GGF-DAMED-01-I
Category: Informational
Discovery and Monitoring Event Descriptions WG

Dan Gunter, Lawrence Berkeley National Lab
James Magowan, IBM UK Ltd

April 30, 2003
Revised : January 9, 2004

An analysis of "Top N" Event Descriptions

<u>Status of This Memo</u>

This memo provides information to the Grid community. It does not define any standards or technical recommendations. Distribution is unlimited.

<u>Copyright Notice</u>

Abstract

We present a methodology for defining, in a schema-independent way, the smallest useful stand-alone units of information found in any Grid schema. We apply this methodology to a set of commonly used monitoring measurements, called the "Top N Events".

<u>Contents</u>

1. Introduction

Throughout the Grid community groups have their own technologies for representing schemas. Up to this point, the main sources of common schemas were those shipped with Grid middleware, such as the Globus Toolkit. Efforts have been underway to produce uniform schemas to allow the sharing of information across current Grid boundaries. Common schemas have value for tools and middleware, but for information exchange often all that is needed are common names and units for the actual values being communicated. This document describes how this can be achieved in the context of a small set of commonly used data – the “Top N”.

This document does not attempt to define a uniform schema for any part of the Grid. Instead, it presents a methodology for defining, in a schema-independent way, the smallest useful stand-alone units of information found in any Grid schema. If the schema were represented as a graph, these units of information would be nodes that had no outgoing edges, i.e. the “leaves” of the graph. In LDAP (Lightweight Directory Access Protocol) terms, these are the attributes of the schema. Following the terminology used in the Grid Monitoring Architecture, we call these units of information “events”.

The key realization that spurred this approach was that, although the structure of relationships between abstractions such as “queue”, “storage system”, and “virtual network path” often varies between schemas, the events are very similar. These events have already been standardized, often by the code inside the measurement tools, to produce comparable measurements even on disparate components. For example, although there are many different ways of measuring latency on a network, most schemas will report a single number for the “delay” on a link.

A second realization was that increasing detail in components and measurement methods could be coerced into a hierarchy. In order to reduce the number of “leaves”, Grid schemas must at some point ignore differences between components. Unfortunately, this necessary optimization causes loss of information – we may no longer know whether the link delay was measured with TCP, or UDP. But by making “delay” the parent of “delay.TCP” and “delay.UDP”, this information can be preserved. Components can use the lowest level of the hierarchy that they can recognize. In object-oriented terms, they can work with the parent or grandparent (etc.) class. Transparent extensibility becomes possible: “delay.TCP” may be replaced with “delay.TCP.experimental” with no effect on information consumers.

There should be some way within the Grid to efficiently exchange events without the need for translators at boundaries to convert from one schema into another. A particular schema may contain tens or thousands of individual pieces of information, most of which change independently. When one of those pieces of information changes, we would like to send just that information. If we have a stand-alone definition of that piece of information that is shared by both schemas, then we can exchange the information directly and still share events with ease.

2. Terminology

This section provides a definition of terms used in this document.

Event	the smallest stand-alone unit of measurement information
Event type	a type of event, e.g. “delay.TCP”
Target	the entity being measured in an event
Target type	a type of target, e.g. “network.link”
Event name	a concatenation of target type and event type, e.g. “network.link.delay.TCP”

3. "Top N" Events

The events selected for the "Top N" had to be in common use, well-understood, and essential for characterizing a Grid component. Beyond that, we tried to pick events that represented the important pieces of hardware, and were a basic set for interoperability. But of course such a small subset of possible events is more or less arbitrary; it would be most accurate to say that we picked ones we considered the most useful for our own work.

Our view of the "Top N" event types are listed below, with descriptions.

1. **CPU Load** – fraction of the CPU that is not idle
2. **System uptime** – The average, over the last minute, of the amount of time that the processor was not idle.
3. **Disk size** – total space on a disk
4. **Disk used** – space on a disk that is allocated
5. **TCP available bandwidth** – Maximum available single-stream TCP bandwidth.
6. **Ping RTT** – round trip time on a network link
7. **TraceRoute number of hops** – Number of hops between source and destination on a network link
8. **Running software status** – current status, e.g., "stopped"
9. **Packet Loss** – one-way packet loss
10. **Available memory** – memory not allocated
11. **Queue length** – scheduling queue length
12. **Host architecture** – name of host architecture
13. **Host OS** – name and version of host operating system
14. **Physical memory** – total memory on a host

