Java and Real Time Storage Applications

Gary Mueller
Janet Borzuchowski
Flavors of Java for Embedded Systems

- Software Java Virtual Machine (JVM)
- Compiled Java
- Hardware Java Virtual Machine
- Java Virtual Machine as the RTOS
Software JVMs and RTOS

• Software JVM runs as a non-critical, non-real time, low priority task
  – Hardware configuration
  – Maintenance and diagnostics
  – Code upgrades and loads

• Java threads including garbage collection can be executed at a low priority
WindRiver® Personal Jworks™

- Software JVM runs as a set of tasks on VxWorks®
- Does not provide real-time response
- Garbage collection and other Java tasks can be executed at a lower priority than other time critical tasks.
  - Retains determinism of VxWorks®
Compiled Java

- Java is compiled into native machine code
- Garbage collection is implemented through runtime libraries
- Provides the benefit of an object oriented language without the performance penalty of an interpreted language
WindRiver® Diab™ FastJ®

- Compiles C, C++ and Java to native machine code
- Configure core libraries to reduce code size
- Memory management options
  - Explicit memory management
    - Similar to C/C++, eliminates garbage collection
  - Standard, non-incremental garbage collection
    - When memory is low or explicitly called
  - Preemptive, incremental garbage collection
    - Runs as preemptable, low priority background task
Hardware JVM

- JVM is implemented in silicon.
- Ultimate in speed and performance
- Implemented as either a co-processor or separate processor on a custom chip
ARM® Jazelle™

- Hardware JVM for ARM® family of processors
- Executes both Java byte codes and ARM® machine codes
- Special instruction to enter Java state and execute Java byte codes
JVM is the RTOS

- JVM is the RTOS
- Combines the JVM and an operating system into a single entity
Esmertec JBed™

- Entire application including device drivers can be written in Java
- Bytecode is translated to machine code prior to downloading or upon class loading
Why Java for Embedded Systems?

• Java is an object oriented language
  – All the advantages of object technology
  – Enforces object concepts

• Space, performance and deterministic problems are being solved

• New software engineers are trained in Java

• Faster time to market
Java and Embedded Systems

• Space
  – Will the JVM and class libraries fit?

• Performance
  – Can the JVM run fast enough to meet hard real time deadlines?

• Deterministic
  – Is the JVM deterministic?
  – Can garbage collection be scheduled?
Contact Information

Gary Mueller
garymueller@qwest.net
303-465-4279

Janet Borzuchowski
janet_borzuchowski@stortek.com
303-673-8297
Software Java Virtual Machine

- *Classic* use of Java
- Most desktop applications execute Java using a JVM running as a process or task on the desktop
- Browsers execute Java within the browser with a JVM
Personal Jworks™ Architecture

- **PersonalJava 3.1**
  - Core Libraries

- **Lightweight JDK 1.1.8AWT**

- **Supporting Native Libraries**

- **WindML 2.0**

- **VxWorks 5.4**

- **BSP/Device Drivers**
Gnu Compiler for Java™ (gcj)

• Compiled applications are linked with the gcj runtime library, libgcj
  – Java source code to native machine code
  – Java source code to Java bytecode
  – Java bytecode to native machine code

• Requires port of libgcj library
  – Core classes
  – Garbage collector
  – Bytecode interpreter
Jazelle™ Run-Time Architecture

Class Libraries
- Network
- Graphics
- Remote Methods
- Native Methods

Java VM
- Verifier
- Class Loader
- Garbage Collector
- Process Manager
- Memory Manager

Jazelle™ Software Kernel

Native Application

Native Operating System

Jazelle™ Enabled ARM® Processor

Trusted Interface
Jbed™ Run-Time Architecture

Java Applications

- `com.jbed.*`
- `java.*`
- `javac.*`
- `javax.microedition.*`

- Http Protocol
- Tftp Protocol
- TBCC
- GC
- Log
- Debug Agent

Net Drivers

Kernel/Run-time

I/O Drivers

Network Devices

Base Hardware

I/O Devices
On the Java Road

• Introducing C++ and Object design and analysis to time critical code
• Evaluating FastJ™
• Exploring Jbed™ option
• Discussing Jazelle™ with ARM®
  – Negotiating a non-disclosure agreement
Steps to Java

• First Step --- FastJ™
  – Similar to current development environment
  – Least disruptive
  – Does not require hardware changes

• Next step --- Jbed™
  – Combination of hardware/software

• Final Step --- Jazelle™
  – Requires hardware changes
Our Environment

- 9840 and 9940 family of tape drives
- ARM7® 32 bit processor
- 2-4MB of RAM
  - Code image
- 32MB - 64MB data buffer
- SCSI, ESCON, and Fibre Channel Interface
- Specialized ASICs
State of Java and Embedded Systems

• Resurgence in Java chips
  – ARM - England
  – Ajile - United States
  – Vulcan Machines Ltd - England
  – NTT Docomo - Japan

• Personal Digital Assistants

• Next generation mobile phones
Java Runtime Environment

• Java is an interpreted language
• Java is compiled to an intermediate language
  – Java Byte Codes
  – Assembly language for the Java Virtual Machine
• Java Virtual Machine executes Java Byte code
• JVM are usually written in C or C++