End-to-End Distributed Application Monitoring using the "Distributed Monitoring Framework"

Brian L. Tierney
(bltierney@lbl.gov)

Lawrence Berkeley National Laboratory

Components

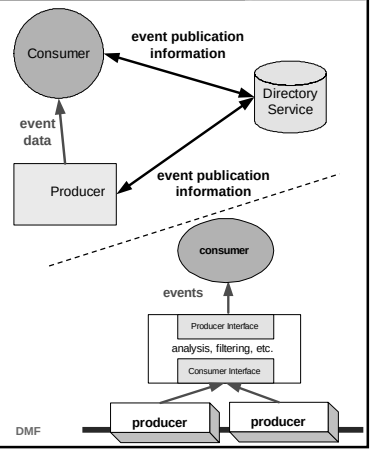
- A complete End-to-End monitoring framework must include the following:
  - host and network sensors
  - sensor management tools
  - event publication service
  - event archive service
  - instrumentation tools
  - event analysis and visualization tools

Common Protocols

- We need a monitoring framework that provides a unifying view to a wide range of sensor data, from network to host to application.
- This requires common protocols and data formats:
  - event data descriptions
  - event dictionaries
  - query format
  - publish/subscribe APIs and protocols
  - timestamp format
  - types
  - etc.
- **Using XML-based solutions for this problem**
  - working with Global Grid Forum to define these

Event Publication: GMA Terminology and Architecture

- (Performance) Event:
  - Typed collection of data with a specific structure
- Producer Interface:
  - makes performance data (events) available
- Consumer Interface:
  - receives performance data (events)
- Directory Service:
  - supports information publication and discovery



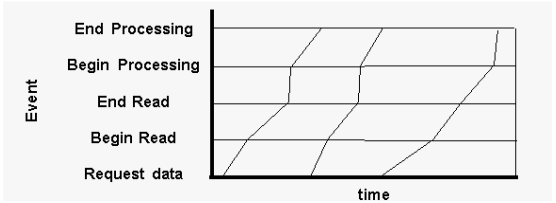
NetLogger Components

- NetLogger Toolkit contains the following components:
  - NetLogger message format
  - NetLogger client library (C, C++, Java, Perl, Python)
  - NetLogger visualization tools
  - NetLogger host/network monitoring tools
  - NetLogger storage and retrieval tools (new)
- Source code (open source) and binaries are available at:
  - http://www-didc.lbl.gov/NetLogger/
- Additional critical component for distributed application analysis:
  - NTP (Network Time Protocol) is required to synchronize the clocks of all systems

DMF

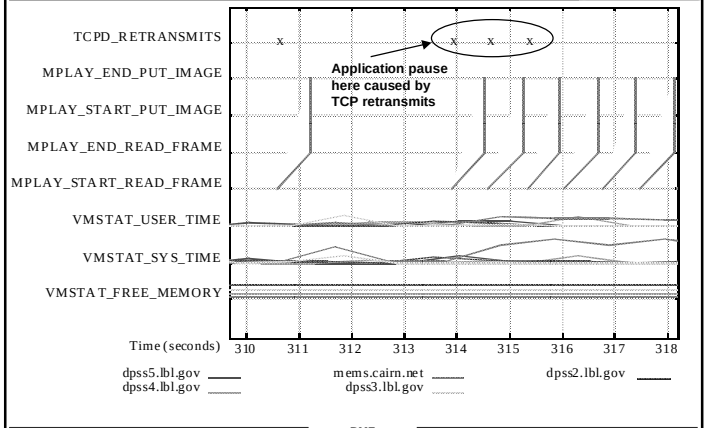
Key Concepts

- NetLogger visualization tools are based on time correlated and/or object correlated events.
  - precision timestamps (default = microsecond)
- If applications specify an "object ID" for related events, this allows the NetLogger visualization tools to generate an object "lifeline"
- In order to associate a group of events into a "lifeline", you must assign an "**Event ID**" to each NetLogger event
  - Sample Event ID: file name, block ID, frame ID, etc.