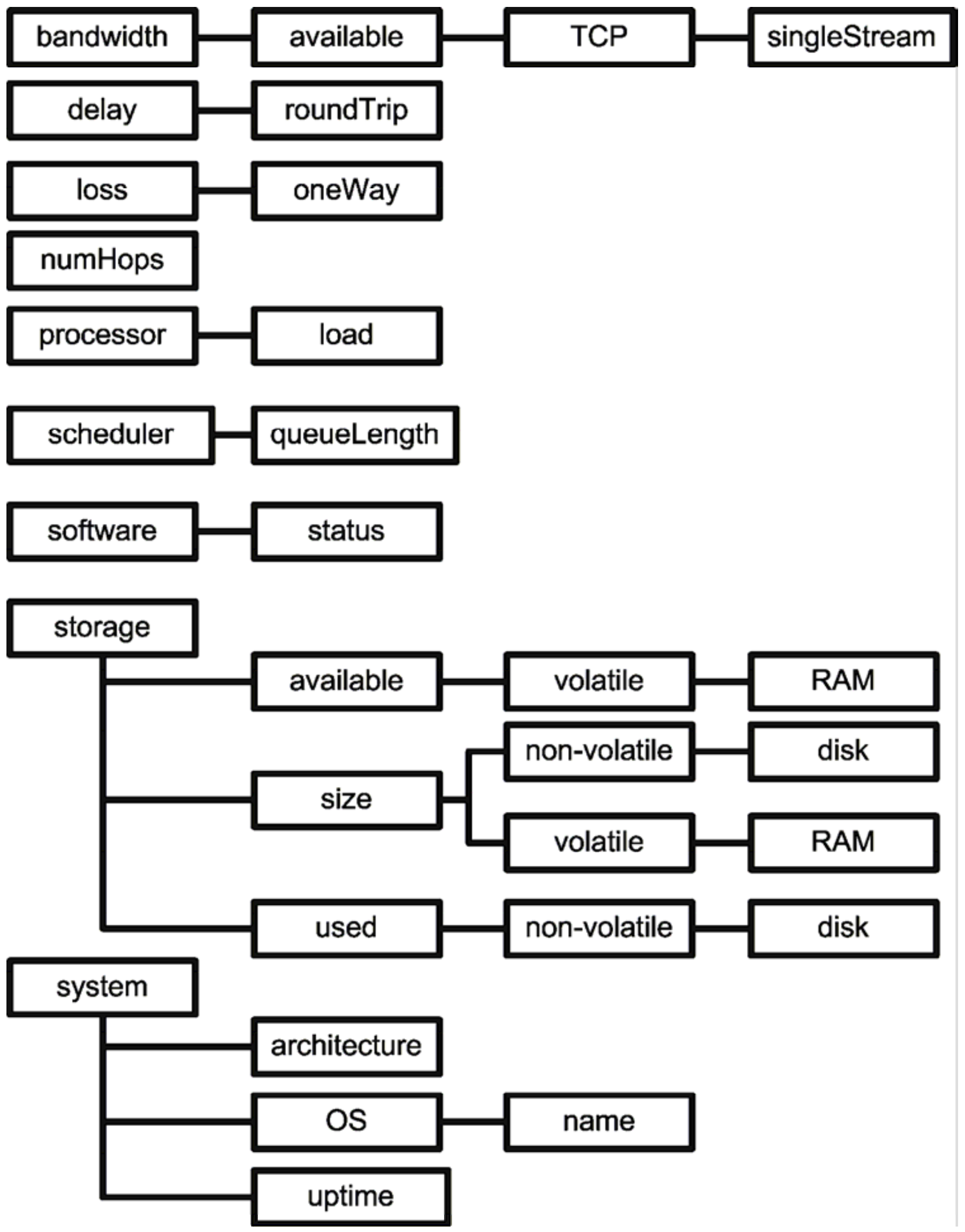

Figure 1: Representation of Top N Event Hierarchy

3.1 Event Type Hierarchy

The "Top N" event types were arranged in a hierarchy, shown below in Figure 1. Some thought was given to extensibility and generality, but where possible the guiding principle was simplicity. As is always the case with a taxonomy, selection and ordering of categories is somewhat arbitrary. For example, should the order be (storage, size, volatile) or (storage, volatile, size)? In general we have tried to arrange the hierarchy so that non-leaves could also have meaningful measurements. For the previous example, (storage, size) seemed more likely to have a useful value than (storage, volatile). The intent here was not to invent a single hierarchy, and to emphasize this we have purposely not rooted this hierarchy, although it could be trivially rooted under something like "Grid". Where possible, names have been borrowed from IETF [1] (Internet

Engineering Task Force) and IPPM [2] (IP Performance Metrics) documents. The names are all in "camelBack"-case, i.e. they start with a lowercase letter and use an uppercase letter to indicate the start of a new word in the case of a multi-word name.

3.2 Event Type Schema

Event types have a simple structure. Associated with each event type are a name, target type, value datatype, units, and description. The name is formed by concatenating the names from each level of the event type hierarchy with a '.'; for example, "bandwidth.available.TCP.singleStream". The value datatype can be anything, but at least boolean, int, float, string, IPv4 address, and IPv6 address. The units are a short string (e.g., less than 256 chars) such as "Mb/s" or "bytes". The description is a potentially longer string that describes how and what is being represented by the value, possibly with embedded URI's to relevant background material.

In the table below, we show the target type, datatype, and default units for each of the Top N event types.

