"Big Data Processing – *on the cheap…*"

Joe Hummel, PhD

Visiting Researcher: U. of California, Irvine
Adjunct Professor: U. of Illinois, Chicago &
Loyola U., Chicago

Materials: http://www.joehummel.net/downloads.html
Email: joe@joehummel.net

Agenda

- Three inexpensive ways to process big data
  1. PowerPivot plugin for Microsoft Excel
  2. LINQ-based programming approach in .NET
  3. Hadoop map-reduce framework

- Discuss trade-offs
- Demos, demos, demos

How big is big?

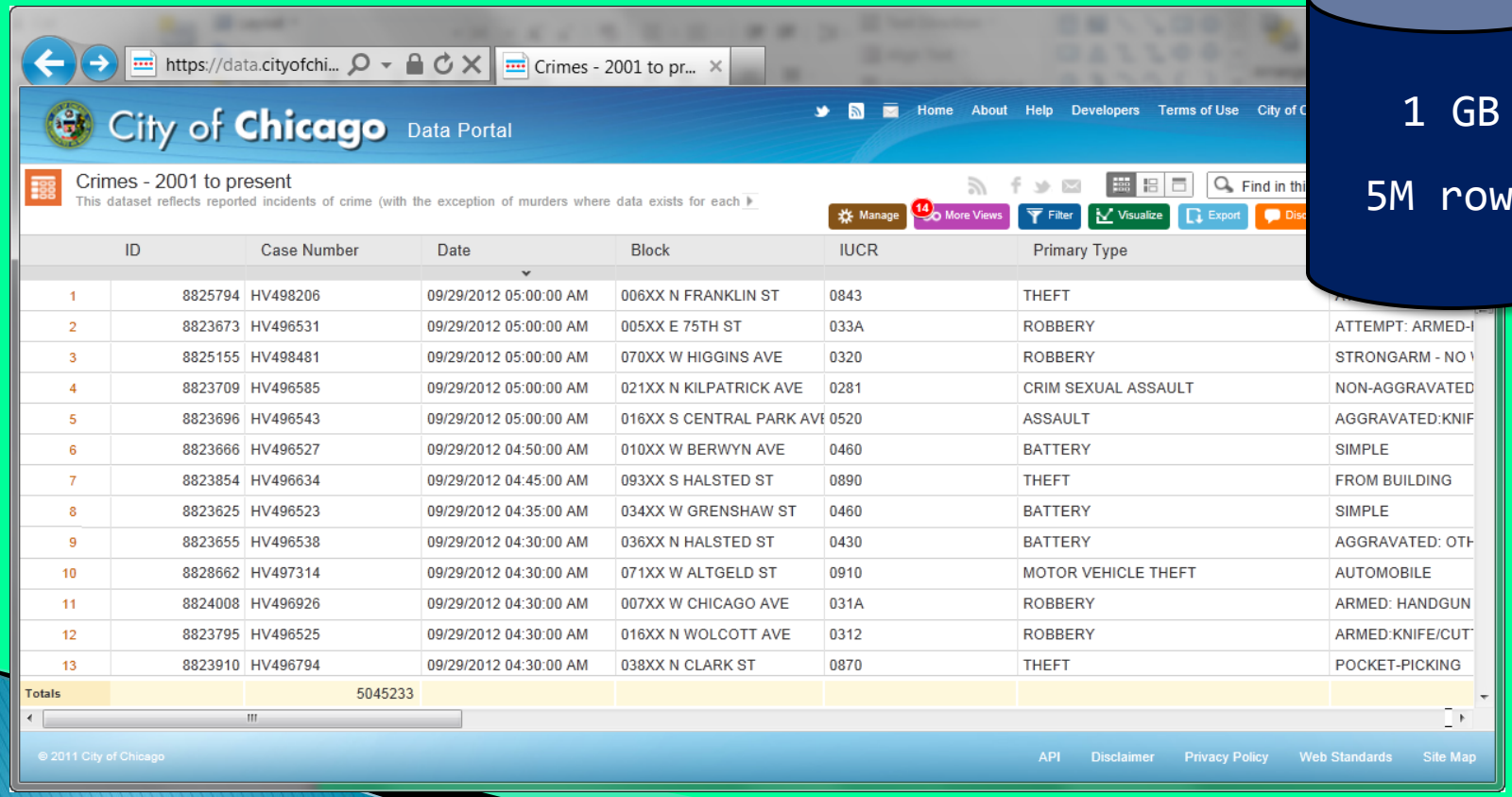
- What is your definition of big data?
  - MBs?
  - GBs?
  - TBs?
  - PBs?

