Semantic Representation and Scale-up of Integrated Air Traffic Management Data

Rich Keller, Ph.D. *
Mei Wei*
Shubha Ranjan+●
Michelle Eshow●

*Intelligent Systems Division / +Aviation Systems Division
●Moffett Technologies, Inc.

NASA Ames Research Center

International Workshop on Semantic Big Data, San Francisco, USA, July 1, 2016

Work funded by NASA’s Aeronautics Research Mission Directorate

Point of contact: rich.keller@nasa.gov
Aviation Data is Big Data

• **Volume**: 30M+ flights yearly
  3.6B passengers forecast for 2016

• **Variety**: flight tracks, weather maps, aircraft maintenance records, flight charts, baggage routing data, passenger itineraries

• **Velocity**: high frequency data from aircraft surveillance systems and on-board health & safety systems 24x7
Build a large queryable semantic repository of air traffic management (ATM) data using semantic integration techniques.
Can semantic representations scale up to accomplish practical tasks using Big Data?

→ Conduct a scale-up experiment to answer the question
Outline

• Aviation Data Integration Problem
• Semantic Integration Approach
• Design of our Scale-up Experiment
• Results
• Approaches to Improving Scale-up Performance
• Conclusions
Background: Aviation Data Integration Problem

- NASA researchers require historical ATM data for future airspace concept development & validation

- NASA Ames’ **ATM Data Warehouse** archives data collected from FAA, NASA, NOAA, DOT, industry
  
  - Warehouse captures 13 sources of aviation data:
    - flight tracks, advisories, weather data, delay stats
    - some from live feeds and some from periodic updates

  - Data holdings available back to 2009

  - 30TB of data; some in a database; most in flat files
Problem: Non-integrated Data

- ATM Warehouse data is replicated & archived in its original format
- Data sets lack standardization
  - data formats
  - nomenclature
  - conceptual structure
- To analyze and mine data, researchers must download data and write special-purpose integration code for each new task
  ➔ Huge time sink!

- Possible cross-dataset mismatches:
  - terminology
  - scientific units
  - temporal/spatial alignment
  - conceptualization organization
Proposed Solution

Relieve users of responsibility for integration

Integrate Warehouse data sources
on the server side
using Semantic Integration
Semantic Integration Approach: Prototype System Diagram

- ATM Warehouse (subset)
- Flight Track
- Weather
- Airspace Advisories
- FAA
- Other Data Sources
- Airlines, Aircraft, Airport Info
- ASPM
- SPARQL Queries
- Large Triple Store
- Common Cross-ATM Ontology
- Translators

Data sources are integrated into a Large Triple Store through translators. The data is then represented in a Common Cross-ATM Ontology.
ATM Ontology

- 150+ classes
- 150+ datatype properties
- 100+ object properties
Ontology Representation of a Flight

**KATL Airport**
- airport name: Hartsfield-Jackson
- FAA airport code: ATL
- ICAO airport code: KATL
- located in state: GA
- offset from UTC: -5

**Flight DAL1512**
- actual arrival: 2012-09-08T20:35
- actual depart: 2012-09-08T19:03
- call sign: DAL1512
- user category: commercial
- flight route string: KATL.CADIT6...

**KORD Airport**
- airport name: O’Hare Int'l.
- FAA airport code: ORD
- ICAO airport code: KORD
- located in state: IL
- offset from UTC: -6

**Delta Air Lines**
- name: Delta Air Lines
- callsign: DELTA
- ICAO carrier code: DAL
- IATA carrier code: DL

**Flight Track for DAL1512**

**Rway 09R/27L**
- runway ID = 09R/27L

**KATL Weather@18:52**
- dewpoint: 19
- report time: 2012-09-08T18:52
- report string: KATL 301852Z 11004KT...
- surface pressure: 1010.1
- surface temperature: 22

**Aircraft N342NB**
- registrant: Delta Air Lines, Inc.
- serial number: 1746
- certificate issue: 2009-12-31
- manufacture year: 2002
- mode S code: 50742752
- registration number: N342NB

