MATLAB EXPO 2016
KOREA

4월 28일 (목)

등록 하기 matlabexpo.co.kr

- Data analytics Workflow
- Map reduce demo refine. (Car register or weather station)

Working with Big Data using MATLAB

성호현 차장

Senior Application Engineer

MathWorks Korea

Challenges of Data

“Any collection of data sets so large and complex that it becomes difficult to process using … traditional data processing applications.”
(Wikipedia)

- Various Data Sources
- Rapid data exploration
- Development of scalable algorithms
- Ease of deployment

Agenda

- How big is big?
- Reading big data
- Processing quite big data
- Processing big data
- Summary

How big is big?

What does “Big Data” even mean?

“Any collection of data sets so large and complex that it becomes difficult to process using … traditional data processing applications.”

(Wikipedia)

“Any collection of data sets so large that it becomes difficult to process using traditional MATLAB functions, which assume all of the data is in memory.”

(MATLAB)

How big is big?

Not a new problem

- In 1085 William 1st commissioned a survey of England
  - ~2 million words and figures collected over two years
  - too big to handle in one piece
  - collected and summarized in several pieces
  - used to generate revenue (tax), but most of the data then sat unused

How big is big?

A new problem

- The Large Hadron Collider was switched back on earlier this year
  - ~600 million collisions per second (only a fraction get recorded)
  - amounts to 30 petabytes per year
  - too big to even store in one place
  - used to explore interesting science, but taking researchers a long time to get through

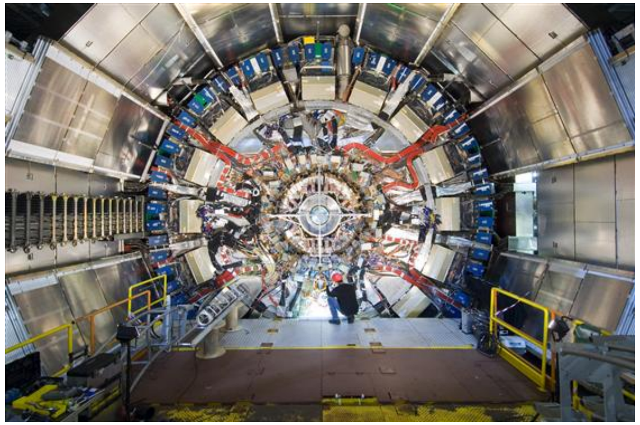
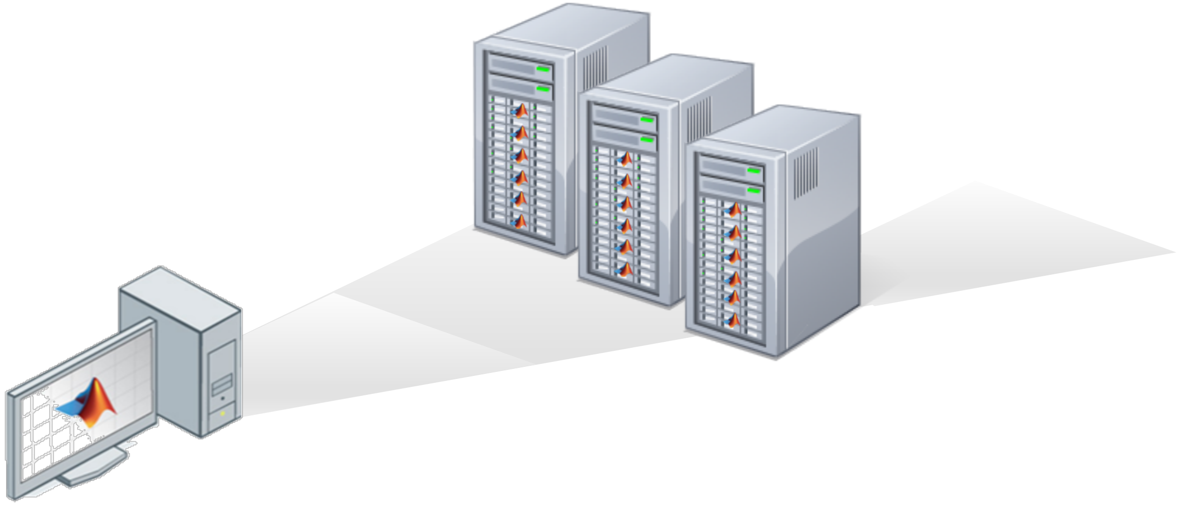


Image courtesy of CERN. Copyright 2011 CERN.

How big is big?

Sizes of data in this talk

- Most of our data lies somewhere in between
  - a few MB up to a few TB
  - <1GB can typically be handled in memory on one machine (small data)
  - 1-100GB can typically be handled in memory of many machines (quite big data)
  - >100GB typically requires processing in pieces using many machines (big data)

