Visualization of Network Data Provenance

Beth Plale
Director
Data to Insight Center

November 12, 2012
Menu

Provenance

Comparison

Detection

Forward

History

Provenance

Distributed

Computing

Overview

collection

Big

Data

Configurations

DDoS

Visualization

Explanatory

Exploratory

WiMAX

Attack

WiMAX
Provenance Collection Toolkit- Karma
BigData Challenges

• Large-scale Graph Data

• Faster Processing Time

• Efficient Storage mechanisms

• Faster Query Response Time

• Data Selection and Minimization
Ongoing Work to handle BigData

• Transition to a Graph Database

• Development of Quality Metrics for Provenance Data

• Automatic Provenance Identification and Collection Mechanisms

• Scalable, Robust, and Meaningful Visualization Techniques
Demos

Comparing Provenance Graphs

History Provenance

Forward Provenance

Analyzing WiMAX network configuration

Analyzing WiMAX DDoS attack

Graph abstraction and failure detection
Introduction to Our Projects
Our first project studies the provenance of a NASA satellite imagery ingest processing pipeline. The instrument generating the images is the AMSR-E (Advanced Microwave Scanning Radiometer, Earth Observing System).
History Provenance (Demo)
Forward Provenance

Our first project studies the provenance of a NASA satellite imagery ingest processing pipeline. The instrument generating the images is the AMSR-E (Advanced Microwave Scanning Radiometer, Earth Observing System).
Forward Provenance (Demo)
Comparing Provenance Graphs

We improve and implement the Direct Classification of node Attendance (DCA)* algorithm to compare two provenance graphs by finding matching sub-graphs.

Graph matching to support visually comparing provenance of two daily sea-ice workflow executions. Red nodes are matched nodes, and the many unmatched blue nodes indicate that the right execution is potentially problematic.

Comparing Provenance Graphs (Demo)
Comparing Provenance Graphs

We improve and implement the Direct Classification of node Attendance (DCA)* algorithm to compare two provenance graphs by finding matching sub-graphs.

The provenance graph on the left is a complete provenance of a successful execution. Comparing it with the provenance graph on the right shows that the right one is a failure, because of the final data product (green) in the left graph is not matched.

Comparing Provenance Graphs (Demo)
Comparing Provenance Graphs

We improve and implement the Direct Classification of node Attendance (DCA)* algorithm to compare two provenance graphs by finding matching sub-graphs.

The provenance graph on the left is a complete provenance of a successful execution. Comparing it with the provenance graph on the right shows that although the right graph is a successful execution, it has some dropped notifications, because all the nodes except some edges in the left graph are matched.

Comparing Provenance Graphs (Demo)
Abstracting Provenance to Detect Execution Failure

Initial provenance graph under circular layout

Clustering neighbor nodes and applying hierarchical layout

Abstract provenance graph shows where and how each process ran, and what output was generated. It indicated that the last (rightmost) BFS process was not triggered successfully, and the “stop twister” process failed without output.

*Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox, Twister: A Runtime for Iterative MapReduce* The First International Workshop on MapReduce and its Applications (MAPREDUCE'10) - HPDC2010
Abstracting Provenance to Detect Execution Failure (Demo)
Analyzing WiMAX DDoS Attack

<table>
<thead>
<tr>
<th>Run id</th>
<th>Frame duration</th>
<th>number of attackers</th>
<th>attack backoff start</th>
<th>attack request retry</th>
<th>bw backoff start</th>
<th>bw request retry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.004</td>
<td>20/80</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>244</td>
<td>0.01</td>
<td>20/80</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>487</td>
<td>0.02</td>
<td>20/80</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Side-to-side performance comparison (right graph) shows that packets dropped increases as frame_duration increases from 0.01s to 0.02s.

Zoomed in visualization shows detailed attributes of each dropped packet (red), delivered packet (blue), sender node (green) and process (green).

Analyzing WiMAX DDoS Attack (Demo 1)
Analyzing WiMAX DDoS Attack (Demo 2)
Analyzing WiMAX Network Configurations

This initial view shows that the WiMAX nodes generated significantly different traffic levels as depicted by the size of their node clusters.

This graph shows the packet traffic at the receiver node. The NetKarma plug-in colors the packets according to their source IP address, showing that traffic from only two of the nodes (yellow and green) was successfully received.

By zooming in and comparing the metadata harvested about each sender, we can see that it’s the difference in configuration of OTG2 that is resulting in different throughput results.
Analyzing WiMAX Network Configurations (Demo)