Event Name	Target Type	Datatype	Units
bandwidth.available.TCP.singleStream	network link	float	Mb/s
delay.roundtrip	network link	float	percent
loss.oneWay	network link	float	ms
numHops	network link	integer	-
processor.load	host	integer	percent
scheduler.queueLength	scheduler	integer	-
software.status	software	enumeration	-
storage.available.volatile.RAM	host	integer	MB
storage.size.nonVolatile.disk	disk partition	integer	MB
storage.size.volatile.RAM	host	integer	MB
storage.used.nonVolatile.disk	disk partition	integer	MB
system.architecture	host	string	-
system.OS.name	host	string	-
system.uptime	host	integer	seconds

3.3 Event Schema

Each instance of an event has an event type, target, value, and timestamp. Event types have been described in 3.2 and targets will be described in 4.2. Timestamps should have at least microsecond resolution and be able to represent dates for at least the next 100 years. The value should conform to the value datatype specified by the event type.

4. Targets

Associated with each instance of an event is a target. The target represents the thing being measured, as opposed to the measurement itself. Usually, targets can be associated with more than one event. For example, a network link is a target which can be measured in many different ways.

Some target types that relate to the "Top N" events are listed below. This is by no means intended to be an exhaustive list of all possible target types. It is unclear that a hierarchical naming scheme for target types will be generally useful; many existing schemas structure these elements in a more complex graph than a simple tree. Therefore, we propose that the target type name be an unstructured "unique name" of some sort. In this document, we will not put forth any technology for this, and instead just use plain English.

1. host
2. process
3. disk partition
4. network link
5. software
6. scheduler

4.1 Target Type Schema

Target types have a simple schema: the name of the target type, an English description, and the identifier type. The target type name is a "camelBack"-case word or phrase using only uppercase and lowercase letters. The English description is a free-form string. The identifier type is a list of the attributes needed to identify a given target (later, more formal and powerful mechanisms, such as XML-Schema, may be introduced for this purpose).

The target types required for the "Top N" events are summarized in the table below.

Target type	Description	Identifier Type
host	A single host	IP address
process	A process running on a host	IP address, Process ID (PID)
disk partition	A partition of a disk	disk device, partition name
network link	A link between two network endpoints	src IP address, src port, dest IP address, dest port
software	An application, middleware, etc.	free-form string
scheduler	A Grid scheduler	IP address, scheduler type, queue name

4.2 Target Schema

A target instance has a target type and identifier. The target type is described above. The identifier should conform to the identifier type in the corresponding target type. Sample targets for each target type are shown in the table below.

Target type	Identifier
host	129.42.17.99
process	129.42.17.99, 15997
disk partition	/dev/sda1, /tmp
network link	129.42.17.99, 32478, 131.243.2.11, 80
software	'Gnu Emacs 20.7.9'
scheduler	129.49.17.199, FooScheduler

5. Examples

A sample event instance for each event type is given. Timestamps are not shown, but would in practice accompany each event instance. The meaning of each column is described below:

- Event type – Type of the event

- Target – A target instance
- Value – The value for the event instance.

Event Name	Target	Value
bandwidth.available.TCP.singleStream	129.42.17.99, 140.221.9.95	134.56
delay.roundtrip	129.42.17.99, 140.221.9.95	23.5
loss.oneWay	129.42.17.99, 140.221.9.95:8199	0.07
numHops	129.42.17.99, 140.221.9.95	10
processor.load	129.42.17.99	12
scheduler.queueLength	129.42.17.99, FooScheduler, short	3
software.status	'Gnu Emacs 20.7.9'	running
storage.available.volatile.RAM	129.42.17.99	400
storage.size.nonVolatile.disk	129.42.17.99,/dev/sda1	12786
storage.size.volatile.RAM	129.42.17.99	512
storage.used.nonVolatile.disk	129.42.17.99,/dev/sda1	3086
system.architecture	129.42.17.99	'1 x Intel Pentium III'
system.OS.name	129.42.17.99	'Linux 2.4.7-10'
system.uptime	129.42.17.99	1314

6. Unresolved Issues

As schemas change so to could our consideration of the default units for a particular event. One solution is to always encode the units along with the event data. Another is to override the default units on a per-schema basis.

Interactions with various schema technologies and concrete schemas are not yet fully tested. In particular, the relationship to SQL, XML, and CIM (Common Information Model) schemas needs to be explored.

7. Security Considerations

Security issues are not discussed in this document.

Author Information

Dan Gunter
Lawrence Berkeley National Laboratory
M/S 50B-2239, 1 Cyclotron Road
Berkeley, CA 94720
dkgunter@lbl.gov

James Magowan
Mail Point 137,
Dynamic e-Business,
Technology and e-Solutions,
IBM Hursley, Winchester, Hants, UK, SO21 2JN
magowan@uk.ibm.com

Glossary

Event	the smallest stand-alone unit of measurement information
Event type	a type of event, e.g. “delay.TCP”
Target	the entity being measured in an event
Target type	a type of target, e.g. “network.link”
Event name	a concatenation of target type and event type, e.g. “network.link.delay.TCP”

References

1. RFC 1514, Host MIB. http://www.ietf.org/rfc/rfc1514.txt
2. IPPM Measurement MIB. http://www.ietf.org/internet-drafts/draft-stephan-ippm-mib-02.txt

Trademarks