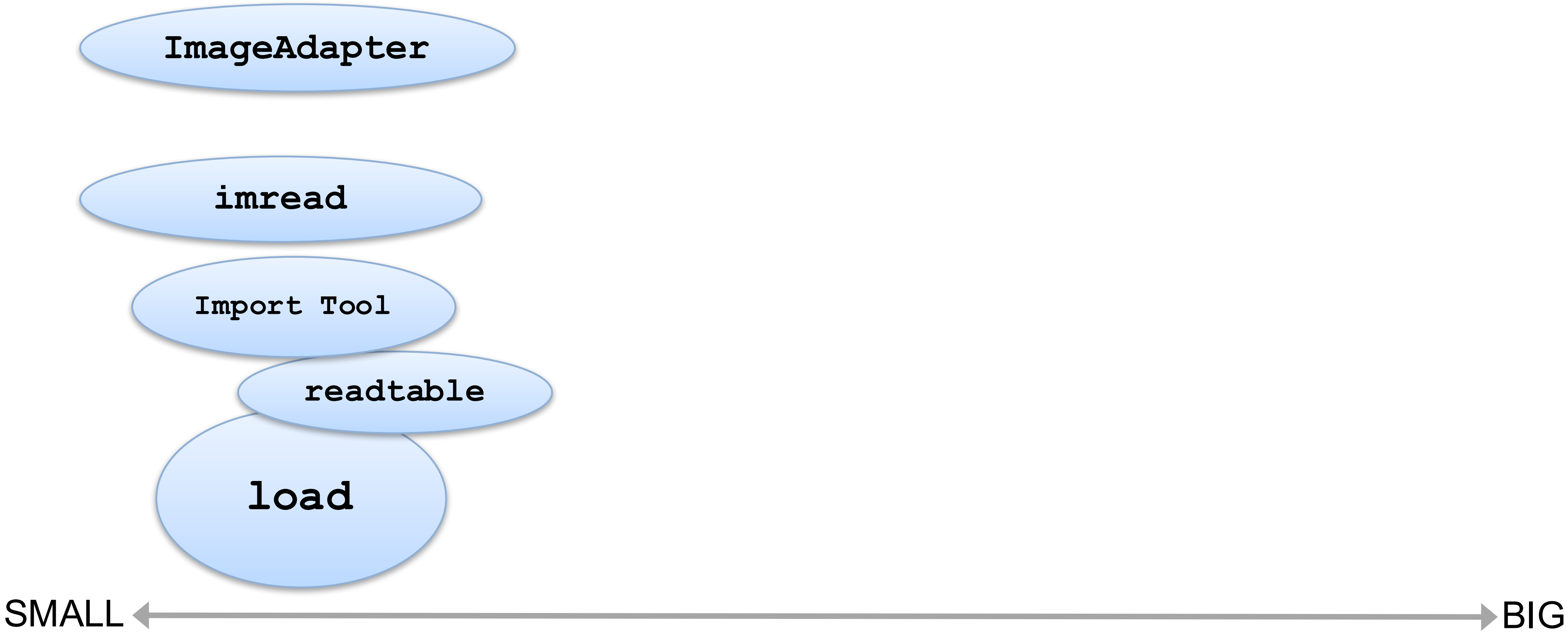


Agenda

- How big is big?
- ➡ Reading big data
- Processing quite big data
- Processing big data
- Summary

Reading big data

What tools are there?



