expression
2. expressions are transformed until they stop changing
3. transformation rules are patterns

1. everything is an expression

expressions are data structures

Mathematica expression:

```
head [ arg1, arg2, ...]
```

LISP expr:

```
(head arg1 arg2 ...)
```

1. everything is an expression

FullForm

```
1 + 1
```

```
2
```

```
FullForm[Unevaluated[1 + 1]]
```

```
Unevaluated[Plus[1, 1]]
```

```
FullForm[Unevaluated[1 + 1 - 3 a]]
```

```
Unevaluated[Plus[1, 1, Times[-1, Times[3, a]]]]
```

1. everything is an expression

... with lots of syntactic sugar

```
#1 + 1 & /@ Range[10]
```

```
{2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
```

```
FullForm[Unevaluated[#1 + 1 & /@ Range[10]]]
```

```
Unevaluated[Map[Function[Plus[Slot[1], 1]], Range[10]]]
```

2. expressions are transformed until they stop changing

definitions are rules

```
Clear[a];

a = 1;

a
```

1

2. expressions are transformed until they stop changing

rules transform expressions: infinite evaluation

```
OwnValues[a]
```

```
{HoldPattern[a] :> 1}
```

```
a // Trace
```

```
{a, 1}
```

```
Clear[b];
a = 1;
```

```
a + b + 1 // Trace
```

```
{{a, 1}, 1 + b + 1, 2 + b}
```

```
b = 2;
```

```
a + b + 1 // Trace
```

```
{{a, 1}, {b, 2}, 1 + 2 + 1, 4}
```

3. rules are patterns

rules have patterns

```
a = 1;
```

```
OwnValues[a]
```

```
{HoldPattern[a] :→ 1}
```

3. rules are patterns

functions are rules

```
Clear[f, g, a, b];
```

```
f[x_Integer] := x + 1
```

```
DownValues[f] // Column
```

```
HoldPattern[f[x_Integer]] :↦ x + 1
```

```
Head[1]
```

```
Integer
```

```
f[1]
```

```
2
```

```
f["a"]
```

```
f[a]
```

```
Head["a"]
```

```
String
```

3. rules are patterns

ordering of rules

```
f[1] := 1000
```

```
DownValues[f] // Column
```

```
HoldPattern[f[1]] :↦ 1000
HoldPattern[f[x_Integer]] :↦ x + 1
```

```
f /@ {0, 1, 2, 3, 4, 5}
```

```
{1, 1000, 3, 4, 5, 6}
```

program as data

expressions are immutable

```
10 = 1
```

```
Set::setraw : Cannot assign to raw object 10. ≫
1
```

```
Plus[1, 1] = 3
```

```
Set::write : Tag Plus in 1 + 1 is Protected. ≫
3
```

```
a = 10
```

```
10
```

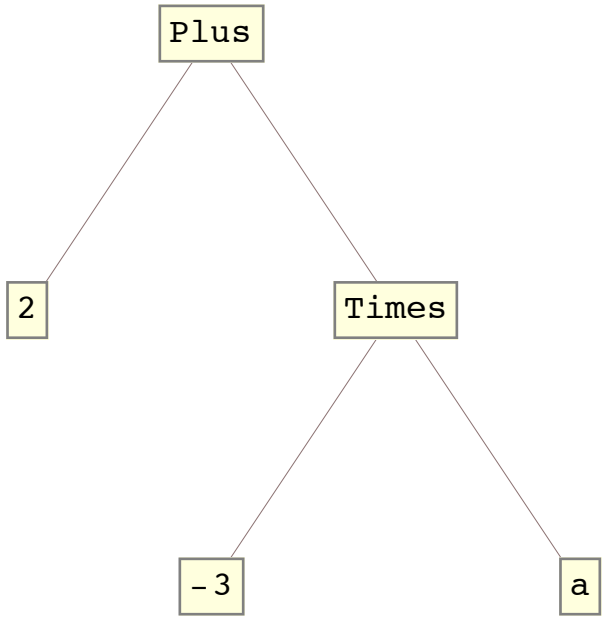
```
a = 1
```

```
1
```

program as data

homoiconicity: expressions ARE the data structure

```
Clear[a];
TreeForm[Unevaluated[1 + 1 - 3 a]]
```