Compare and contrast

Approach	Pros	Cons	Target	Scalable?
PowerPivot	No programming, built-in UI and visualization	Lack of scalability	GBs	Limited by RAM
LINQ	Flexibility of analysis	Programming	GBs, few TBs	Limited by local resources
Hadoop	Scalability, ease of programming	Must fit into Map-Reduce framework; not necessarily fast	GBs, TBs, PBs	Yes! (via cluster or cloud)

Data set for talk

- We'll be working with Chicago crime data…
  - https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2
  - http://www.cityofchicago.org/city/en/narr/foia/CityData.html

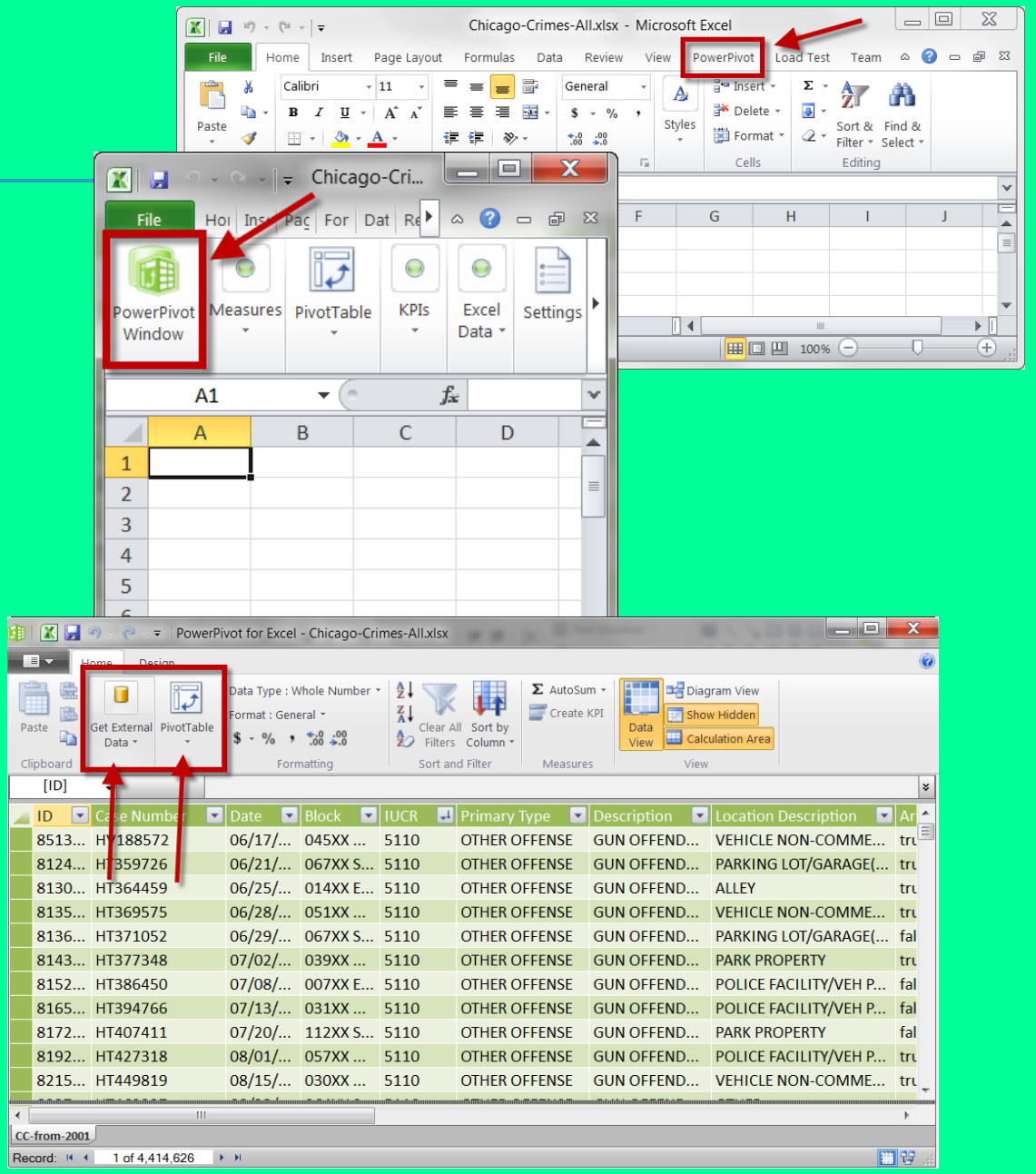


(1) PowerPivot

- Freely-available plugin for Excel 2010
  - http://www.powerpivot.com/
- Turns Excel into an in-memory database
  - *More precisely, turns spreadsheet into an OLAP cube*
- Note:
  - *If you have 32-bit Excel, install 32-bit PowerPivot*
  - *If you have 64-bit Excel, install 64-bit PowerPivot*
  - *GBs of data will require 64-bit*
  - *[How to tell what version of Excel you have? File menu, help…]*

Demo

- PowerPivot…
  - *Install*
  - *PowerPivot menu*
  - *PowerPivot Window*
  - *Get Data…*
  - *PivotTable…*

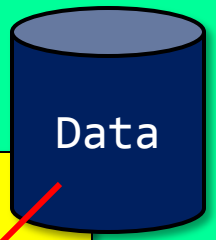


(2) LINQ

- LINQ == *Language Integrated Query*
- Traditional programming + SQL
  - http://msdn.microsoft.com/en-us/library/bb397926.aspx
  - http://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b
- Included with .NET, which is freely-available:
  - Windows: *Microsoft .NET Framework SDK*
  - Linux, Mac, Windows: *Mono project*

Demo

- LINQ…




```
// Using LinqToCsv package:
IEnumerable<CrimeReport> crimes =
    csv.Read<CrimeReport>("CC-from-2001.txt",  new CsvFileDescription {
                                SeparatorChar = ',', FirstLineHasColumnNames = true});

var query =
      from crime in crimes
      group crime by crime.IUCR into g
      orderby g.Count() descending
      select new { IUCR = g.Key, Count = g.Count() };
```