ImageAdapter

imread

Import Tool

readtable

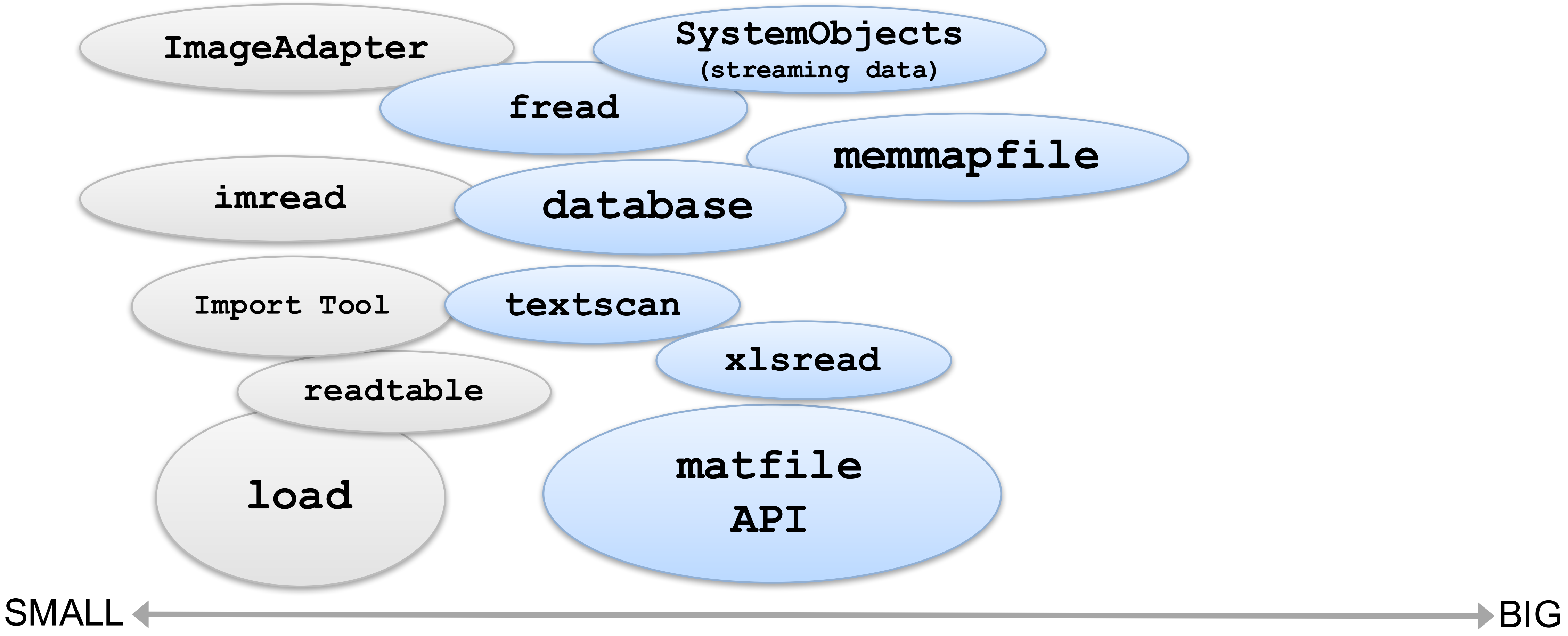
load

SMALL

BIG

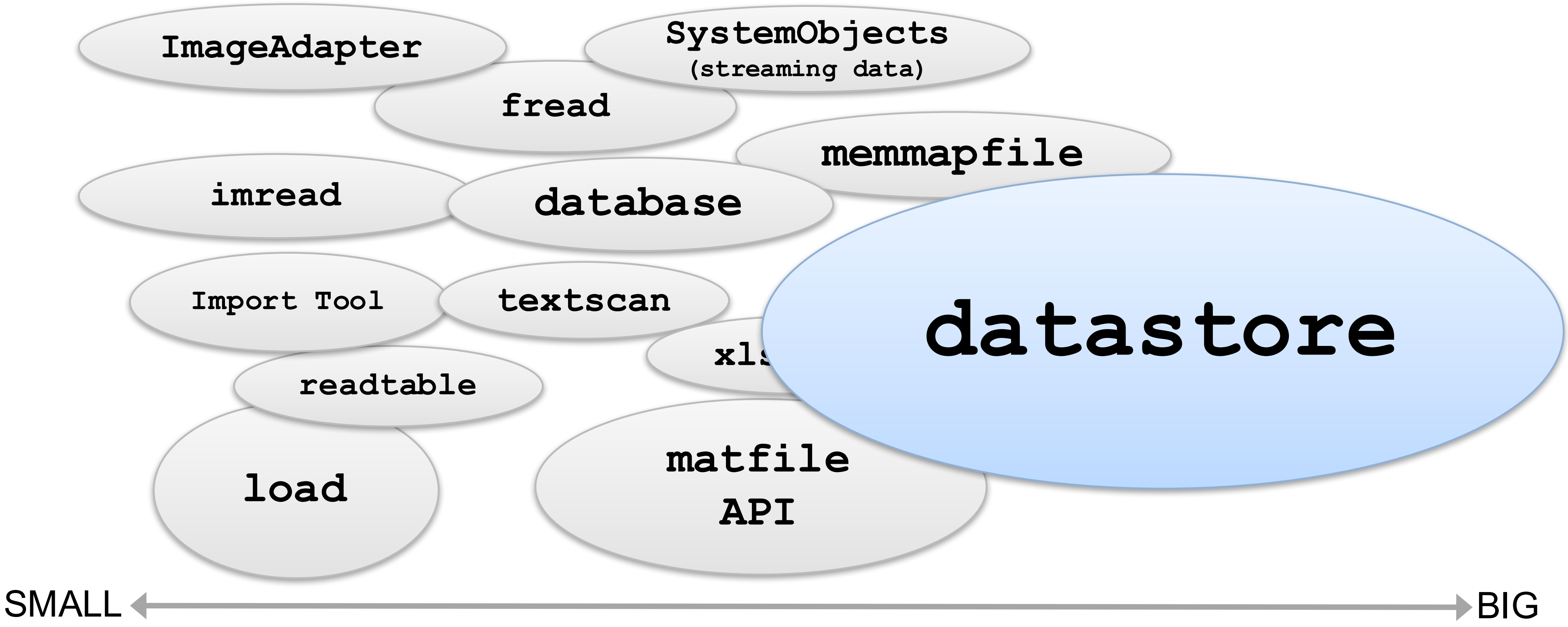
Reading big data

What tools are there?



Reading big data

What tools are there?



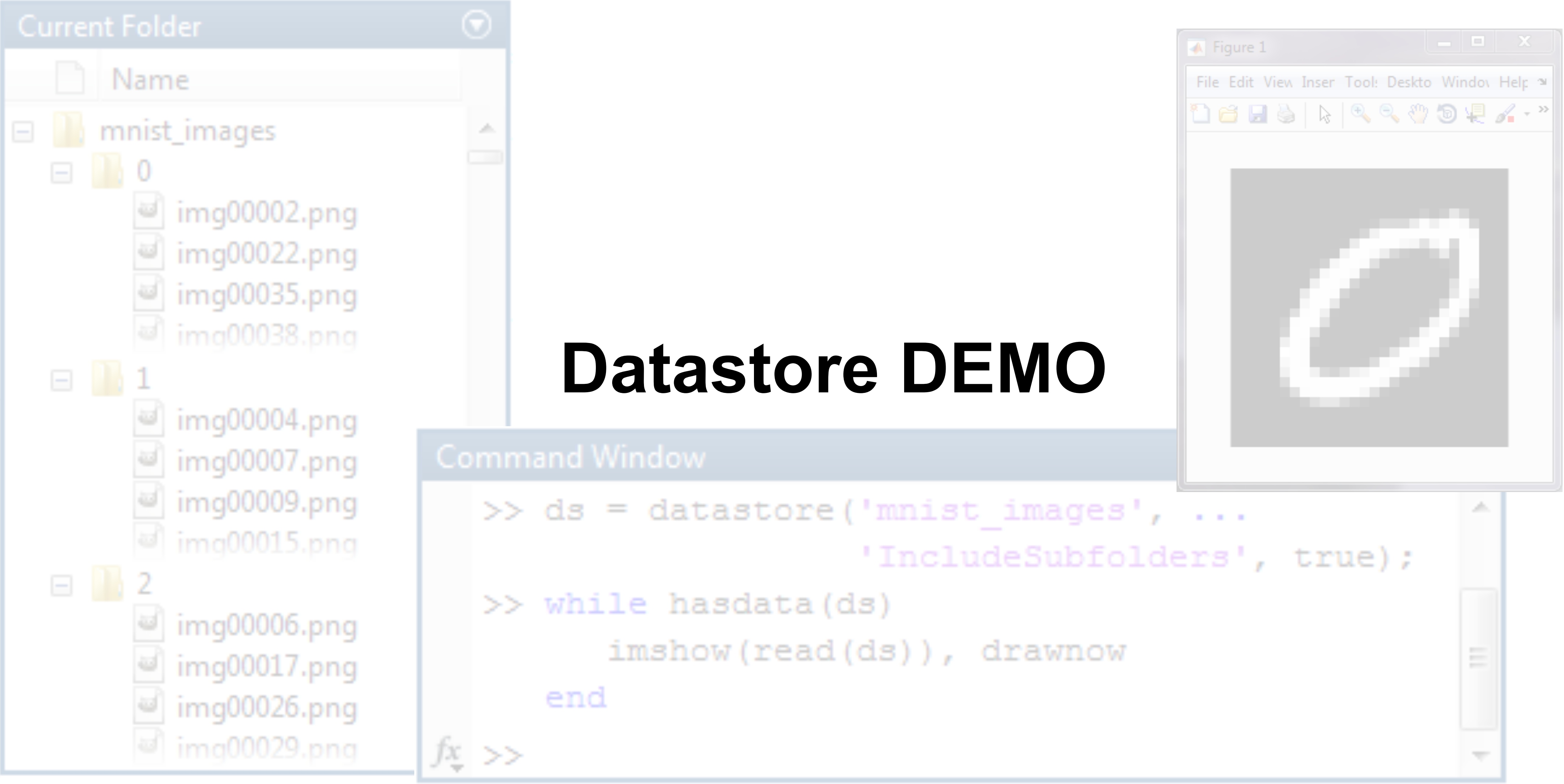
Reading big data

Datastore:

- Simple interface for data in multiple files/folders
- Presents data a piece at a time
- Access pieces in serial (desktop) or in parallel (cluster)
- Back-ends for tabular text, images, databases and more

Reading big data

Datastore DEMO



Current Folder

Name

mnist_images
- 0
  - img00002.png
  - img00022.png
  - img00035.png
  - img00038.png
- 1
  - img00004.png
  - img00007.png
  - img00009.png
  - img00015.png
- 2
  - img00006.png
  - img00017.png
  - img00026.png
  - img00029.png

Command Window

```
>> ds = datastore('mnist_images', ...
                  'IncludeSubfolders', true);
>> while hasdata(ds)
       imshow(read(ds)), drawnow
   end
>>
```

Agenda

- How big is big?
- Reading big data
- ➡ Processing quite big data
- Processing big data
- Summary