examples

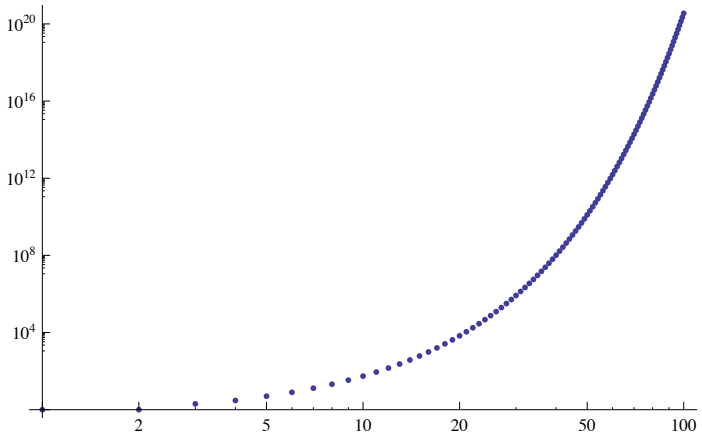
Fibonacci sequence

```
fib[n_] := fib[n] = fib[n - 2] + fib[n - 1];
fib[1] = 1;
fib[2] = 1;
```

```
Table[fib[n], {n, 1, 10}]
```

```
{1, 1, 2, 3, 5, 8, 13, 21, 34, 55}
```

```
ListLogLogPlot[Table[fib[n], {n, 1, 100}]]
```



examples

scrape a web page

```
Grid@Partition[Show[Import@#, ImageSize → 50] & /@ Union@
    Flatten@Table[Cases[Import["http://cufp.org/conference/sessions/2013?page=" <> IntegerString@n, "XMLObject"],
       s_String /; StringMatchQ[s, RegularExpression[".*\\.jpg"]], Infinity], {n, 0, 3}], 5, 5, 1, {}]
```

examples

"everything is a one-liner in *Mathematica* ... for a sufficiently long line." (Theo Gray)

```
Show[ImageAssemble[
  Round[Rescale[ImageData[i = Nest[Darker, ImageResize[ExampleData[{"TestImage", "Elaine"}], 50], 3]]] 9] /.
   n_Integer :> Nest[Lighter, i, n]], ImageSize -> 400]
```

gateway drug ...

... to declaritive programming

```
y = 0;
For[i = 1, i ≤ 10, i++,
  y += i^2
 ];
y
```

```
385
```

```
Fold[#1 + #2^2 &, 0, Range[10]]
```

