Welcome

- The tutorial will
  - describe the design and implementation issues made in the Jalapeño research virtual machine for Java
- Intended audience
  - those interested in using Jalapeño as a research infrastructure
  - those interested in building a VM for Java
- Assumptions
  - basic knowledge of Java
- Tutorial is not
  - an overview of Java
  - a thorough survey of research JVMs

* Java is a registered trademark of Sun Microsystems

Tutorial Outline

- Java and Jalapeño overview
- Basic Jalapeño details
  - Objects, method dispatch, stack layout, etc.
  - Writing in Java
  - Exception handling
  - Dynamic type checking
  - Interface invocation
  - Threading and Synchronization
  - Memory management
- Optimization
  - Adaptive optimization system
  - Optimizing compiler
- DejaVu/jdp debugger

Jalapeño Team

Adaptive system, Optimizing compiler, DejaVu
Matthew Arrend (co-op)
Jang-Deok Choi (project lead)
Julian Dubb
Stephen Fisk
David Grove
Michael Hind (manager)

Running, Baseline & Quick Compiler
Bowen Alpern
Dick Attanasio
Marks Barlis
Anthony Cauchi
Mark Bergner (manager)
Tim Ng
Stephen Smith

Other contributors (past & present)

- David Bacon
- John Barton
- Matthew Barnette (academic visitor)
- Rajesh Bansal
- Steven Boldick (academic visitor)
- Michael Barut
- John Cavallaro (co-op)
- Craig Chambers (academic visitor)
- Penny Cheng (co-op)
- Brian Cooper (co-op)
- Tracey Ferguson (co-op)
- Andrew Sigo (co-op)
- Susan Farnham
- Gwang-Hee Hwang (co-op)
- Han Lee (co-op)
- Bernad Liskner
- Yuvalit Lifshitz (co-op)
- Alan Loganov (co-op)
- Jon Massen (co-op)
- Kurt Lewison (academic visitor)
- Sam Mabbott (co-op)
-EDITORS
e. n. a. (academic visitor)
- Anne Haberman (co-op)
- Tony Cline (co-op)
- Howard Leinwand
- James Shapland
- Abigail Shapland
- Andrew Shapland
- V. E. Woodson
- John Whitaker
- John Widler (co-op)
- Hyun-Gyu Yoo (co-op)

- EDITORS
- Mohammad Ali
- Guha Srinivasan
- Brian Cooper (co-op)
- Tracey Ferguson (co-op)
- Andrew Sigo (co-op)
- Steven Boldick (academic visitor)
- Rajesh Bansal
- Matthew Barnette (academic visitor)
- Steven Boldick (academic visitor)
- Michael Barut
- John Cavallaro (co-op)
- Craig Chambers (academic visitor)
- Penny Cheng (co-op)
- Brian Cooper (co-op)
- Tracey Ferguson (co-op)
- Andrew Sigo (co-op)
- EDITORS
e. n. a. (academic visitor)
- Anne Haberman (co-op)
- Tony Cline (co-op)
- Howard Leinwand
- James Shapland
- Abigail Shapland
- Andrew Shapland
- V. E. Woodson
- John Whitaker
- John Widler (co-op)
- Hyun-Gyu Yoo (co-op)

www.research.ibm.com/jalapeno

Adaptive system, Optimizing compiler, DejaVu
Matthew Arrend (co-op)
Jang-Deok Choi (project lead)
Julian Dubb
Stephen Fisk
David Grove
Michael Hind (manager)
What is Jalapeño?

- Cleanroom research VM for Java implementation
  - Libraries from OTI (Visual Age for Java, Micro Edition)
- Implemented in Java (~300KLOC)
  - Reduces seams between JVM and applications
  - JVM can be dynamically optimized
- Targeted for servers, compile-only strategy
  - Multiple compilers, mixing code is seamless
- Aggressive optimizing compiler
  - 3 levels of IR, all with Java type info (CFG, SSA, dominators, etc.)
  - Multiple optimization levels
- Lightweight thread implementation
  - Java threads are multiplexed on OS threads, important for scalability, GC transition
  - Quasi-preemptive scheduling (using compiler-generated yield points)
- Adaptive optimization system
  - Yieldpoint-based sampling, cost/benefit model, what to recompile and what opt. level
  - Online feedback-directed inlining
- Type-accurate (exact) parallel GC/Allocation
  - Copying and noncopying, generational and nongenerational, hybrids, concurrent ref-counting

Jalapeño History

Goal: Provide an infrastructure for OO language/execution and Java middleware research

- Nov ’97: project initiated
- Mar ’98: opt compiler
- Jan ’00: adaptive system
- Feb ’00: Linux/IA32 port
- World-class performance
  - Dec ’99: scalability
  - Dec ’00: SPECjvm98

University releases
- Jan ’01: AIX/RS6000, VM, baseline and opt compiler
  - PPC/Linux courtesy of U Mass
- Apr ’01: Basic Linux/IA32 support, adaptive system
- Oct ’01: improved functionality/performance on Linux/IA32

Why use Jalapeño as a Research Infrastructure?

- Written in Java
- Designed for research and experimentation, clear structure, extensible, modular (GCs, compilers, etc.)
  - Flexible adaptive optimization architecture
  - Full-fledged optimizing compiler infrastructure
- State-of-the-art research implementation
  - Credible research results
  - Competitive performance with top commercial systems
- Established user community
  - Jalapeño-researchers mailing list
- Adequate documentation (user guide, javadoc api)

Why not use Jalapeño as a Research Infrastructure?

- Incomplete functionality
  - Missing library support (AWT, etc.), user-defined class loaders, security manager, verification, ...
- Possible JNI/threading performance issues
- Research project robustness
  - No testing organization
  - But, nightly regression tests
- IA/32 backend optimizer not fine-tuned
- Only adequate documentation (user guide, javadoc api)
  - No technical writing staff
What universities are currently using Jalapeño?

- Source licenses with 14 universities
  - UMass (Eliot Moss, Steve Blackburn)
  - UT-Austin (Kathryn McKinley)
  - Bern (Paolo Castano)
  - Jena (Wolfram Amme)
  - Wisconsin (Ras Bodil)
  - Purdue (Tony Hosking)
  - Colorado (Amir Dian)
  - Rutgers (Barbara Ryden)
  - Kent (Richard Jones)
  - Illinois at Urbana-Champaign (David Padua)
  - Michigan State University (Jaejin Lee)
  - U of New Mexico (Danka Stefanovic)
  - Utah (Wilson Hsia)
  - Appalachian State (Jay Fenwick, Cindy Norris)
  - UCIrvine (Michael Franz)
- Pending: Rice, UCSB, McGill, Maryland, Edinburgh, Technion, Georgia Tech
- UMass serves as distribution site

What are the current research uses?

- Variety of research activities
  - memory mgmt
  - optimization
  - language/architecture interface
  - real time issues
  - mobile code issues
- ... (add your favorite topic here)
- Publications in PLDI, OOPSLA, ECOOP, SIGMetrics, Java Grande, etc.

"High-Performance Java Codes for Computational Fluid Dynamics" by Riley, Chatterjee, and Biswas (ACM Java Grande '01)

```
Langley Aerothermodynamic Upwind Relaxation Algorithm
4-way Pentium and 12-way PowerPC
```

![Graph showing speedup for different environments]

"High-Performance Java Codes for Computational Fluid Dynamics" by Riley, Chatterjee, and Biswas (ACM Java Grande '01)

```
Two-Dimensional Triangular Adaptive Grid
(Object-oriented version)
```

![Bar chart showing degradation relative to C]

Section 1 - Intro