Big Code

and Security

Roberto Giacobazzi
Big Data

Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few.

This data is “big data”

<table>
<thead>
<tr>
<th>Memory unit</th>
<th>Size</th>
<th>Binary size</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobyte (kB/KB)</td>
<td>$10^3$</td>
<td>$2^{10}$</td>
</tr>
<tr>
<td>megabyte (MB)</td>
<td>$10^6$</td>
<td>$2^{20}$</td>
</tr>
<tr>
<td>gigabyte (GB)</td>
<td>$10^9$</td>
<td>$2^{30}$</td>
</tr>
<tr>
<td>terabyte (TB)</td>
<td>$10^{12}$</td>
<td>$2^{40}$</td>
</tr>
<tr>
<td>petabyte (PB)</td>
<td>$10^{15}$</td>
<td>$2^{50}$</td>
</tr>
<tr>
<td>exabyte (EB)</td>
<td>$10^{18}$</td>
<td>$2^{60}$</td>
</tr>
<tr>
<td>zettabyte (ZB)</td>
<td>$10^{21}$</td>
<td>$2^{70}$</td>
</tr>
<tr>
<td>yottabyte (YB)</td>
<td>$10^{24}$</td>
<td>$2^{80}$</td>
</tr>
</tbody>
</table>
Big Data

Big data is any type of data - structured and unstructured data such as text, sensor data, audio, video, click streams, log files and more.

**New insights** are found when analyzing these data types together.

Example:

- Monitor 100’s of live video feeds from surveillance cameras to target points of interest
- Exploit the 80% data growth in images, video and documents to improve customer satisfaction
Sources of Big Data

- 2+ billion people on the Web by end 2011
- 30 billion RFID tags today (1.3B in 2005)
- 4.6 billion camera phones worldwide
- 100s of millions of GPS enabled devices sold annually
- 12+ TBs of tweet data every day
- 25+ TBs of log data every day
- 76 million smart meters in 2009... 200M by 2014
- 2+ billion people on the Web by end 2011
- ? TBs of data every day
Big Data Market

**Big Data Market Forecast by Component, 2011-2017**

<table>
<thead>
<tr>
<th>Year</th>
<th>Big Data XaaS Revenue</th>
<th>Big Data Professional Services Revenue</th>
<th>Big Data Application (Analytic and Transactional) Software</th>
<th>Big Data NoSQL Database Software</th>
<th>Big Data SQL Database Software</th>
<th>Big Data Infrastructure Software</th>
<th>Big Data Networking Revenue</th>
<th>Big Data Storage Revenue</th>
<th>Big Data Compute Revenue</th>
<th>Total Big Data Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>$0.35</td>
<td>$2.45</td>
<td>$0.49</td>
<td>$0.10</td>
<td>$0.72</td>
<td>$0.16</td>
<td>$0.18</td>
<td>$1.16</td>
<td>$1.64</td>
<td>$7.2</td>
</tr>
<tr>
<td>2012</td>
<td>$0.61</td>
<td>$3.87</td>
<td>$0.94</td>
<td>$0.19</td>
<td>$1.02</td>
<td>$0.26</td>
<td>$0.28</td>
<td>$1.83</td>
<td>$2.45</td>
<td>$11.4</td>
</tr>
<tr>
<td>2013</td>
<td>$1.05</td>
<td>$6.10</td>
<td>$1.80</td>
<td>$0.39</td>
<td>$1.45</td>
<td>$0.43</td>
<td>$0.44</td>
<td>$2.89</td>
<td>$3.64</td>
<td>$18.2</td>
</tr>
<tr>
<td>2014</td>
<td>$1.74</td>
<td>$9.29</td>
<td>$3.29</td>
<td>$0.73</td>
<td>$1.99</td>
<td>$0.70</td>
<td>$0.67</td>
<td>$4.40</td>
<td>$5.23</td>
<td>$28.0</td>
</tr>
<tr>
<td>2015</td>
<td>$2.47</td>
<td>$12.37</td>
<td>$5.02</td>
<td>$1.14</td>
<td>$2.47</td>
<td>$0.96</td>
<td>$0.89</td>
<td>$5.86</td>
<td>$6.70</td>
<td>$37.9</td>
</tr>
<tr>
<td>2016</td>
<td>$2.91</td>
<td>$14.14</td>
<td>$6.15</td>
<td>$1.41</td>
<td>$2.73</td>
<td>$1.12</td>
<td>$1.02</td>
<td>$6.70</td>
<td>$7.50</td>
<td>$43.7</td>
</tr>
<tr>
<td>2017</td>
<td>$3.24</td>
<td>$15.38</td>
<td>$7.00</td>
<td>$1.62</td>
<td>$2.90</td>
<td>$1.24</td>
<td>$1.11</td>
<td>$7.28</td>
<td>$8.06</td>
<td>$47.8</td>
</tr>
</tbody>
</table>
The 3Vs of Big Data

Three characteristics define Big Data:

- **Volume**: The sheer volume of data being stored today is exploding. In the year 2000, 800,000 petabytes (PB) of data were stored in the world. We expect this number to reach 35 zettabytes (ZB) by 2020. Twitter alone generates more than 7 terabytes (TB) of data every day, Facebook 10 TB, and some enterprises generate...