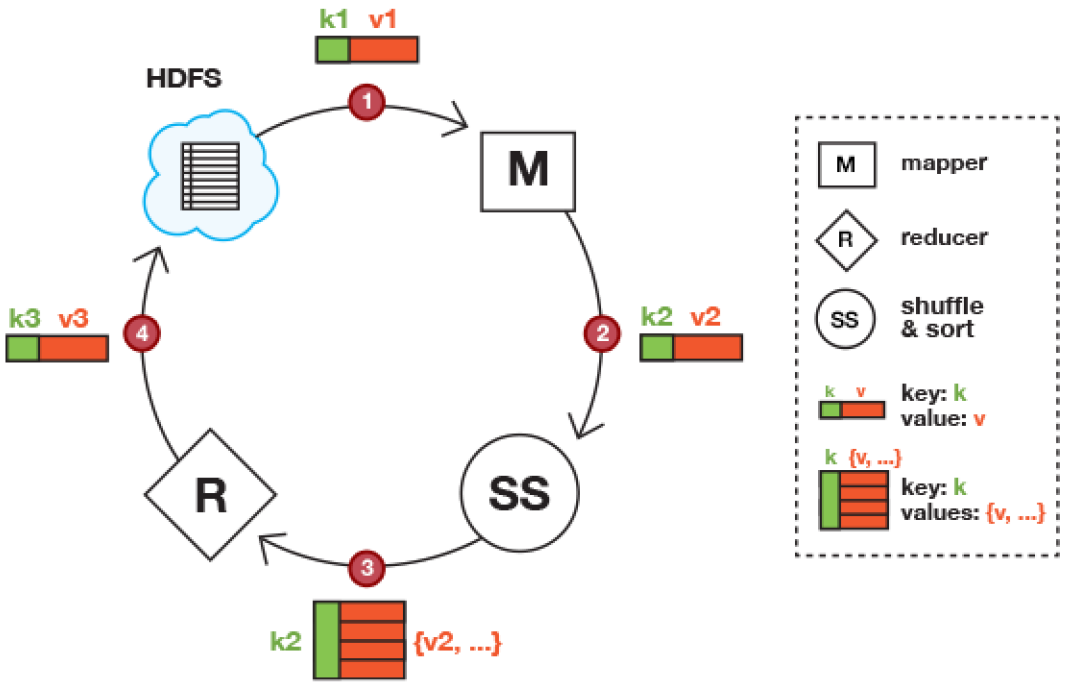
```
385
```

advanced topics

- scoping
- evaluation control
- *MathLink* protocol

MapReduce

MapReduce in a nutshell



HadoopLink

WordCount

```
textRaw = Import["http://www.gutenberg.org/cache/epub/1342/pg1342.txt"];
```

```
StringTake[textRaw, 200]
```

```
The Project Gutenberg EBook of Pride and Prejudice, by Jane Austen

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away o
```

```
Reverse@SortBy[Tally[StringSplit[textRaw, RegularExpression["[\\W_]+"]]], Last] // Short
```

```
{{the, 4218}, {to, 4187}, {of, 3705}, <<7101>>, {10, 1}, {000, 1}}
```

HadoopLink

create key-value pairs

```
paras = StringSplit[textRaw, RegularExpression["\n{2,}"]];

paraPairs = Transpose[{paras, Table[1, {Length@paras}]}];

Grid[{#}, Frame → All, Background → {{LightGreen, LightRed}}] & /@ paraPairs[[1 ;; 4]] // Column
```

The Project Gutenberg EBook of Pride and Prejudice, by Jane Austen	1
This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org	1
Title: Pride and Prejudice	1
Author: Jane Austen	1

HadoopLink

export to the Hadoop filesystem

```
<< HadoopLink
```

```
$$link = OpenHadoopLink[
    "fs.default.name" → "hdfs://hadoopheadlx.wolfram.com:8020",
    "mapred.job.tracker" → "hadoopheadlx.wolfram.com:8021"
  ];
```

```
inputfile["pap"] = "/user/paul-jean/hadooplink/pap-paras.seq";
```

```
DFSExport[$$link, inputfile["pap"], paraPairs, "SequenceFile"]
```

```
/user/paul-jean/hadooplink/pap-paras.seq
```