Processing quite big data

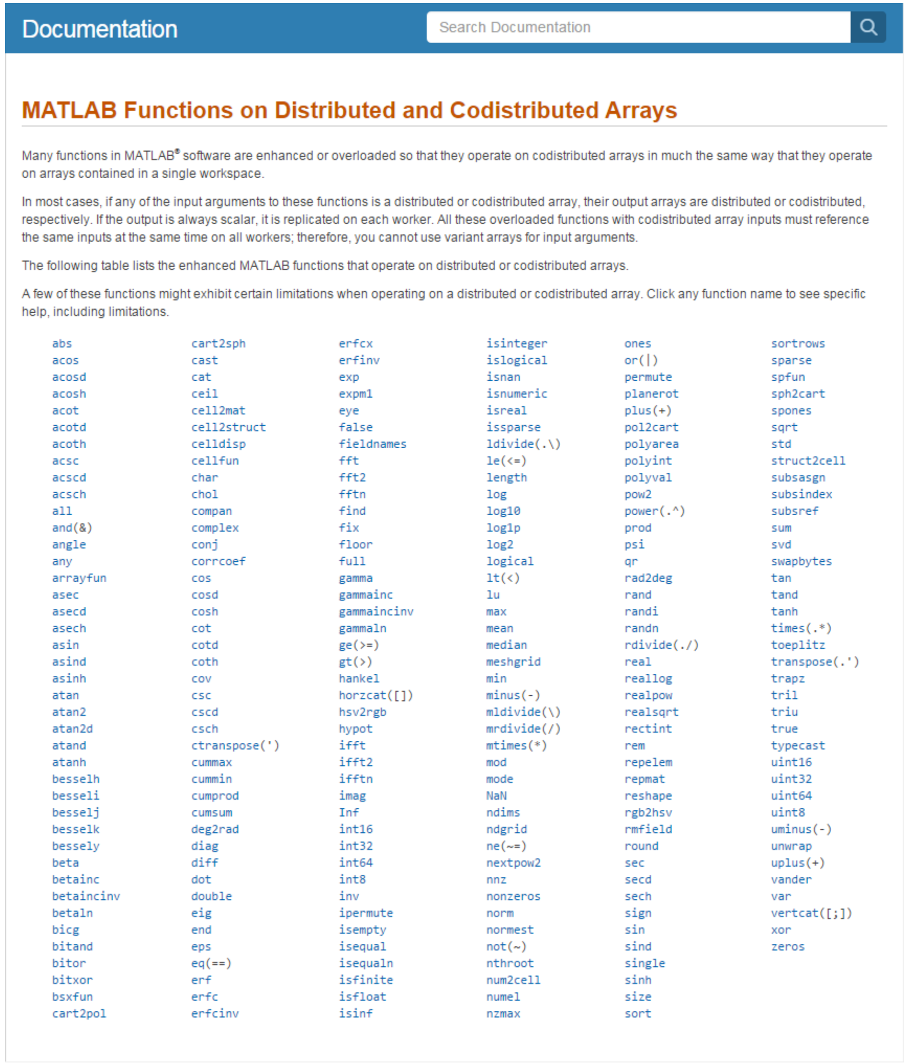
When the data fits in cluster memory

- Using distributed arrays
  - Use the memory of multiple machines as though it was your own
  - Client sees a “normal” MATLAB variable
  - Work happens on cluster

Processing quite big data

Distributed array functions

- Many common MATLAB functions supported: (about 250)
- Includes most linear algebra
- Scale up your maths



MATLAB Functions on Distributed and Codistributed Arrays

Many functions in MATLAB® software are enhanced or overloaded so that they operate on codistributed arrays in much the same way that they operate on arrays contained in a single workspace.

In most cases, if any of the input arguments to these functions is a distributed or codistributed array, their output arrays are distributed or codistributed, respectively. If the output is always scalar, it is replicated on each worker. All these overloaded functions with codistributed array inputs must reference the same inputs at the same time on all workers; therefore, you cannot use variant arrays for input arguments.

The following table lists the enhanced MATLAB functions that operate on distributed or codistributed arrays.

