Infrastructures and Interfaces for Data Science

Jeff Hammerbacher
What Will Be Covered

• For a large organization with dozens of data scientists
  • What are these data scientists making?
  • What infrastructure will help them do data science faster?
  • What interfaces to this infrastructure are most useful?
Assertions

- Data scientists make data and probabilistic models
- Primary infrastructure: warehouse-scale computer
- Productivity gains last ten years: infrastructure
- Productivity gains next ten years: interface
Data Science
Work Products of Data Science

• Models (data)
  • Cleansed, integrated data set
  • Interfaces to explore and manipulate the data set

• Models (probability)
  • Understanding
  • Prediction
Infrastructure
3 Kinds of Infrastructures

- HPC
- MPP
- WSC
HPC: Software

- Distributed file system
  - OneFS, GPFS, Lustre, GlusterFS, Ceph, PanFS, SiliconFS
- Resource management and job scheduling
  - LSF, SGE, Torque, Condor, SLURM
- Data management
  - NetCDF, HDF5
- Library and application development
  - MPI
HPC: Pros

- Fast inter-node communication
- Arbitrary communication topologies
- Scale compute separate from storage
- Sophisticated resource management and scheduling
- High performance, POSIX-compliant file system
- Large collection of libraries and applications
HPC: Cons

- Specialized hardware for network
- Throughput bottleneck on network
- Resource management optimized for compute
- Data management with files
- Library and application development with MPI
- Libraries and applications developed by scientists
HPC: Workloads

- Linear algebra
- Differential equations
- Distance matrices
- Monte Carlo
- FFT
HPC: Industries

• Science ($4B)
  • Government
  • University
• Life sciences ($1B)
• CAE ($1B)
• Defense
• EDA
• Earth sciences
MPP

Network

Compute and Storage
MPP: Software

- IBM DB2 and Netezza
- Teradata and Teradata Aster
- Microsoft PDW
- Pivotal Greenplum
- HP Vertica
- Actian ParAccel
- Calpont InfiniDB
- StormDB Stado
- Percona Shard-Query
MPP: Pros

- Commodity hardware (some of them)
- Compute co-located with storage
- Data management with tables
- Library and application development with SQL
- Large collection of applications
MPP: Cons

- No high-quality open source implementations
- Adding/removing nodes is expensive (usually)
- Schema-on-write/early binding of schemas
- Difficult to handle non-tabular data
- Difficult to handle non-SQL development
- Difficult to mix in non-SQL workloads
MPP: Workloads

- Data preparation
- Index creation and maintenance
- Interactive SQL query
  - Filtering
  - Aggregation
  - Join
- Model fitting, barely
MPP: Industries

- Financial services
- Communications
- Retail
- Manufacturing
WSC

Network

Compute and Storage
WSC: Software

• Distributed file storage: HDFS
• Data management
  • Avro
  • Parquet
  • Metastore
  • HBase
  • Sentry
  • SolrCloud
WSC: Software

- Resource management: YARN
- Parallel Frameworks
  - MapReduce
  - Impala
  - Spark
  - SolrCloud
- Interfaces
  - Crunch, Scrunch
  - Pig
  - SQL
  - Keyword search
WSC: Pros

- Commodity hardware
- Open source software
- Compute co-located with storage
- General purpose resource management
- Data management with files, tables, indexes, ...
- Application development with MapReduce, SQL, ...
WSC: Cons

- Commodity interconnect
- Limited communication topologies
- Non-POSIX file system
- Limited number of libraries and applications
WSC: Workloads

- Data storage
- Data transformation
- Interactive SQL queries
- Interactive keyword and faceted search queries
- Machine learning: model fitting and evaluation
- Graph computations
- Stream processing
WSC: Industries

- Communications
  - Web
  - Media
  - Telecommunications
- Government
- Retail
- Financial services
Alternatives

• Scale up
• Cloud
• SaaS federation
Interface
Wish List: Part One

• Graphics and code
• First class language
• Expressive types
• Domain syntax
• Mixed initiative
Graphics and code: RStudio
Graphics and code: iPython Notebook

**Simple spectral analysis**

An illustration of the [Discrete Fourier Transform](#)

\[ X_k = \sum_{n=0}^{N-1} x_n e^{\frac{2\pi i}{N} kn} \quad k = 0, \ldots, N - 1 \]

using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
   rate, x = wavfile.read('/home/fperez/teach/py4science/book/examples/test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin spectrogram routine:

```
In [3]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
   ax1.plot(x); ax1.set_title('Raw audio signal')
   ax2.specgram(x); ax2.set_title('Spectrogram');
```