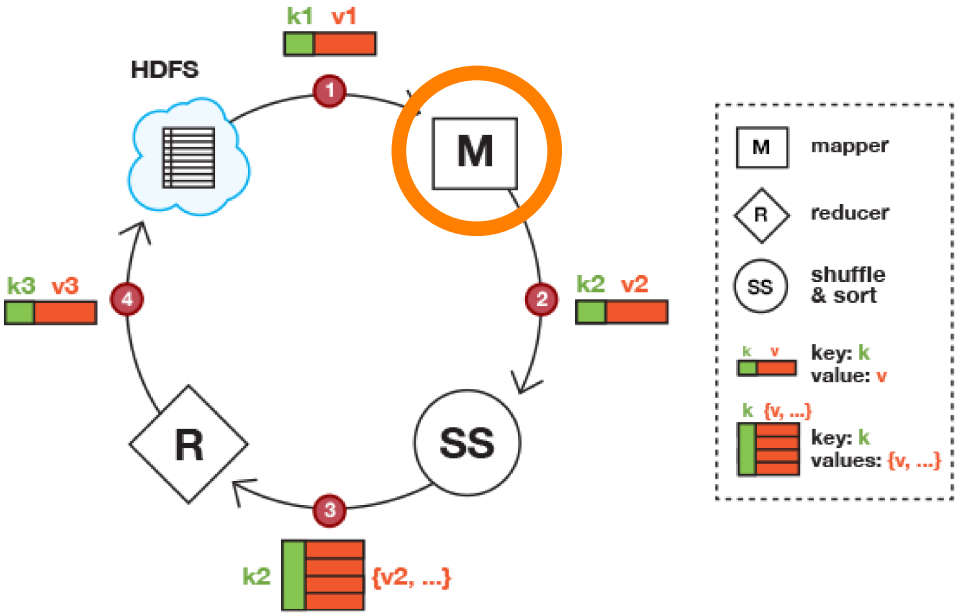
```
Grid[Partition[Names["HadoopLink`*"], 4], Alignment → Left, BaseStyle → {FontSize → 14}]
```

DFSAbsoluteFileName	DFSCloseSequenceStream	DFSCopyDirectory	DFSCopyFile
DFSCopyFromLocal	DFSCopyToLocal	DFSCreateDirectory	DFSDeleteDirectory
DFSDeleteFile	DFSDirectoryQ	DFSExport	DFSFileByteCount
DFSFileDate	DFSFileExistsQ	DFSFileNames	DFSFileQ
DFSFileType	DFSImport	DFSOpenSequenceStream	DFSReadList
DFSRenameDirectory	DFSRenameFile	DFSSequenceStream	HadoopLink
HadoopMapReduceJob	IncrementCounter	OpenHadoopLink	Yield

HadoopLink

mapper

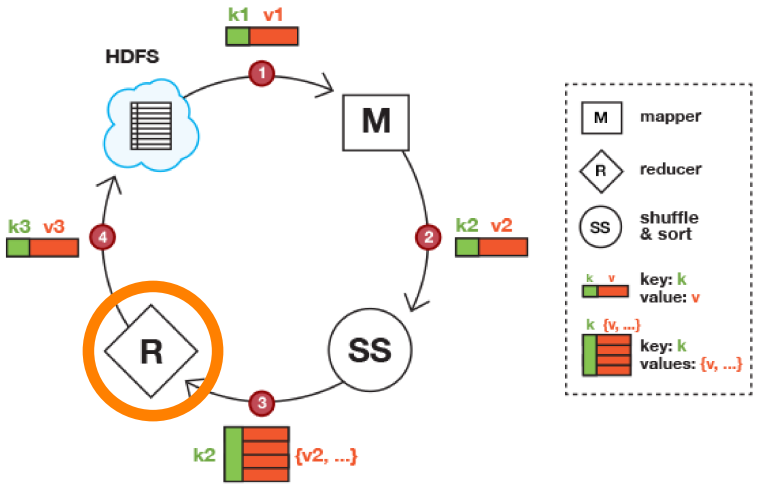
```
WordCountMapper = Function[{k, v},
    With[{
      words = ToLowerCase /@ StringSplit[k, RegularExpression["[\\W_]+"]]},
     Yield[#, 1] & /@ words
    ]
   ];
```



HadoopLink

reducer

```
SumReducer = Function[{k, vs},
   Module[
    {sum = 0},
    While[vs@hasNext[],
     sum += vs@next[]
    ];
    Yield[k, sum]
   ]
  ];
```