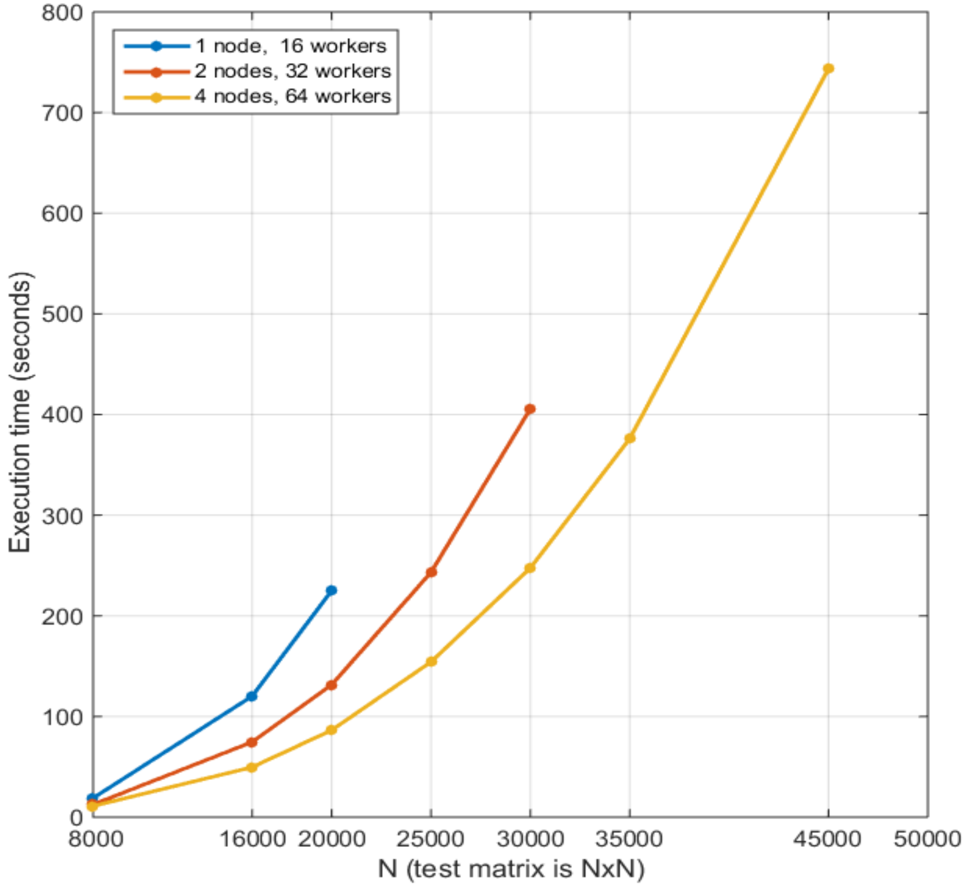
A few of these functions might exhibit certain limitations when operating on a distributed or codistributed array. Click any function name to see specific help, including limitations.

```
abs
acos
acosd
acosh
acot
acotd
acoth
acsc
acscd
acsch
all
and(&)
angle
any
arrayfun
asec
asecd
asech
asin
asind
asinh
atan
atan2
atan2d
atand
atanh
besselh
besseli
besselj
besselk
bessely
beta
betainc
betaincinv
betaln
bicg
bitand
bitor
bitxor
bsxfun
cart2pol
cart2sph
cast
cat
ceil
cell2mat
cell2struct
celldisp
cellfun
char
chol
compan
complex
conj
corrcoef
cos
cosd
cosh
cot
cotd
coth
cov
csc
cscd
csch
ctranspose(')
cummax
cummin
cumprod
cumsum
deg2rad
diag
diff
dot
double
eig
end
eps
eq(==)
erf
erfc
erfcinv
erfcx
erfinv
exp
expm1
eye
false
fieldnames
fft
fft2
fftn
find
fix
floor
full
gamma
gammainc
gammaincinv
gammaln
ge(>=)
gt(>)
hankel
horzcat([])
hsv2rgb
hypot
ifft
ifft2
ifftn
imag
Inf
int16
int32
int64
int8
inv
ipermute
isempty
isequal
isequaln
isfinite
isfloat
isinf
isinteger
islogical
isnan
isnumeric
isreal
issparse
ldivide(.\)
le(<=)
length
log
log10
log1p
log2
logical
lt(<)
lu
max
mean
median
meshgrid
min
minus(-)
mldivide(\)
mrdivide(/)
mtimes(*)
mod
mode
NaN
ndims
ndgrid
ne(~=)
nextpow2
nnz
nonzeros
norm
normest
not(~)
nthroot
num2cell
numel
nzmax
ones
or(|)
permute
planerot
plus(+)
pol2cart
polyarea
polyint
polyval
pow2
power(.^)
prod
psi
qr
rad2deg
rand
randi
randn
rdivide(./)
real
reallog
realpow
realsqrt
rectint
rem
repelem
repmat
reshape
rgb2hsv
rmfield
round
sec
secd
sech
sign
sin
sind
single
sinh
size
sort
sortrows
sparse
spfun
sph2cart
spones
sqrt
std
struct2cell
subsasgn
subsindex
subsref
sum
svd
swapbytes
tan
tand
tanh
times(.*)
toeplitz
transpose(.')
trapz
tril
triu
true
typecast
uint16
uint32
uint64
uint8
uminus(-)
unwrap
uplus(+)
vander
var
vertcat([;])
xor
zeros
```

Processing quite big data

Multiplication of 2 NxN matrices

```
>> C = A * B
```



N	Execution time (seconds)		
	1 node, 16 workers	2 nodes, 32 workers	4 nodes, 64 workers
8000	19	13	11
16000	120	75	50
20000	225	132	86
25000	-	243	154
30000	-	406	248
35000	-	-	376
45000	-	-	743
50000	-	-	-

Processor: Intel Xeon E5-class v2
16 cores, 60 GB RAM per compute node, 10 Gb Ethernet

Processing quite big data

Distributed DEMO

Agenda

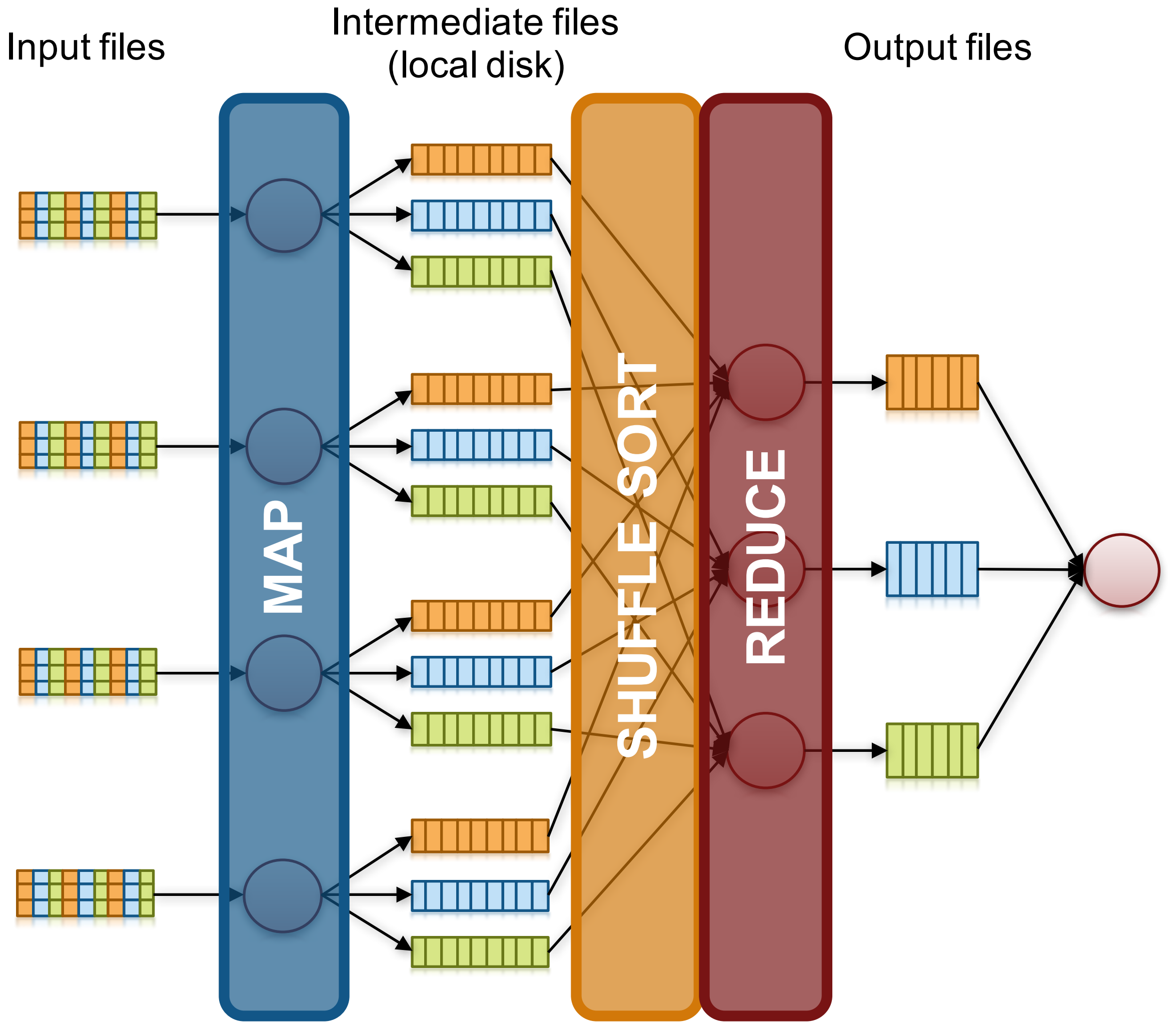
- How big is big?
- Reading big data
- Processing quite big data
- ➡ Processing big data
- Summary