---

**IPython Notebook**

Notebook

Actions

- New
- Open
- Download

Cell

Actions

- Format
- Code
- Markdown
- Output
- Toggle
- ClearAll
- Insert
- Above
- Below
- Move
- Up
- Down
- Run
- Selected
- All

Kernel

Actions

- Interrupt
- Restart
- Kill kernel upon exit

Help

Links

- Python
- IPython
- NumPy
- SciPy
- MPL
- SymPy

Shift-Enter: run selected cell
Ctrl-Enter: run in terminal mode
Ctrl-m hot: show keyboard shortcuts
First class language: LINQ

```csharp
//c#
public void Linq2()
{
    List<Product> products = GetProductList();

    var soldOutProducts =
        from p in products
        where p.UnitsInStock == 0
        select p;

    Console.WriteLine("Sold out products:");
    foreach (var product in soldOutProducts)
    {
        Console.WriteLine("{0} is sold out!", product.ProductName);
    }
}
```
Expressive types: Biocaml

```ocaml
type raw_to_item = [ 'header_line_not_first of int
  | 'header_line_without_version of (string * string) list
  | 'header_line_wrong_sorting of string
  | 'invalid_header_tag of int * string
  | 'invalid_tag_value_list of int * string list
  | 'reference_sequence_not_found of Biocaml_bam.raw_alignment
  | 'wrong_auxiliary_data of
    [ 'array_size of int
      | 'null_terminated_hexarray
      | 'null_terminated_string
      | 'out_of_bounds
      | 'unknown_type of char
      | 'wrong_int32 of string ] * string
  | 'wrong_cigar of string
  | 'wrong_cigar_length of int
  | 'wrong_flag of Biocaml_bam.raw_alignment
  | 'wrong_mapq of Biocaml_bam.raw_alignment
  | 'wrong_pnext of Biocaml_bam.raw_alignment
  | 'wrong_pos of Biocaml_bam.raw_alignment
  | 'wrong_qname of Biocaml_bam.raw_alignment
  | 'wrong_tlen of Biocaml_bam.raw_alignment ]
```
Domain syntax: R formulas

\[
kid\_score \sim mom\_hs + mom\_iq + mom\_hs: mom\_iq
\]
Domain syntax: ggplot2

```r
ggplot(mpg, aes(displ, hwy))+
  geom_point(aes(color = factor(cyl)))+
  geom_line()
```
Domain syntax: IBM ILOG OPL

```oql
{string} Products = ...;
{string} Components = ...;

float demand[Products][Components] = . .;
float profit[Products] = ...;
float stock[Components] = ...;

dvar float+ production[Products];

maximize
  sum (p in Products) profit[p] * production[p];

subject to {
  forall (c in Components)
  sum (p in Products)
    usageFactor[p, c] * production[p]
  <= stock[c];
}
```
Domain syntax: Fortress

- Programming language notation can become closer to mathematical notation (Unicode helps a lot)
  
  \[
  v_{\text{norm}} = v / \|v\|
  \]
  
  \[
  \sum_{k=1:n} a[k] x^k
  \]
  
  \[
  C = A \cup B
  \]
model {
  theta ~ dbeta(a, b)
  y ~ dbin(theta, n)
  Y.pred ~ dbin(theta, n.pred)
  P.crit <- step(Y.pred - n.crit + 0.5)
}
Domain syntax: Figaro

class Firm {
    val efficient = Flip(0.3)
    val bid = If(efficient, Uniform(0.0, 10.0), Uniform(5.0, 20.0))
}
val firms = Array.fromFunction(i => new Firm)(20)
def clause(firm: Firm) = (Constant(1.0), Constant(firm))
val winner = Dist(firms map clause)
val winningBid = Chain(winner, (f: Firm) => f.bid)
winningBid.constrain((bid: Double) => 20.0 - bid)
## Mixed initiative: FishEye

The directory has 10 files:

<table>
<thead>
<tr>
<th>file</th>
<th>size</th>
<th>owner</th>
<th>group</th>
<th>permissions</th>
<th>type</th>
<th>schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllstarFull.csv</td>
<td>198529</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>csv</td>
<td>Schema</td>
</tr>
<tr>
<td>FR-2011-01-03-2.xml</td>
<td>1854708</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>xml</td>
<td>Schema</td>
</tr>
<tr>
<td>FR-2011-01-03.xml</td>
<td>1852582</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>xml</td>
<td>Schema</td>
</tr>
<tr>
<td>TSCAINV_wCASRN_June2011.csv</td>
<td>8049630</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>csv</td>
<td>Schema</td>
</tr>
<tr>
<td>TSCAINV_wPMNACC_June2011.txt</td>
<td>1539739</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>csv</td>
<td>Schema</td>
</tr>
<tr>
<td>apache.txt</td>
<td>14959</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>apachelog</td>
<td>Schema</td>
</tr>
<tr>
<td>cd_catalog.xml</td>
<td>4742</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>xml</td>
<td>Schema</td>
</tr>
<tr>
<td>electricityprices.txt</td>
<td>184130</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>structured-text</td>
<td>Schema</td>
</tr>
<tr>
<td>egs7day-M1.txt</td>
<td>108469</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>csv</td>
<td>Schema</td>
</tr>
<tr>
<td>milkprices.txt</td>
<td>970</td>
<td>cloudera</td>
<td>cloudera</td>
<td>nw-r-r--</td>
<td>csv</td>
<td>Schema</td>
</tr>
</tbody>
</table>
Mixed initiative: Wrangler

<table>
<thead>
<tr>
<th>Year</th>
<th>Property_crime_rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reported crime in Alabama</td>
</tr>
<tr>
<td>1</td>
<td>4029.3</td>
</tr>
<tr>
<td>2</td>
<td>2004</td>
</tr>
<tr>
<td>3</td>
<td>2005</td>
</tr>
<tr>
<td>4</td>
<td>2006</td>
</tr>
<tr>
<td>5</td>
<td>2007</td>
</tr>
<tr>
<td>6</td>
<td>2008</td>
</tr>
<tr>
<td>7</td>
<td>4081.9</td>
</tr>
<tr>
<td>8</td>
<td>Reported crime in Alaska</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2004</td>
</tr>
<tr>
<td>11</td>
<td>2005</td>
</tr>
<tr>
<td>12</td>
<td>2006</td>
</tr>
<tr>
<td></td>
<td>3370.9</td>
</tr>
<tr>
<td></td>
<td>3615</td>
</tr>
<tr>
<td></td>
<td>3582</td>
</tr>
</tbody>
</table>
Mixed initiative: Profiler
Mixed initiative: Statwing

- **Correlation**: Age is very weakly positively correlated with Size of Team at Work.
  - **Statistical Relationship**:
    - **Statistical Significance**: Very clearly significant
    - **Effect Size**: Trivial
  - **Summary**:
    - Correlation Direction: Positive

- **There is a strong statistically significant relationship between Compensation (with Bonus, 2013) and Age**.
Wish List: Part One

- Graphics and code
- First class language
- Expressive types
- Domain syntax
- Mixed initiative
Wish List: Part Two

- Continuous feedback
- Improve over time
- Rapid problem resolution
- Unify data at rest and in flight
- Libraries across languages
Continuous feedback: Parallax
Continuous feedback: APPROXIMATE

SELECT DATE_TRUNC('day', event_time),
  APPROXIMATE COUNT(DISTINCT user_id),
  APPROXIMATE COUNT(DISTINCT url)
FROM weblog
GROUP BY 1
Improve over time: Wrangler
Improve over time: QueRIE
Rapid problem resolution: Penny
Unify data at rest and in flight: TruCQ

```sql
create view impsn_count as
  (select campaign, sum(pc) c, cq_close(*) t
  from   i_c_p <slices '1 minute'>
group by campaign);

create view impsn_count_ar as
  (select campaign, sum(pc) as c, t
  from   i_c_p_ar
  group by campaign, t);
```
Unify data at rest and in flight: Summingbird

def wordCount[P <: Platform[P]]
(source: Producer[P, String], store: P#Store[String, Long]) =
source.flatMap { sentence =>
toWords(sentence).map(_ -> 1L)
}.sumByKey(store)
Libraries across languages

• Arrays
• Data frames
• Serialization
• Workflow
• Statistics
• Machine learning
• Simulation
• Visualization
Wish List: Part Two

- Continuous feedback
- Improve over time
- Rapid problem resolution
- Unify data at rest and in flight
- Libraries across languages
Language Improvements

- Compiler directives
  - Parakeet

- Syntax extensions
  - R formulas
  - Matlab and APL array operations
  - Mathematica symbolic manipulation
  - Fortress mathematical operators
  - Fortress units and dimensions
Language Improvements

- Embedded DSLs
  - LINQ
  - plyr
  - ggplot2
  - FACTORIE
  - Figaro
  - Infer.NET
  - Weaver
Language Improvements

- External DSLs
  - IBM ILOG OPL
  - Palantir Hedgehog
  - Splunk SPL
  - Palantir VizQL
  - BUGS
  - MIT Church
  - Cray Chapel
  - Google Sawzall
  - Apache Pig
Yahoo! Pipes