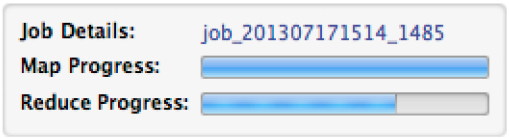


HadoopLink

run the job

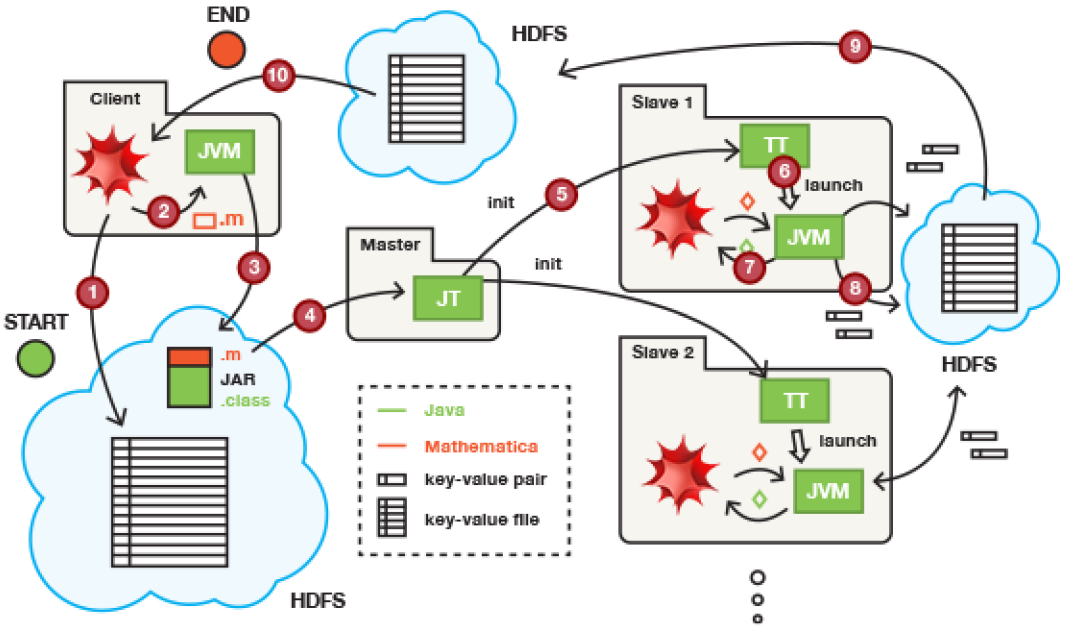
```
inputfile["pap"] = "/user/paul-jean/hadooplink/pap-paras.seq";

outputdir["pap"] = "/user/paul-jean/hadooplink/pap-wordcount";

HadoopMapReduceJob[
 $$link,
 "pap wordcount",
 inputfile["pap"],
 outputdir["pap"],
 WordCountMapper,
 SumReducer
]
```

Out[21]=



HadoopLink

control flow



genome search engine

prep data

```
mtseq = GenomeData[{"Mitochondrion", {1, -1}}];
```

```
StringTake[mtseq, 30]
```

```
GATCACAGGTCTATCACCCTATTAACCACT
```

```
querybases = "GCACACACACA";
```

```
StringPosition[mtseq, querybases]
```

```
{{515, 525}}
```

genome search engine

create key-value pairs

```
mtchars = Characters[mtseq];
mtbases = Transpose[{mtchars, Range@Length@mtchars}];
Grid[{#}, Frame → All, Background → {{LightGreen, LightRed}}] & /@ mtbases[[1 ;; 20]]
```