**A319-111**
- AC type designator: A319
- model ID: A391-111
- number engines: 2

**Airbus**

**Aeronautical**

---

**KEY**
Experimental Methodology

1. Develop ontology
2. Write data source translators
3. Run translators to generate data for a period covering one day of air traffic to/from a major airport (Atlanta): 1342 flights; ~2.4M triples
4. Load data into two commercial triple stores (AllegroGraph/Franz and GraphDB/Ontotext)
5. Develop a set of SPARQL performance benchmark queries and run on both triple stores
6. Replicate one day’s worth of data x 31 to approximate one month of air traffic: ~40+K flights; ~36M triples*
7. Run queries again to compare results

*Estimate: 10B triples/yr. for US domestic flights
Sample Benchmark SPARQL Queries - from a set of 17 queries for evaluating performance on scale-up -

• Flight Demographics:
  – F1: Find Delta flights using A319s departing Atlanta-area airports
  – F3: Find flights with rainy departures from Atlanta airport

• Airspace Sector Capacity:
  – S6: Find the busiest US airspace sectors for each hour in the day

• Traffic Management Statistics:
  – T1: Find flights that were subject to ground delays

• Weather-Impacted Traffic:
  – W1: Calculate hourly impact of weather on flight delays

• Flight Delay Data:
  – A3: Compare hourly airport arrival capacity with demand
Results for 17 benchmark queries

<table>
<thead>
<tr>
<th>Flight Period</th>
<th>Execution Time</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td></td>
</tr>
<tr>
<td>1 Day</td>
<td>11 ms</td>
<td>9.6 sec</td>
<td>1.19 sec</td>
<td></td>
</tr>
<tr>
<td>1 Month</td>
<td>8 ms</td>
<td>1651.2 sec</td>
<td>(170x increase)</td>
<td>96.65 sec</td>
</tr>
</tbody>
</table>

Observations:
- ~30% of queries experienced no increase in execution time
- ~60% of queries scaled in proportion to increase in triples
- 1 query experienced exponential increase (350x – 700x, depending on triple store)

Conclusion: Scaling to multi-year flight periods does not appear feasible unless multi-hour or multi-day response times are acceptable
5 Potential Scale-Up Approaches

1. **Hardware**: triple ‘appliances’ for faster storage, retrieval & processing

2. **Algorithm**: better graph matching algorithms

3. **Software**: better query planners; new indexing approaches

   - Hardware designers, researchers, triple store architects \((1,2,3)\)
   - Application developers, triple store users \((4,5)\)

4. **Query reformulation**: rewrite queries

5. **Triple reduction**: reduce graph search space
4. Query Reformulation

- SPARQL queries can (in theory) be rewritten to improve efficiency
- Lack of transparency regarding how SPARQL queries are translated into code and executed makes rewriting difficult
- Tools to assist with optimization are missing or poorly documented
- **Wanted!:** ■performance monitoring tools
  ■query plan inspector  ■index formulation tools
- SQL performance analysis tools are mature; SPARQL tools are primitive (in our experience)
Current Status Update

• Have scaled up to 1 month of actual flight data from the three NY Metropolitan airports:
  ~257M triples
  → considerably more than the 36M/month reported for Atlanta airport in the paper

• Will be re-testing benchmark queries against this data, but not easily comparable to existing data due to changed geographic region
Summary

- Described a real-world practical application for big semantic data: integrating *heterogeneous ATM data*
- Reviewed experiments performed to scale-up data and measure impact on query performance
- Discussed approaches to improving performance

**Conclusion**: Adequate tools not yet available to support real-world performance tuning for SPARQL queries in commercial triple stores

**Caveat**: Experience limited to only 2 triple stores!
In the end

Q: Can semantic representations scale to accomplish practical tasks using Big Data?

A: Well, I’m still not sure!

(...to be continued)
Triple Reduction

- Reduce the underlying search space by modifying the representation
- Undesirable trade-off possible: → trade representational fidelity for efficiency

Example: *representation of Aircraft Track Points*
TrackPoint Representation Tradeoff

**Representation #1**  vs.  **Representation #2**

(2 inst. per minute: ~70% of all instances)  (1 inst. per minute: ~54% of all instances)

**AircraftTrackPoint**
- reporting time: 2012-09-08T19:03:00
- sequence number: 31
- ground speed: 461

**hasFix**

**GeographicFix**
- altitude: 3700.0
- latitude: 33.6597
- longitude: -84.495555

**AircraftTrackPoint**
- reporting time: 2012-09-08T19:03:00
- sequence number: 31
- ground speed: 461
- altitude: 3700.0
- latitude: 33.6597
- longitude: -84.495555