

Modularity, Hardware Profiling, and Mixed ISA Execution in Managed Runtimes

Suresh Srinivas
Software & Solutions Group (SSG)
Intel Corporation

Intel & Managed Runtimes (MRTE)

- **Why:**

- Usage Model: MRTE's another MIPS sucking app ;)
- Opportunity: Evolve Intel Architecture to help MRTEs
 - IA pretty good at running MRTEs already
 - Working with MRTV's provide features enabling performance, security and reliability

- **What:**

- Intel Research:
 - Research HW/SW technologies to improve MRTE performance on Intel Architectures
 - Open Runtime Platform & StarJIT Compiler
 - Advanced MRTE Optimizations
- Intel Product Group:
 - MRTV (Managed Runtime Vendor) Enabling and Optimization on Intel Architectures
 - Intel μ architectures and MRTE Performance
 - VTune MRTE Performance Analyzer

Agenda

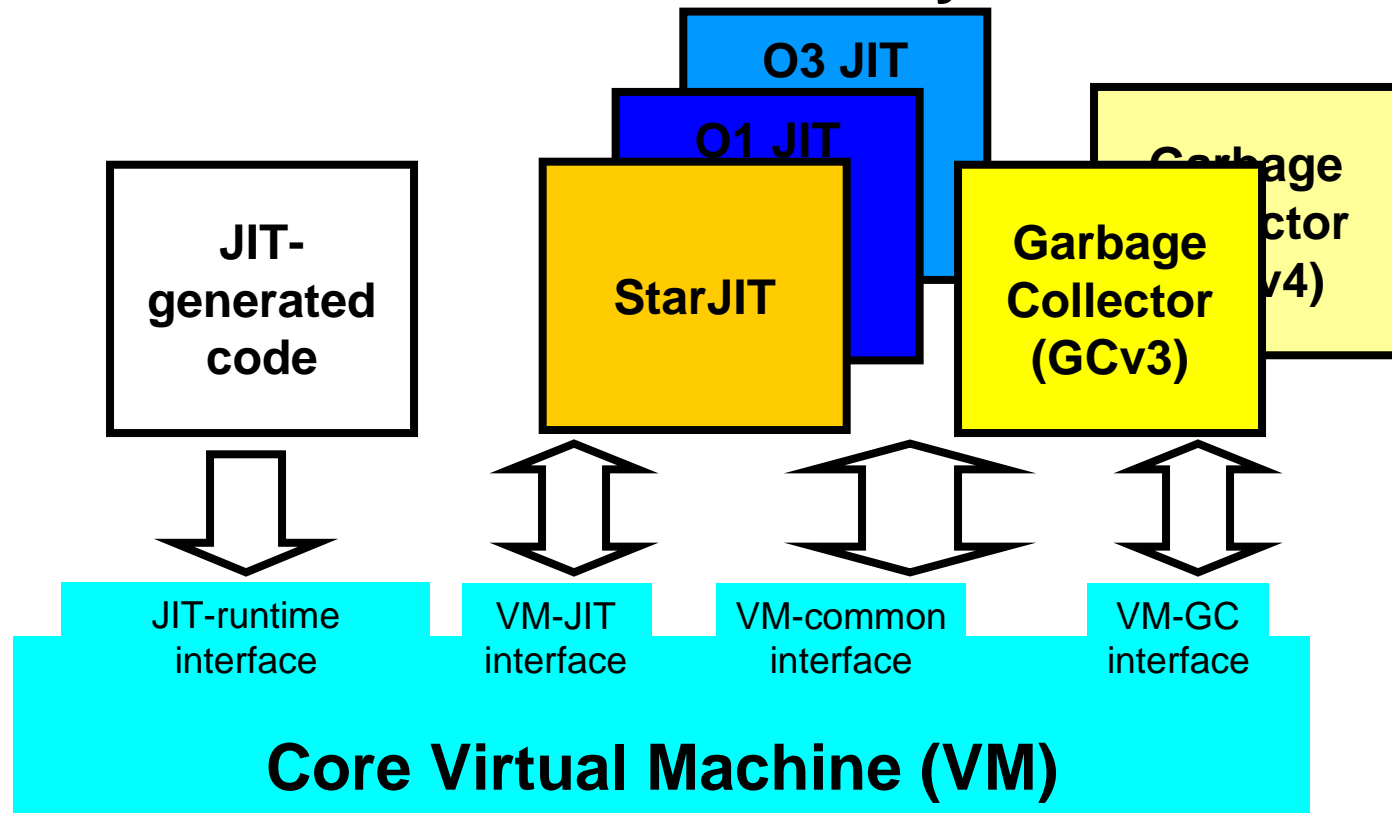
- **Modularity in Managed Runtimes**
- **HW Based DPGO in Managed Runtimes**
- **Executing Multiple ISA's within Managed Runtimes**
- **Opportunities for the future**

Modularity in Managed Runtimes

- **Issue:** Currently shipping Production Runtimes rarely provide limited publicly exposed interface to VM services (eg Profiling but not to JIT's, GC's, low-level access to HW)
- **Opportunity:**
 - If interfaces are exposed and high performance, tuned pieces can be swapped in
 - Upside is potentially large
 - Overcomes other costs such as additional validation
 - Encourages innovation and ability to differentiate
 - Example JVMTI interface allows VTune to support profiling Mixed Mode Apps
 - Smaller companies to supply components without heavy lifting
 - Research within production runtimes and richer workloads
- **Intel has been researching Modularity in Managed Runtimes for several years**