G	1	A	2	T	3	C	4	A	5
C	6	A	7	G	8	G	9	T	10
C	11	T	12	A	13	T	14	C	15
A	16	C	17	C	18	C	19	T	20

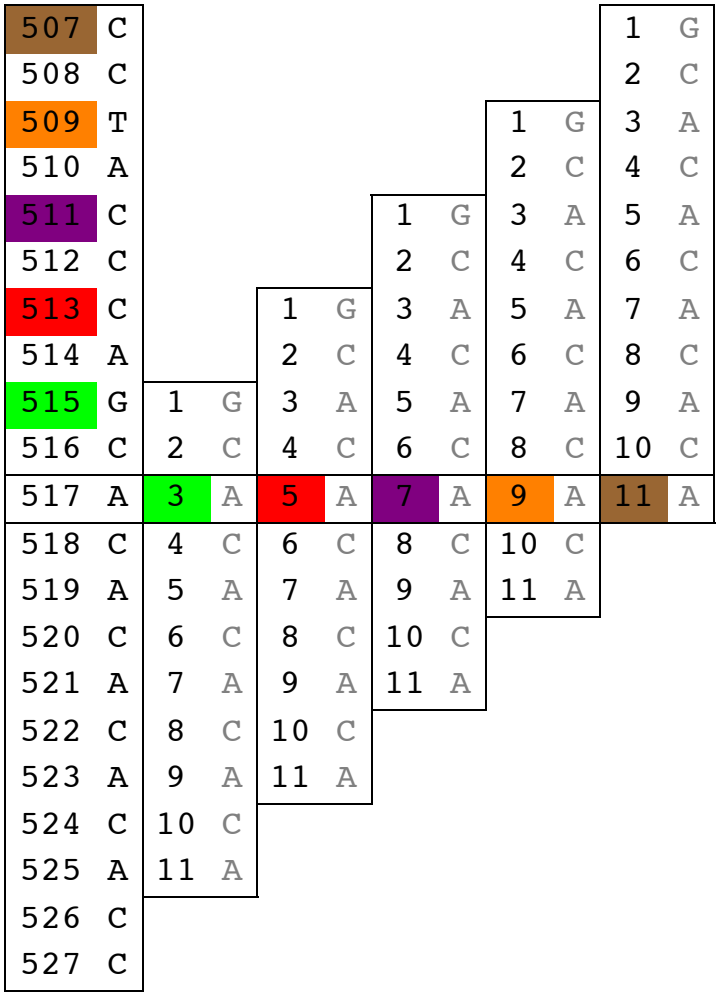
genome search engine

mapper

```
querybases = "GCACACACACA";

GenomeSearchMapper[qchunks : {__String}] :=
 Function[{base, genomepos},
  Module[{pos, querypositions},
   querypositions = Flatten@Position[qchunks, base];
   With[
       {querypos = #},
       Yield[genomepos - (querypos - 1), querypos]
      ] & /@ querypositions
  ]
 ]
```

