Programming MapReduce in Mathematica

Paul-Jean Letourneau
Data Scientist, Wolfram Research

Commercial Users of Functional Programming
Sept 22, 2013
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:

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

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:

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

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]
```
bioinformatics

NEW IN Wolfram Mathematica 7

Integrated Genomic & Protein Data

Invent New Gene Network Visualizations
Reproduce Classic Protein Structure Plots
Visualize Protein Folds Using Built-in Protein Structure Data
The “Gene splicing structures” pod shows the various patterns of introns and exons in the gene’s DNA sequence:

Legend:
- UTR
- exon
- intron
- UTR
distributed computation

*Wolfram Blog*

*Mathematica* Gets Big Data with HadoopLink

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

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

core principles of *Mathematica*

eamples

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:

```lisp
(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 & /@ Range[10]
{2, 3, 4, 5, 6, 7, 8, 9, 10, 11}

FullForm[Unevaluated[# + 1 & /@ Range[10]]]
Unevaluated[Map[Function[Plus[Slot[1], 1]], Range[10]]]
```
2. expressions are transformed until they stop changing

definitions are rules

```math
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

```math
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

\[
\text{DownValues}[f] // \text{Column}
\]

\[
\text{HoldPattern}[f[1]] \rightarrow 1000
\]

\[
\text{HoldPattern}[f[x\_Integer]] \rightarrow x + 1
\]

\[
f @\{0, 1, 2, 3, 4, 5\}
\]

\[
\{1, 1000, 3, 4, 5, 6\}
\]
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

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

![Tree representation of the expression](image)
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[#1, 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)

```mathematica
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

```mathematica
y = 0;
For[i = 1, i <= 10, i++,
    y += i^2
]
y
385

Fold[#+#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
```

<table>
<thead>
<tr>
<th>The Project Gutenberg EBook of Pride and Prejudice, by Jane Austen</th>
</tr>
</thead>
<tbody>
<tr>
<td>This eBook is for the use of anyone anywhere at no cost and with</td>
</tr>
<tr>
<td>almost no restrictions whatsoever. You may copy it, give it away or</td>
</tr>
<tr>
<td>re-use it under the terms of the Project Gutenberg License included</td>
</tr>
<tr>
<td>with this eBook or online at <a href="http://www.gutenberg.org">www.gutenberg.org</a></td>
</tr>
<tr>
<td>Title: Pride and Prejudice</td>
</tr>
<tr>
<td>Author: Jane Austen</td>
</tr>
</tbody>
</table>
HadoopLink

export to the Hadoop filesystem

<< HadoopLink

$$link = \text{OpenHadoopLink}\{\}
    "fs.default.name" \rightarrow \text{hdfs://hadoopheadlx.wolfram.com:8020},
    "mapred.job.tracker" \rightarrow \text{hadoopheadlx.wolfram.com:8021}\
\};

\text{inputfile["pap"]} = \"/user/paul-jean/hadooplink/pap-paras.seq\";

\text{DFSExport}[$$link, \text{inputfile["pap"], paraPairs, "SequenceFile"]}

\text{/user/paul-jean/hadooplink/pap-paras.seq}

\text{Grid[Partition[Names["HadoopLink\*"], 4], Alignment \rightarrow Left, BaseStyle \rightarrow \{FontSize \rightarrow 14\}]}

\text{DFSAbsoluteFileName} \quad \text{DFSCloseSequenceStream} \quad \text{DFSCopyDirectory} \quad \text{DFSCopyFile}

\text{DFSCopyFromLocal} \quad \text{DFSCopyToLocal} \quad \text{DFSCreateDirectory} \quad \text{DFSDeleteDirectory}

\text{DFSDeleteFile} \quad \text{DFSDirectoryQ} \quad \text{DFSExport} \quad \text{DFSFileByteCount}

\text{DFSFileDate} \quad \text{DFSFileExistsQ} \quad \text{DFSFileNames} \quad \text{DFSFileQ}

\text{DFSFileType} \quad \text{DFSIimport} \quad \text{DFSOpenSequenceStream} \quad \text{DFSReadList}

\text{DFS RenameDirectory} \quad \text{DFS RenameFile} \quad \text{DFS SequenceStream} \quad \text{HadoopLink}

\text{HadoopMapReduceJob} \quad \text{IncrementCounter} \quad \text{OpenHadoopLink} \quad \text{Yield}
HadoopLink

mapper

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

**reducer**

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

**run the job**

```python
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
]
```
HadoopLink

coreflow
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

```plaintext
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

```
<table>
<thead>
<tr>
<th>507</th>
<th>C</th>
<th>1</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>508</td>
<td>C</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td><strong>509</strong></td>
<td>T</td>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>510</td>
<td>A</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>511</td>
<td>C</td>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>512</td>
<td>C</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td><strong>513</strong></td>
<td>C</td>
<td>1</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>514</td>
<td>A</td>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>515</td>
<td>G</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>A</td>
</tr>
<tr>
<td>516</td>
<td>C</td>
<td>6</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>C</td>
</tr>
<tr>
<td>517</td>
<td>A</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>A</td>
</tr>
<tr>
<td>518</td>
<td>C</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>519</td>
<td>A</td>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>520</td>
<td>C</td>
<td>7</td>
<td>A</td>
</tr>
<tr>
<td>521</td>
<td>A</td>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>522</td>
<td>C</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td>523</td>
<td>A</td>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>524</td>
<td>C</td>
<td>11</td>
<td>A</td>
</tr>
<tr>
<td>525</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
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 = "GCACACACACACA";
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]
]
```
genome search engine

import the results

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

Join@@ {DFSImport[$$link, #, "SequenceFile"] & /@files}

First /@ StringPosition[mtseq, querybases]

{515}
challenges

memory consumption

Hadoop job_201308211825_0001 failures on hadoopheadlxl

<table>
<thead>
<tr>
<th>Attempt</th>
<th>Task</th>
<th>Machine</th>
<th>State</th>
<th>Error</th>
<th>Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>attempt_201308211825_0001_m_000001_0</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000001_1</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop7x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000001_2</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop5x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_0</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_1</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop3x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_2</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_0</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop4x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_1</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_2</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop7x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_3</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_0</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop8x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_1</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop4x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_2</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop3x.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td>Last 4KB Last 8KB All</td>
</tr>
</tbody>
</table>
challenges

memory consumption

<table>
<thead>
<tr>
<th>Attempt</th>
<th>Task</th>
<th>Machine</th>
<th>State</th>
<th>Error</th>
<th>Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>attempt_201308211825_0001_m_000001_0</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000001_1</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop7k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000001_2</td>
<td>task_201308211825_0001_m_000001</td>
<td>hadoop5k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_0</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_1</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop3k.wolfram.com</td>
<td>FAILED</td>
<td>Java.lang.Exceptions: Child Error</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000002_2</td>
<td>task_201308211825_0001_m_000002</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_0</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop4k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_1</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Java.lang.Exceptions: Child Error</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_2</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop7k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000004_3</td>
<td>task_201308211825_0001_m_000004</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Java.lang.Exceptions: Child Error</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_0</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop8k.wolfram.com</td>
<td>FAILED</td>
<td>Java.lang.Exceptions: Child Error</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_1</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop4k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
<tr>
<td>attempt_201308211825_0001_m_000005_2</td>
<td>task_201308211825_0001_m_000005</td>
<td>hadoop3k.wolfram.com</td>
<td>FAILED</td>
<td>Error: GC overhead limit exceeded</td>
<td></td>
</tr>
</tbody>
</table>

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]
```