Processing really big data

When you can never see all the data

- Can never have all the data loaded
- Must process small pieces of data independently
- Extract (“map”) some pertinent information from each independent piece
  - Typically summary statistics, example records, etc.
  - No communication between pieces
- Combine (“reduce”) this information to give a final (small) result
  - Intermediate results from each piece must be communicated

Introduction to Map-Reduce



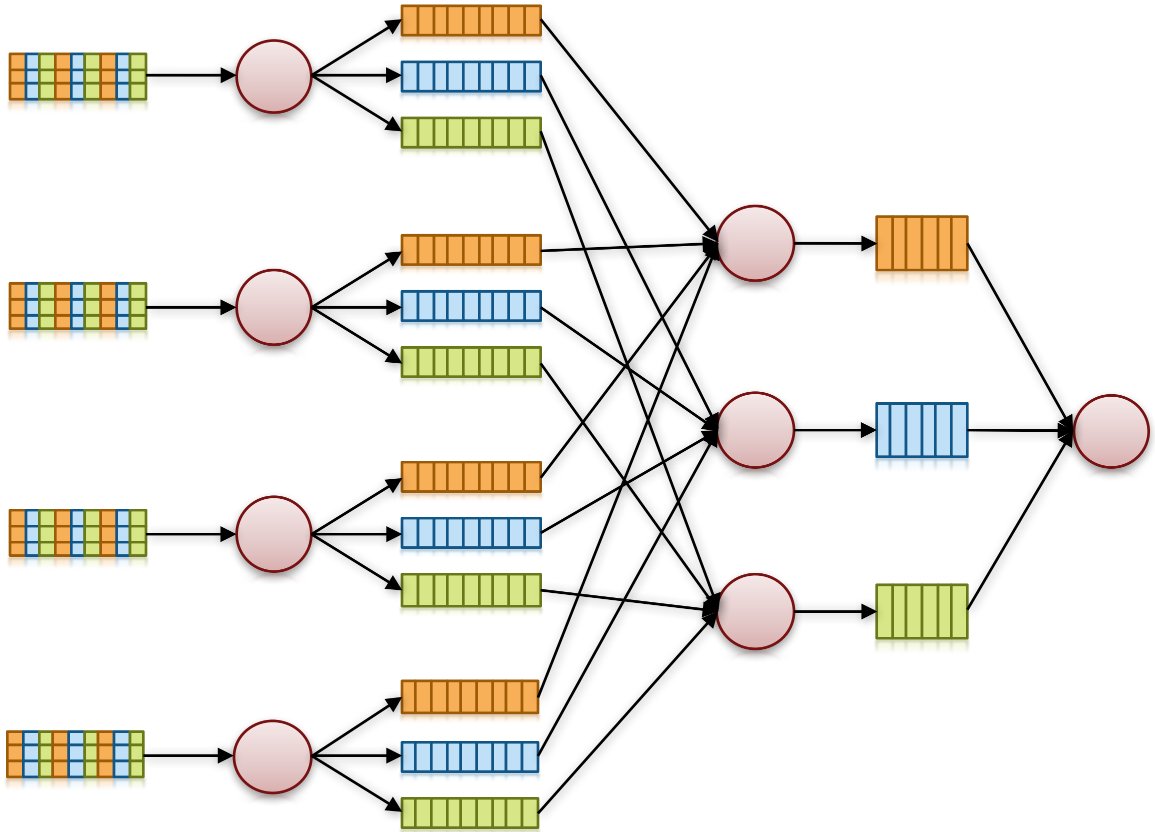
Introduction to Map-Reduce

Example:
National popularity contest

Input files

Intermediate files
(local disk)