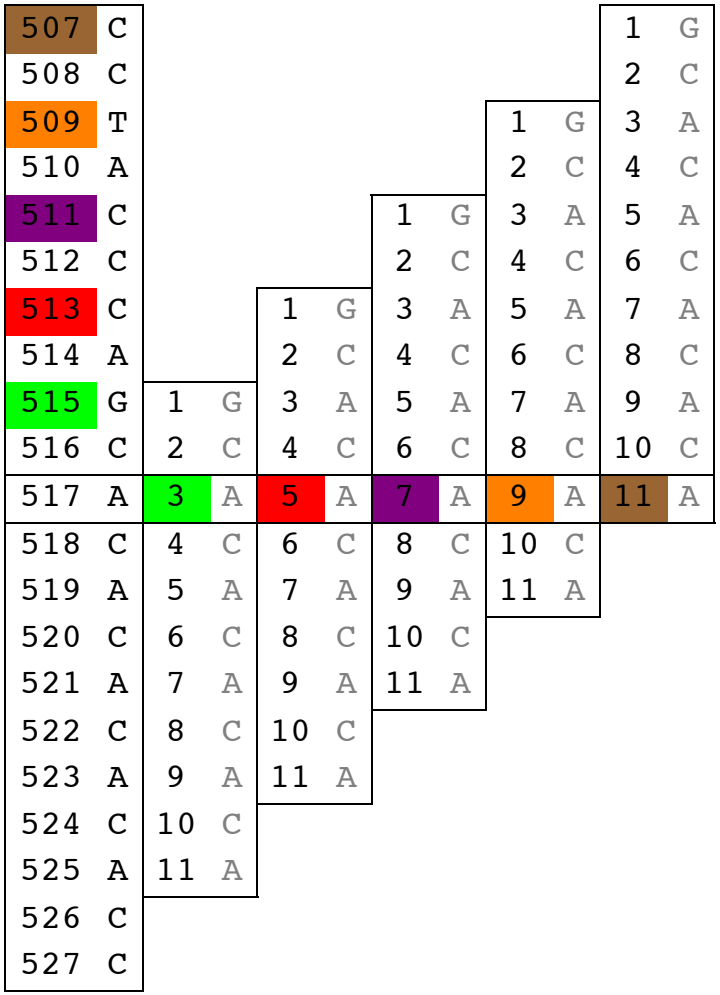
genome search engine

mapper



genome search engine

mapper



genome search engine

reducer

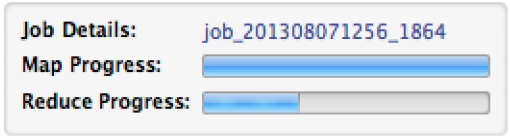
```
GenomeSearchReducer[qchunks : {__String}] :=
 Function[{matchposition, chunkoffsets},
  Module[{numchunks, sumoffsets, goalsum},
   numchunks = Length@qchunks;
   sumoffsets = 0;
   goalsum = numchunks * (numchunks + 1) / 2;
   While[chunkoffsets@hasNext[],
    sumoffsets += chunkoffsets@next[];
   ];
   If[sumoffsets == goalsum,
    Yield[StringJoin@qchunks, matchposition]
   ]
  ]
 ]
```

genome search engine

run the job

```
querybases = "GCACACACACA";
input = DFSFileNames[$$link, "mt-bases.index", "hadooplink"];
out = "/user/paul-jean/hadooplink/mt-search-GCACACACACA";

HadoopMapReduceJob[
 $$link,
 "mt search GCACACACACA",
 input,
 out,
 GenomeSearchMapper[querybases],
 GenomeSearchReducer[querybases]
]
```

Out[36]=



genome search engine

import the results

```
files = DFSFileNames[$$link, "part-*", "/user/paul-jean/hadooplink/mt-search-GCACACACACA-bases.out"]
```

```
Join @@ (DFSImport[$$link, #, "SequenceFile"] & /@ files)
```

```
{{GCACACACACA, 515}}
```

```
First /@ StringPosition[mtseq, querybases]
```

```
{515}
```

challenges

memory consumption

Hadoop job_201308211825_0001 failures on hadoopheadlx

Attempt	Task	Machine	State	Error	Logs
attempt_201308211825_0001_m_000001_0	task_201308211825_0001_m_000001	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000001_1	task_201308211825_0001_m_000001	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000001_2	task_201308211825_0001_m_000001	hadoop5lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_0	task_201308211825_0001_m_000002	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_1	task_201308211825_0001_m_000002	hadoop3lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_2	task_201308211825_0001_m_000002	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_0	task_201308211825_0001_m_000004	hadoop4lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_1	task_201308211825_0001_m_000004	hadoop6lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_2	task_201308211825_0001_m_000004	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_3	task_201308211825_0001_m_000004	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_0	task_201308211825_0001_m_000005	hadoop5lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_1	task_201308211825_0001_m_000005	hadoop4lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_2	task_201308211825_0001_m_000005	hadoop3lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All

