

Java GAT overview and status

GridLab /
Vrije Universiteit Amsterdam

Rob van Nieuwpoort

GAT Design Philosophy

- The GAT API is an object oriented specification, but language independent.
 - Tension between existing features in the language and the GAT specification.
 - Programmers do not want to learn new APIs for features already in the language.
 - GAT should “feel” the same across languages. If a programmer knows the C GAT, he/she can use the Java GAT.
 - Find the right tradeoff.
-
-

Why Java?

- Java is widely used, object oriented.
 - Java is “write once, run everywhere”.
 - Compile application on your desktop machine.
 - This creates machine independent bytecode.
 - Copy application files and GAT to any grid site. (portal does this for you)
 - Just run it. No recompiling / configuration.
 - Performance with current JITs is good.
 - Typically 10% slower than C.
 - Ideal for grid programming ?
-
-

Target user community

- Triana
 - programming environment for data-flow applications (gravitational wave detection).
 - GridLab Portal
 - user interface to the GridLab software. Run and monitor your applications on the grid.
 - Virtual Laboratories in E-science (VLE)
 - Dutch grid project.
 - You ?
-
-

Java Application

Java GAT API

file Resource management monitoring ...

GAT Engine

Java GAT CPI

file Resource management monitoring ...

Local
file

GridLab
Remote
file

FTP
Remote
file

GridLab
GRMS

GridLab
Mercury

NWS



Java GAT Implementation

- Adaptors are Java JAR files, loaded dynamically into the application.
- Memory management is done for you (GC).
- No status object, but exceptions (nested).
- Late binding:
 - The GAT engine selects the best adaptor for each method: Create GAT File object.
 - Copy file from site A to site B and C
 - A -> B copy with GridFTP
 - A -> C copy with FTP

```
package file;
```

```
import java.net.URI;  
import org.gridlab.gat.*;
```

```
class RemoteCopy {  
    public static void main(String[] args) {  
        try {  
            URI src = new URI("file://fs0.das2.cs.vu.nl/bin/echo");  
            URI dest = new URI("file://litchi.zib.de/hiha.dat");  
  
            GATContext context = new GATContext();  
  
            File file = GAT.createFile(context, src);  
            file.copy(dest);  
  
        } catch (Exception e) {  
            System.err.println("Could not copy file: " + e);  
            System.exit(1);  
        }  
    }  
}
```

Status

- Defined API and CPI (Adaptor interface).
 - Wrote GAT engine.
 - Started working on adaptors.
 - File / LogicalFile API.
 - Local adaptors.
 - Remote adaptors using the GridLab data services.
 - Resource management API.
 - Local adaptor.
 - Remote adaptor for GridLab GRMS.
-
-

Future Work

- Better error tracing and logging.
 - More GridLab adaptors.
 - Monitoring, advert service, etc.
 - Adaptors for Java Cog.
 - Test suite.
 - Applications.
-
-