[<code1,count>, <code2,count>, …]

```
0486    366903
0820    308074
.
.
.
```

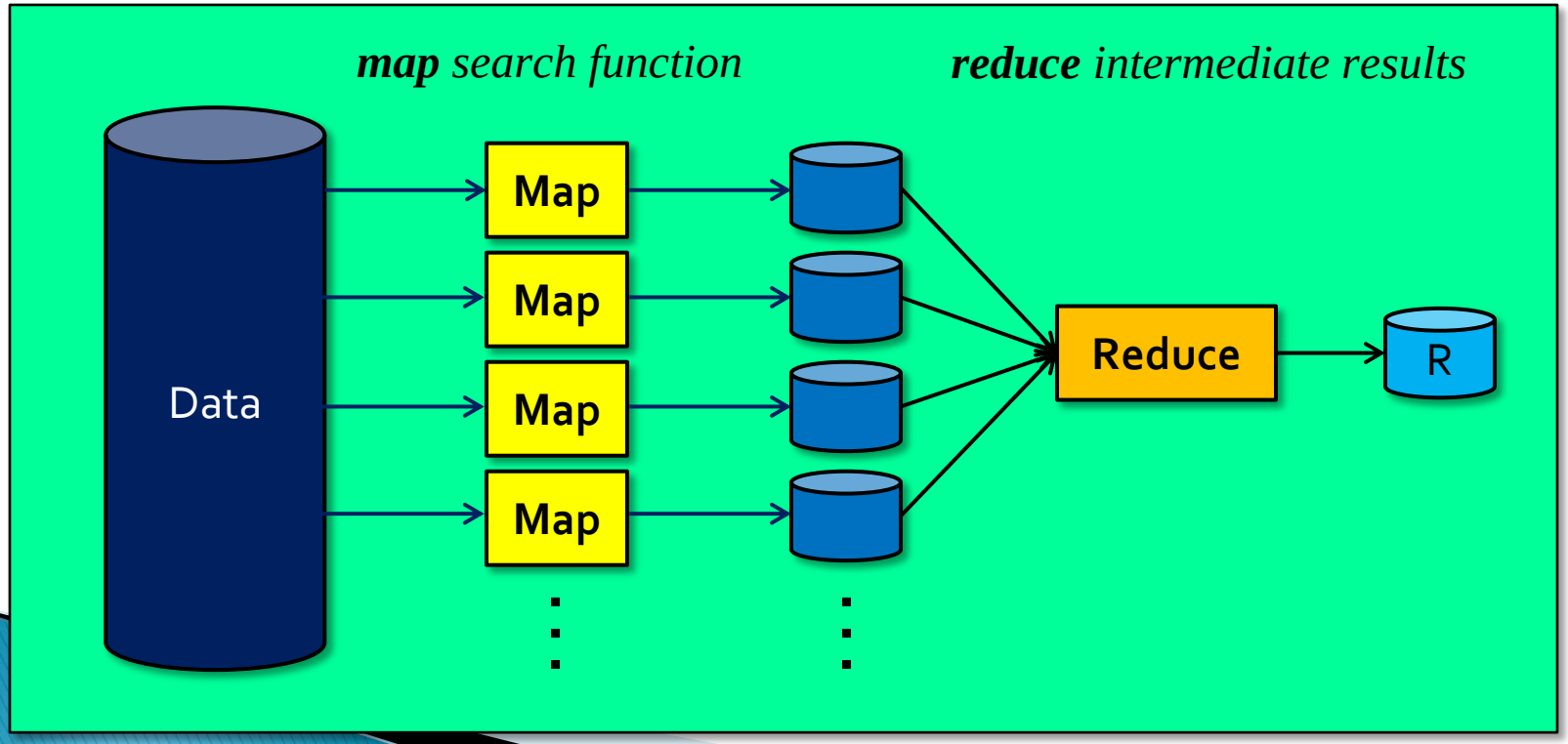
More efficient version (10x)

- Parse data inline…

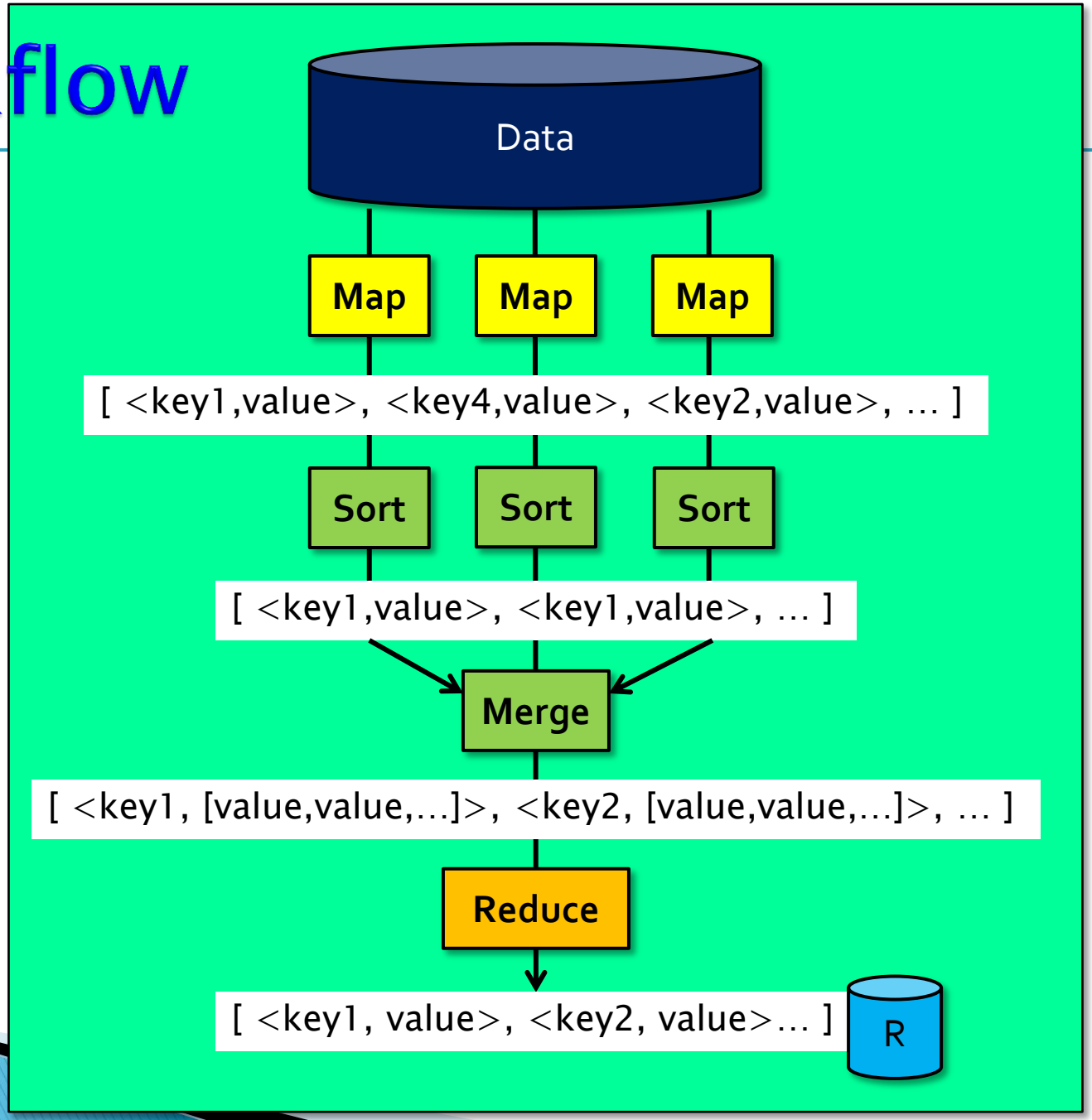
```
var query =
  File.ReadLines("CC-from-2001.txt").Skip(1).AsParallel().
  Select(line =>
  {
    string[] values = line.Split(new char[] { ',' });
    return (values.Length > 4) ? values[4] : "????";
  }).
  GroupBy(iucr => { return iucr; }).
  OrderBy(g => { return -g.Count(); }).
  Select(g => new { IUCR = g.Key, Count = g.Count() });
```