Go back to JobTracker

challenges

memory consumption

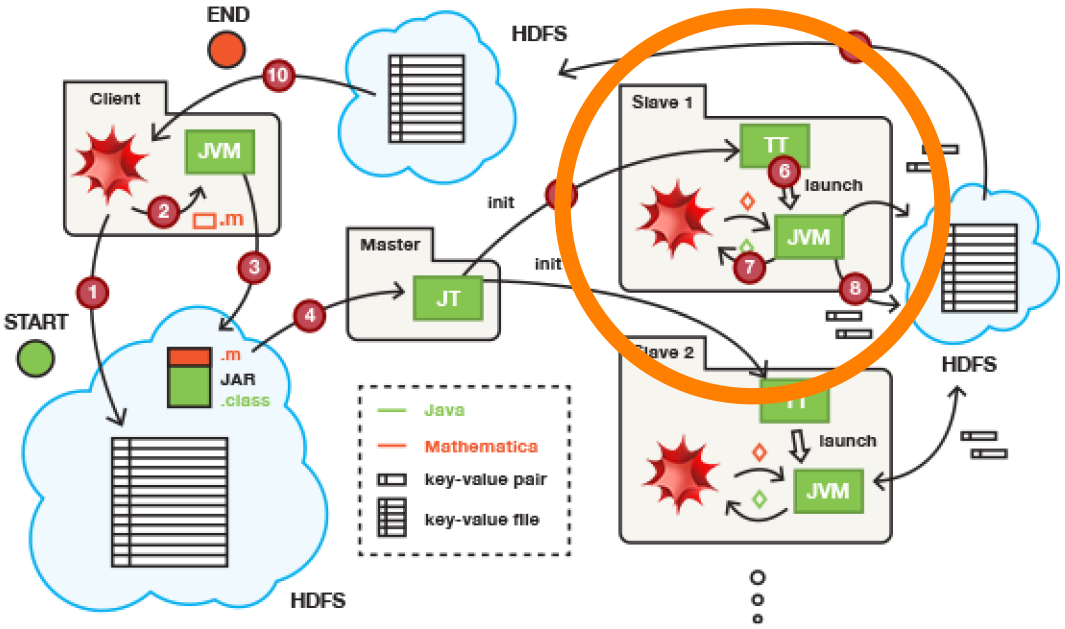
Hadoop job_201308211825_0001 failures on hadoopheadlx

Attempt	Task	Machine	State	Error	Logs
attempt_201308211825_0001_m_000001_0	task_201308211825_0001_m_000001	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000001_1	task_201308211825_0001_m_000001	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000001_2	task_201308211825_0001_m_000001	hadoop5lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_0	task_201308211825_0001_m_000002	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_1	task_201308211825_0001_m_000002	hadoop3lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000002_2	task_201308211825_0001_m_000002	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_0	task_201308211825_0001_m_000004	hadoop4lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_1	task_201308211825_0001_m_000004	hadoop6lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_2	task_201308211825_0001_m_000004	hadoop7lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000004_3	task_201308211825_0001_m_000004	hadoop8lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_0	task_201308211825_0001_m_000005	hadoop5lx.wolfram.com	FAILED	java.lang.Throwable: Child Error at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:242) Caused by: java.io.IOException: Task process exit with nonzero status of 1. at org.apache.hadoop.mapred.TaskRunner.run(TaskRunner.java:229)	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_1	task_201308211825_0001_m_000005	hadoop4lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All
attempt_201308211825_0001_m_000005_2	task_201308211825_0001_m_000005	hadoop3lx.wolfram.com	FAILED	Error: GC overhead limit exceeded	Last 4KB Last 8KB All

Go back to JobTracker

challenges

HadoopLink architecture



challenges

job-level configurations

```
HadoopMapReduceJob[
 $$link,
 "hs search GCACACACACA",
 input,
 output,
 GenomeSearchMapper[querybases],
 GenomeSearchReducer[querybases],
 "mapred.child.java.opts" -> "-Xmx512m"
]
```

conclusions

core principles of *Mathematica*

- everything is an expression
- expressions are transformed until they stop changing
- transformation rules are patterns

examples

- Fibonacci sequence, web scraping, recursive image

MapReduce with *Mathematica*

- mapper and reducer functions
- running MapReduce jobs using HadoopLink
- challenges: constrain memory consumption, job-level configurations

the end

@rule146

```
rl = MapThread[Rule, {Tuples[{1, 0}, 3], IntegerDigits[146, 2, 8]}];
ar = NestList[Partition[#, 3, 1, 2] /. rl &, RandomInteger[1, 200], 150];
gr = ArrayPlot[ar, PixelConstrained → 2]
```