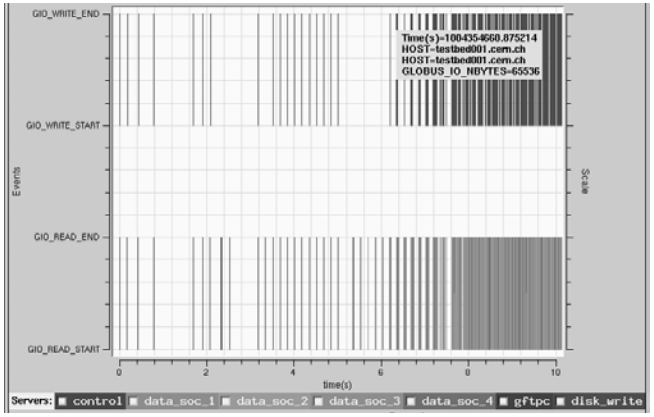
DMF

Example: Combined Host and Application Monitoring



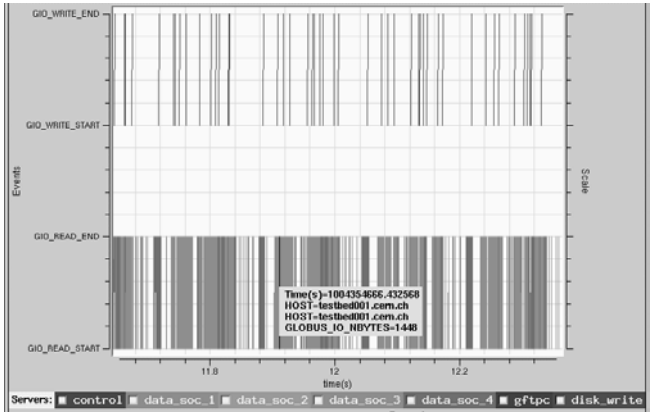
DMF

NetLogger Analysis: GridFTP results (globus_url_copy)



DMF

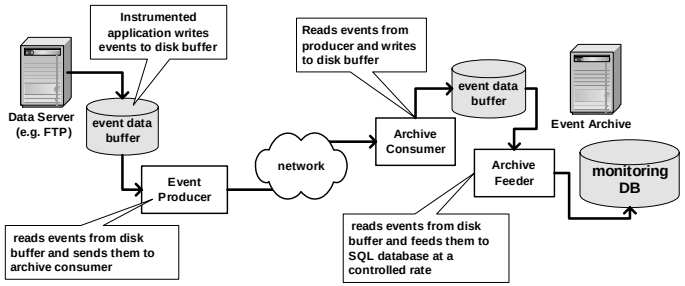
NetLogger Analysis: GridFTP results (globus_url_copy)



DMF Enhancements to NetLogger

- Rewrite of client library
  - Multiple log formats allowed with same API
    - ASCII (ULM)
    - Binary log format
      - **much better performance than ASCII**
      - **strong type information on the wire**
  - Other language APIs automatically generated with SWIG
    - Much faster than “100% native” implementations, esp. for script languages such as Perl, Python, and TCL
    - Changes and bug fixes in core automatically propagated to all APIs
- SOAP/WSDL descriptions
- Enhanced reliability
  - periodically try to reconnect broken TCP pipe
  - stores data on local disk while net is down

Archive Architecture



Requirements for FTP Server monitoring:

- 6200 events/sec
- 313 KBytes/second, (1.1 GBytes / hour)

PPDG Interactions

- Information and Monitoring Service Requirements document
- Monitoring event definitions
  - ensure schemas meet PPDG’s needs
    - Part of new GGF effort
  - Interface with EU DataGrid to ensure interoperability
- Application instrumentation
  - NetLogger support
- Event Archive
  - Deploy and evaluate prototype archive system

For More Information

DMF: http://www-didc.lbl.gov/DMF/

GMA: http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

email: bltierney@lbl.gov