- **Velocity**: Batch vs. Streaming Data

- **Variety**: Structured & Unstructured

IBM characterizes Big Data by its volume, velocity, and variety—or simply, V3.
Big Code?

Code is not Data!

Structured
Diversified
Intensional
Big Code?
in US
Big Code?

Normalized Comparison

This is a chart showing combined results from all data sets, listed individually below.
Big Code?

\[
\begin{align*}
(k) \, \text{skip} \, (k') & \quad \omega \cdot (k, \varepsilon) \rightarrow \omega \cdot (k', \varepsilon) \\
(k) \, \text{true} & \quad \omega \cdot (k, \varepsilon) \rightarrow \omega \cdot (k, \varepsilon) \\
(k) \, \text{false} & \quad \omega \cdot (k, \varepsilon) \rightarrow \omega \cdot (k, \varepsilon) \\
(k) \, \text{x=random} \, (k') & \quad \forall y \in \text{Var} \setminus \{x\} : \varepsilon'(y) = \varepsilon(y) \\
& \quad \omega \cdot (k, \varepsilon) \rightarrow \omega \cdot (k', \varepsilon') \\
(k) \, \text{x=nexpr} \, (k') & \quad v \in \text{nexpr}[\mathcal{E}] (\varepsilon) \\
& \quad \omega \cdot (k, \varepsilon) \rightarrow \omega \cdot (k', \varepsilon', \varepsilon \mapsto v) \\
\end{align*}
\]
Big Data generates Big Code

**Computer Science**
Fixed programs, transient data
there will always be another input

**Data Science**
Fixed data, transient programs
there will always be another query
Big Code: getting dynamic
What Big Code trigger?

similarity
How Big Code works?

+ 

= Big Code
The 3Vs of Big Data

What Is Big Data? Hint: You’re a Part of It Every Day

Three characteristics define Big Data: **volume**, **variety**, and **velocity** (as shown in Figure 1-1). Together, these characteristics define what we at IBM refer to as “Big Data.” They have created the need for a new class of capabilities to augment the way things are done today to provide better line of sight and controls over our existing knowledge domains and the ability to act on them.

The IBM Big Data platform gives you the unique opportunity to extract insight from an immense volume, variety, and velocity of data, in context, beyond what was previously possible. Let’s spend some time explicitly defining these terms.

Can There Be Enough? The Volume of Data

The sheer **volume** of data being stored today is exploding. In the year 2000, 800,000 petabytes (PB) of data were stored in the world. Of course, a lot of the data that’s being created today isn’t analyzed at all and that’s another problem we’re trying to address with BigInsights. We expect this number to reach 35 zettabytes (ZB) by 2020. Twitter alone generates more than 7 terabytes (TB) of data every day, Facebook 10 TB, and some enterprises generate

![Diagram showing the 3Vs of Big Data: Volume, Variety, and Velocity.](image)
The 3Ds of Big Code?
Big Data pipeline in Big Code

Many levels of engagement with the data, not all requiring deep database expertise. Solutions to problems such as this will not come from incremental improvements to business as usual, as industry may make on its own. Rather, they require us to fundamentally rethink how we manage data analysis.

Fortunately, existing computational techniques can be applied, either as is or with some extensions, to at least some aspects of the Big Data problem. For example, relational databases rely on the notion of logical data independence: users can think about what they want to compute, while the system (with skilled engineers designing those systems) determines how to compute it efficiently.

Similarly, the SQL standard and the relational data model provide a uniform, powerful language to express many query needs and, in principle, allows customers to choose between vendors, increasing competition. The challenge ahead of us is to combine these healthy features of prior systems as we devise novel solutions to the many new challenges of Big Data.

In this paper, we consider each of the boxes in the figure above, and discuss both what has already been done and what challenges remain as we seek to exploit Big Data. To be done.

Standard

To be done

Overall System

Heterogeneity
Scale
Timeliness
Privacy
Human Collaboration
The **MUSE** architecture

...an example of Big Code!

