TANGO V8
ANOTHER TURBO CHARGED MAJOR RELEASE

• What is Tango
• Event System
• Java Device Servers
• Tango Mobile
• Embedding Tango
• Tango V9
What is Tango
What is Tango

• A software bus for distributed objects

Linux, Windows  
Java, C++, Python
What is Tango

- Graphical interface and state machine design
- Code generation: C++, Java and Python
- Editing and code re-generation
- Fast development cycle
What is Tango

• GUI frameworks for C++, Python and Java
• Synoptic editor
What is Tango

• Administration and survey system
• Graphical system configuration
Event System

• Was based on CORBA notification service
  • External notification daemon running on each host
  • Two network hops
• Today based on ØMQ: C++ ØMQ library and jzmq
  • Integrated in the device server process
  • Based on the publisher-subscriber pattern of ØMQ
  • Event multicasting is possible
    • Needs to be configured
    • Complex network set-up: multi casting addresses
    • Pragmatic General Multicast (PGM) protocol of the ØMQ library
• Major effort for compatibility
  • Servers and clients can handle both event systems
Event System Performance

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Event Rate</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 double = 64 bits</td>
<td>100 KHz</td>
<td>250 us</td>
</tr>
<tr>
<td>1 KByte</td>
<td>85 KHz</td>
<td>300 us</td>
</tr>
<tr>
<td>1 MByte</td>
<td>1 KHz</td>
<td>2 ms</td>
</tr>
</tbody>
</table>

• Improvement
  • A factor 30 for events transferring only a few bytes
    • ZMQ event grouping into network packages
  • A factor of 10 for events transferring > 100 Kbytes

Intel Xeon 2Ghz
Java Device Servers

- Java device server API was re-written by Soleil
- Includes now all features of the C++ API
- Validated equivalence of the Java API with the C++ API
  - The C++ tests suite was executed on a Java test server
- Annotations are a key driver when it comes to create a Tango device
- Code is clearly focused and simplified
  - No knowledge of inherited methods needed
Java Device Servers

• The ‘@Device’ annotation defines the class ‘TestDevice’ as a Tango class
• The ‘@Attribute’ annotation defines the class field ‘myAttribute’ as a Tango Attribute

```java
@Device
public class TestDevice {

    @Attribute
    public double myAttribute;

    public double getMyAttribute () {
        return myAttribute;
    }

    public void setMyAttribute (double SetValue) {
        myAttribute = SetValue;
    }
}
```
Tango Mobile

• Three different solutions available:
  1. The Tango Java client API was ported to Android for local applications
  2. Applications for iOS and Android using the Cordova (PhoneGap) framework

Allows the writing of java script applications which connect to Tango device servers via proxy servlets on a TomCat server
Tango Mobile

• Three different solutions available:
  3. A browser based web solution, which uses the web interface of the Taurus GUI framework

  Allows writing java script applications which communicate via web sockets with a Tornado web server
Embedding Tango

Low cost fully fledged ARM based computers running Linux

Raspberry Pi
Very low cost

Beagle Bone
Used for projects at ELETTRA and the ESRF
Tango Version 9

• Data Pipes:
  • A third communication type between clients and servers on top of commands and attributes
  • Data blobs can be transferred
    • A variable set of data composed out of basic data types like a C-structure
    • The composition might change with every data transfer
    • Self-describing data
  • The usage will be the transfer of synchronized sets of data
    • For example scan results
Tango Version 9

• Forwarded (Relay) Attributes:
  • The same attribute, or data value, might be used by several Tango classes
  • In a second class we want to instantiate automatically an attribute with the same interface
  • Forwarding of all read and write requests to the source attribute
  • No manual coding will be required
    • The code generator Pogo can produce the necessary code
Thank you

http://www.tango-controls.org