(3) Hadoop

- Freely-available framework for big data
  - http://hadoop.apache.org/
- Based on concept of Map-Reduce:



Workflow


Data
Map
Map
Map
[<key1,value>, <key4,value>, <key2,value>, …]
Sort
Sort
Sort
[<key1,value>, <key1,value>, …]
Merge
[<key1, [value,value,…]>, <key2, [value,value,…]>, …]
Reduce
[<key1, value>, <key2, value>…]
R

Demo

- Hadoop on Azure…

```
// Javascript version:
var map = function (key, value, context)
{
    var values = value.split(",");
    context.write(values[4], 1);
};

var reduce = function (key, values, context)
{
    var sum = 0;
    while (values.hasNext())
    {
        sum += parseInt(values.next());
    }
    context.write(key, sum);
};
```

Hadoop++

- Rich ecosystem around Hadoop
  - Pig
  - Hive
  - HBASE
  - …

```
// interactive PIG with explicit Map-Reduce functions:
pig.from("CC-from-2001.txt").
  mapReduce("IUCR-Count.js", "IUCR, Count:long").
  orderBy("Count DESC").
  take(10).
  to("output-from-2001")
```

```
// interactive PIG without explicit Map-Reduce:
schema = "ID,Case Number,Date,Block,IUCR,..."
pig.from("CC-from-2001.txt", schema).
  groupBy("IUCR").
  select("group, SUM($1.Count").
  orderBy("Count DESC").
  take(10).
  to("output-from-2001")
```

Hadoop on Azure

- Microsoft is offering free access to Hadoop
  - Request invitation @ http://www.hadooponazure.com/
- Hadoop connector for Excel
  - *Process data using Hadoop, analyze/visualize using Excel*

That's it!

Thank you for attending

- Presenter: Joe Hummel
  - *Email*: joe@joehummel.net
  - *Materials*: http://www.joehummel.net/downloads.html

- Upcoming products of interest…
  - PowerView:
    - *Plugin for Excel 2013*
  - Hadoop on Azure
  - Hadoop on Windows