Output files



Nicola	9%
David	53%
Jeremy	38%

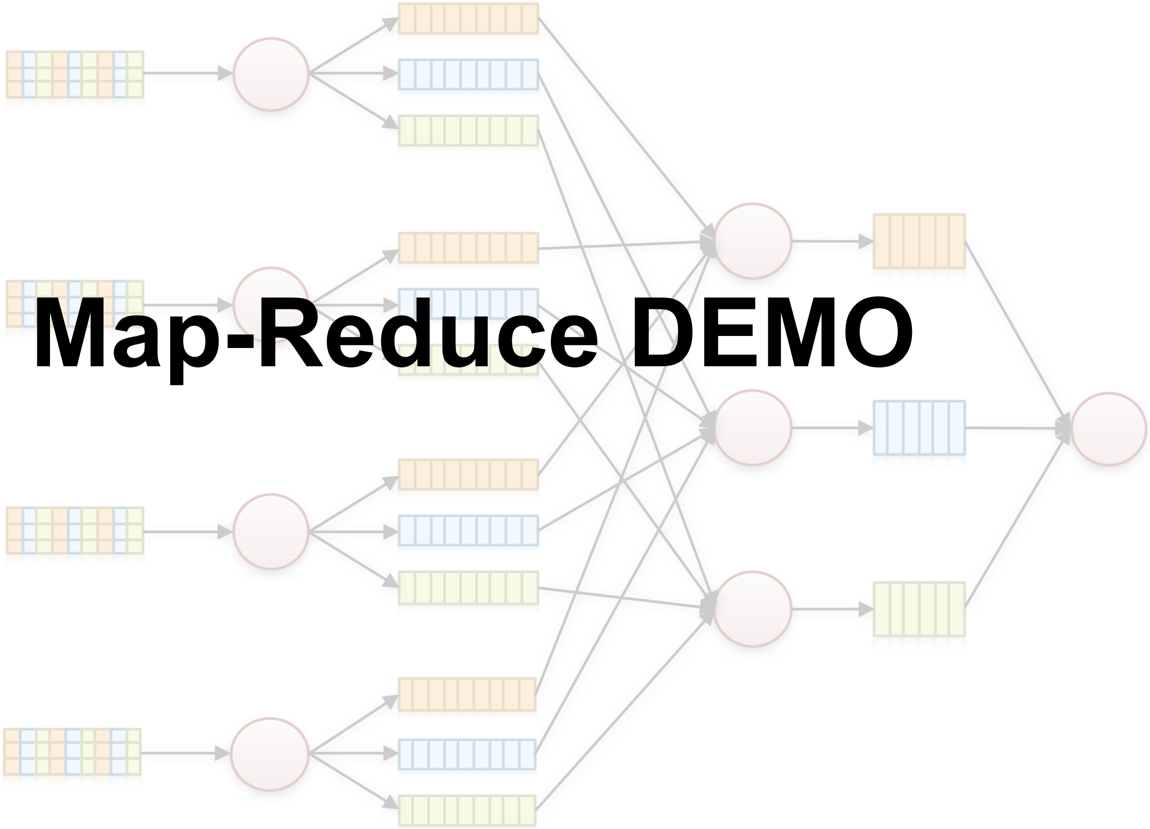
Newspaper pages

For each page how many times do “David”, “Nicola” and “Jeremy” get mentioned?

Total mentions

Relative popularity

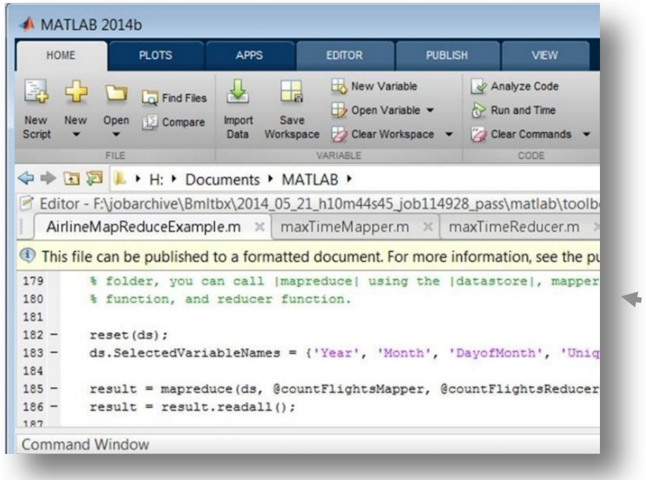
Processing medium data



Map-Reduce DEMO

MATLAB

with Hadoop



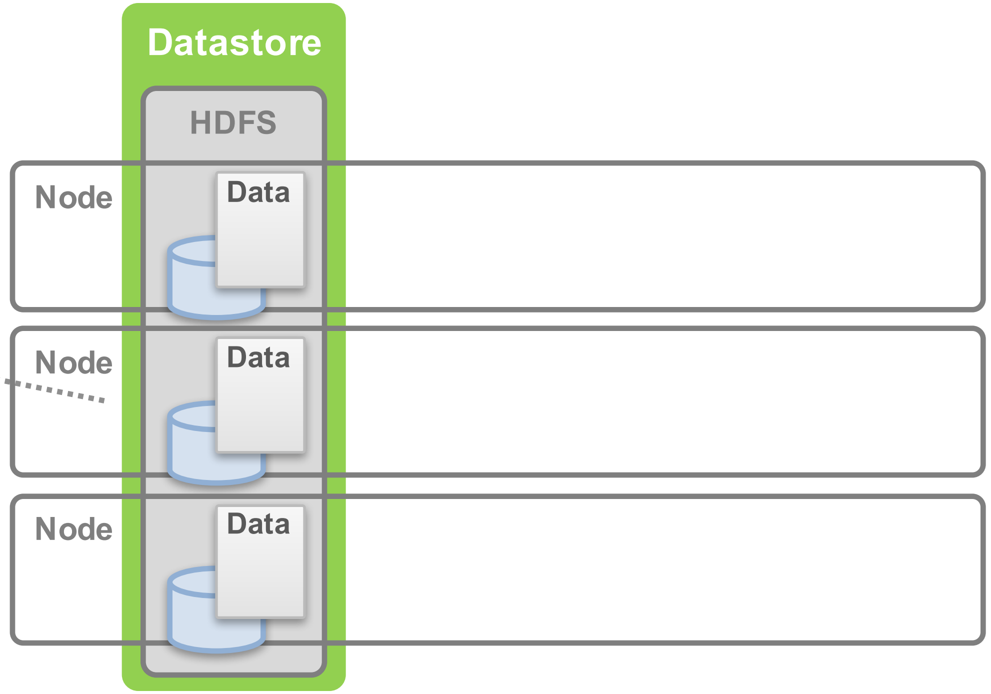
Editor - F:\jobarchive\Bmltbx\2014_05_21_h10m44s45_job114928_pass\matlab\toolb

This file can be published to a formatted document. For more information, see the pu

```
    % folder, you can call |mapreduce| using the |datastore|, mapper
    % function, and reducer function.

    reset(ds);
    ds.SelectedVariableNames = {'Year', 'Month', 'DayofMonth', 'Uniq

    result = mapreduce(ds, @countFlightsMapper, @countFlightsReducer
    result = result.readall();
```