---

**Artifact Generation**
- Program Analysis, Theorem Proving, Testing

---

**Graph Database and Mining Engine**

---

**Analytics**
- **Inspection**
  - Property Checking and Repair
- **Discovery**
  - Learning and Synthesis

---

**Query**
- "Synthesize a program that does X"
Typical Big Data Tools

- A/B Testing & group comparison
- Data mining and rule learning
- Classification
- Unsupervised analysis & Cluster analysis
- Data fusion and Integration
- Machine learning & pattern recognition
- Sentiment analysis
- Regression analysis and predictive modeling
- Simulation
- Time series analysis
- Network analysis
- Visualization

McKinsey Global Institute
A/B Testing & group comparison

Comparing experienced programmers (control group) against a SW enclave

- Extracting patterns for programming/debugging
- Learning attacks/defense strategies
- DB assisted extreme programming
- Models of SW design
- Design history trace in code
Data mining and rule learning

Mining over code mutations for extracting rules for code fixing and debugging

- Extracting metamorphic and polymorphic behaviors in malware
- Mining-based program analysis
- Advanced regression testing
- Reverse engineering
- Advanced similarity analysis
- Temporal model of code evolution
Classification (supervised)

Classification of similar SW with respect to key properties: safety, security, efficiency, etc.

- Provide a query-answer design of optimal code
- New metrics for SW quality by similarity analysis
- DB-assisted testing
- Advanced similarity analysis
- Advanced analysis of Code
- Analysis of analyses
Unsupervised cluster analysis

Classification of SW with respect to its use & design style (no training set)

- Malware detection by similarity
- Correlation with Big Data of use
- SW community building

- Advanced similarity analysis
- Graph-based DB of Code & User Data
Data fusion and Integration

Systematic (and automatic?) code design!

- Automatic code synthesis
- Code refactoring
- DB assisted code transformation
- Advanced code integration and fusion techniques
- Adequate artifacts associated with code (analysis, specification etc)
- Query-based code design
Machine learning & pattern recognition

Find useful code patterns

- Reusable legacy code
- Find bug patterns
- Find attack patterns in security

- Machine Learning as Program Analysis?
- Adequate semantic abstractions for making learning possible!
Sentiment analysis
Regression analysis & predictive modeling

Code development prediction

- Marketing
- Prediction of inconsistencies
- Predictive SW testing and analysis

- Advanced code integration and fusion techniques
- Adequate artifacts associated with code (analysis, specification etc)
Simulation

Large scale system simulation

- Marketing
- Security
- Massive virtualization
- Enhanced dynamic and static analysis
Time series analysis

Code design/production forecast

- Marketing
- Bug spreading
- Malware evolution
- Massive virtualization
- Enhanced dynamic and static analysis
Network analysis

Connect code

- Building communities
- Spread know-how
- Extract reusable code

- Graph-based DB for code
- Advanced query language for Big Code
- Advanced similarity analysis
Another visualization technique is one that depicts spatial information flows. The example we show here is entitled the New York Talk Exchange. It shows the amount of Internet Protocol (IP) data flowing between New York and cities around the world. The size of the glow on a particular city location corresponds to the amount of IP traffic flowing between that place and New York City; the greater the glow, the larger the flow. This visualization allows us to determine quickly which cities are most closely connected to New York in terms of their communications volume.

The New York Talk Exchange was displayed at New York's Museum of Modern Art in 2008 (senseable.mit.edu/nyte/).

History flow is a visualization technique that charts the evolution of a document as it is edited by multiple contributing authors. Time appears on the horizontal axis, while contributions to the text are on the vertical axis; each author has a different color code and the vertical length of a bar indicates the amount of text written by each author. By visualizing the history of a document in this manner, various insights easily emerge. For instance, the history flow we show here that depicts the Wikipedia entry for the word "Islam" shows that an increasing number of authors have made contributions over the history of this entry. One can also see easily that the length of the document has grown over time as more authors have elaborated on the topic, but that, at certain points, there have been significant deletions, too, i.e., when the vertical length has decreased. One can even see instances of "vandalism" in which the document has been removed completely although, interestingly, the document has tended to be repaired and returned to its previous state very quickly.


For more examples of history flows, see the gallery provided by the Collaborative User Experience Research group of IBM (www.research.ibm.com/visual/projects/history_flow/gallery.htm).

Tag cloud
This graphic is a visualization of the text of this report in the form of a tag cloud, i.e., a weighted visual list, in which words that appear most frequently are larger and words that appear less frequently smaller. This type of visualization helps the reader to quickly perceive the most salient concepts in a large body of text.

Clustergram
A clustergram is a visualization technique used for cluster analysis displaying how individual members of a dataset are assigned to clusters as the number of clusters increases. The choice of the number of clusters is an important parameter in cluster analysis. This technique enables the analyst to reach a better understanding of how the results of clustering vary with different numbers of clusters.

THINK BIG
THANK YOU!