Performance vs. Extensibility and Ease of Use: Next Steps in the NMWG Schemata

Martin Swany
University of Delaware

Requirements and Goals

- Current schema continues to evolve to address community needs
  - Application/middleware developers
  - End users
  - Network managers
  - Network researchers
- Broad community ⇒ conflicting goals
  - Easy to read/parse/understand/extend
  - Efficient encoding/decoding/querying
- Goal: single interface usable in different ways, building on Grid infrastructure for authentication, authorization, accounting and basic services (SOAP, WSDL, etc.)
  - XML is the current syntax

Data/Metadata Separation

- For a series of data, much of the metadata is consistent
- Therefore we separate the consistent parts of the metadata by reference
  - Reference is to an XML object, which may or may not be included in the same SOAP message
- Simply need to refer to an ID that is unique within some scope
  - Single document, single connection, single session, global (with URI, Grid Svc Handle,…)

Data Normalization

- All data can be identified by 3 broad classes of metadata and a timestamp
  - **Characteristic**
    - What was being measured, the type of event
  - **Entity/Subject/Target**
    - What entity was measured, generated the event
  - **Parameters/Methodology**
    - What parameters were fed to the measurement tool, what were the conditions under which the event was generated, by whom, what system, etc.
- Self-contained instances of data can be formed by performing logical joins on these basic components

Normalization Example

- Consider a set of available bandwidth measurements made periodically
- Long-running middleware can initially determine the characteristic and its parameters
- Subsequent queries can request values in a certain timestamp range to learn recent values and variance
  - Potentially significant reduction in overhead
- Even historic packet/application traces

Example 2 - Traceroute

- We can normalize a single execution of traceroute into a series of packets sent from ***src*** to ***dst*** with various TTLs
  - As opposed to encoding the output as a string or a list of numeric quantities.
  - The individual elements have a common ID
- This makes forward and reverse path trees (or path fan-in and fan-out) expressible as a simple set of queries
- Some versions of traceroute allow specification of the initial TTL
  - Perhaps after the first “few” traceroutes return consistent hops 1-3, subsequent queries can begin at the diverging hop

Functions on the data

- Clearly, some statistical analysis of data can be useful
- Often more efficient to get summary data
- Some folks want to do it themselves
- So, you can get the raw data or have a specific function performed on the data
- Use WSDL to describe functions' input, output even if those functions are invoked in series on the server side

Derived data streams

- Timeseries transformations based previously mentioned functions
- The subject becomes a view
  - a subject, characteristic, parameters, timestamp query
- The characteristic and parameters encode the transformation on the original data
- OWAMP only makes averages available
  - Our approach can describe the transformed data stream

Other potential benefits

- Result verification against private data
  - Create service instance to perform remote analysis
- In general, I might have some data that I can allow you to run certain queries against, but not download completely

Summary

- We must allow for various levels of use via the same mechanisms
  - Inline metadata is a logical join of data and metadata
- Function chaining with WSDL signatures allows for management of remote processing
  - Derived data streams
  - Description of processing steps for available data