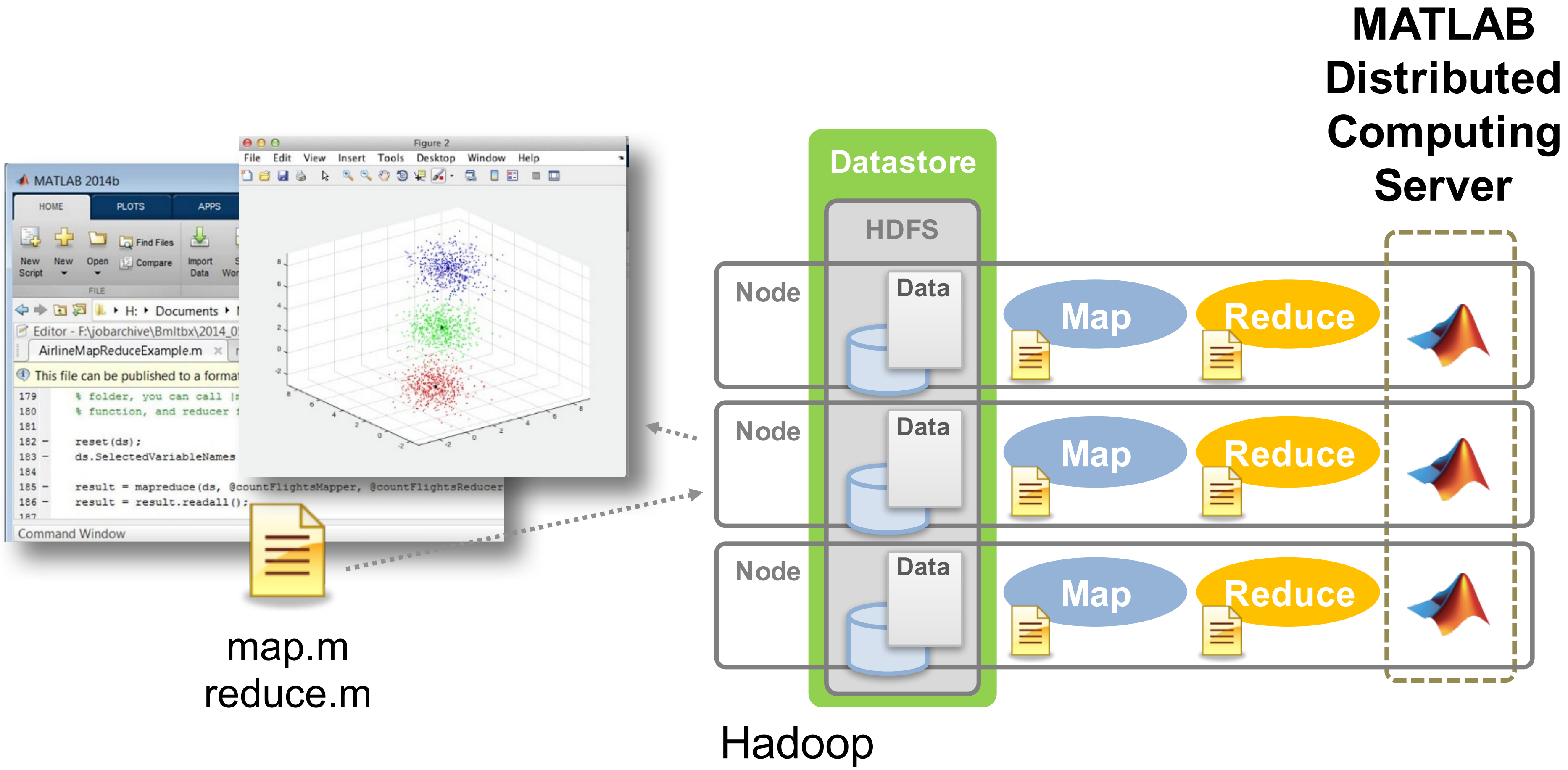


`Datastore` access data stored in HDFS from MATLAB

Hadoop

MATLAB Distributed Computing Server

with Hadoop



```
    % folder, you can call |m
    % function, and reducer f

    reset(ds);
    ds.SelectedVariableNames

    result = mapreduce(ds, @countFlightsMapper, @countFlightsReducer
    result = result.readall();
```

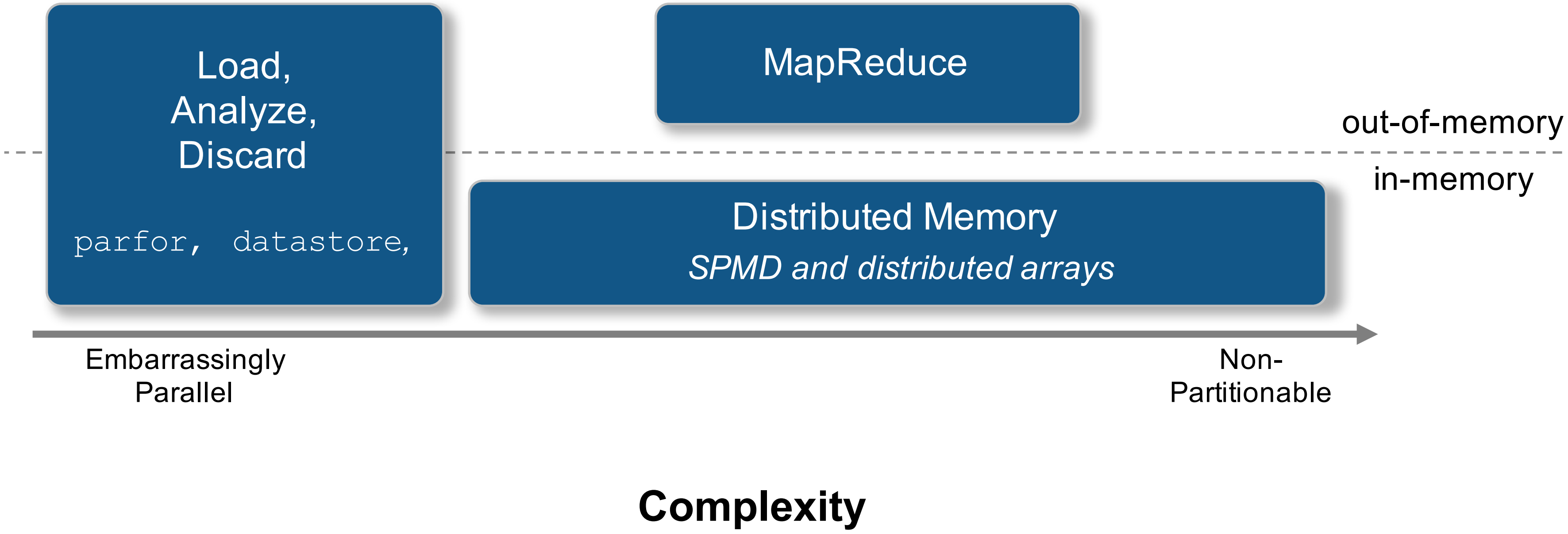
map.m
reduce.m

Hadoop

Agenda

- How big is big?
- Reading big data
- Processing quite big data
- Processing big data
- ➡ Summary

Techniques for Big Data in MATLAB



Summary

Reading data:

1. When data gets big you need to work on pieces
2. Use datastore to read pieces from a large data-set

Processing data:

1. If it fits in memory, use MATLAB
2. If it fits in cluster memory, use distributed arrays
3. If you need to scale beyond cluster memory, use map-reduce