Modularity in Intel's Open Runtime Platform (ORP)

- ORP supports multiple JITs and GCs through well-defined interfaces and modularity



Hardware based DPGO in Managed Runtimes

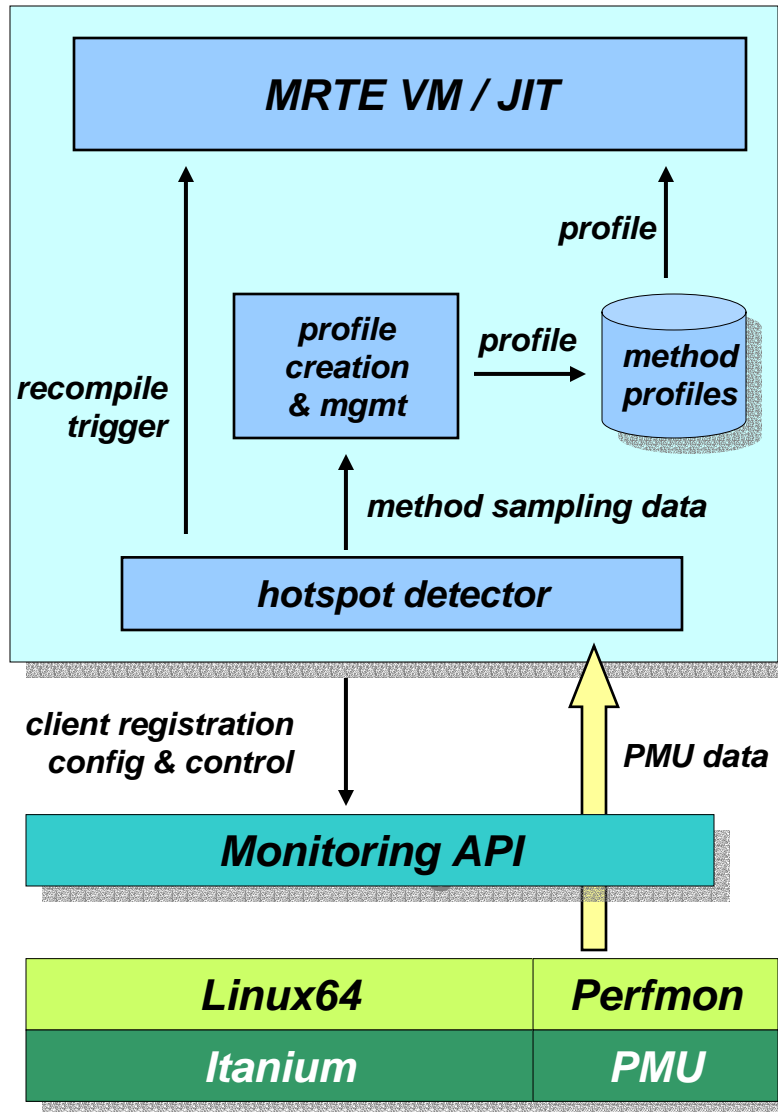
- **Issue:** Currently shipping Production Runtimes rely primarily on SW for Dynamic Profile Guided Optimization (DPGO)
- **Opportunity:**
 - HW can provide much more data at lower overheads
 - Specific loads that are missing in D-Cache
 - Regions of code that trigger I-Cache/I-TLB misses
 - History of Branches and their direction
- **Intel has been researching DPGO**
 - Within the context of Itanium Processor Family (IPF) an in-order processor which stalls on cache-misses/TLB misses

Hardware Support for DPGO

- IPF HW attributes events like branches, cache and TLB misses to individual instructions so that software can know exactly where to optimize in the program
- IPF implements three hardware structures:
 - **Branch Trace Buffer (BrTB)**
 - **Instruction Event Address Register (IEAR)**
 - **Data Event Address Register (DEAR)**

Branch Trace Buffer (BrTB)	Last 4 to 8 branches	Branch PC, Branch Target PC, Mispredict Status
Data Event Address Register (DEAR)	Last D-Cache Miss	Instruction PC, Data Address, Miss Latency in cycles

Hardware based MRTE DPGO Architecture



- **Hardware based DPGO**
 - Sampling producer/consumer API
 - HW based Hotspot detection
 - Integrated profile creation, mgmt & use
 - Edge profiles
 - Dcache miss profiles
 - Profile-to-IR mapping
 - Profile decoration API (adds profile info to IR)
 - Optimizations using profiles
- **Monitoring Sampling API**
 - Per-thread data collection
 - Sampling config & control, data buffer mgmt
- **PMU data**
 - BTrB samples
 - DEAR samples

Hardware based MRTE DPGO Optimizations

Data Side

- **Heuristic Prefetching**
 - DEAR miss sampling, delinquent load tree identification
 - Heuristic prefetching based on property of delinquent load tree.
 - Implemented in the BEA JRockit system and shows a 6-8% speedup
- **Mississippi Delta Prefetch**
 - Integrates hardware performance monitor, GC and JIT to inject prefetches
 - Abstracts hardware cache miss sampling up to metadata
 - Leverages GC to discover and maintain useful global properties
 - Evolves GC into memory hierarchy controller and optimizer
 - Implemented in high performance fully dynamic system (ORP/StarJIT) and shows a 14% speedup
 - PLDI 2004 paper

Hardware based MRTE DPGO Optimizations

Instruction-Side

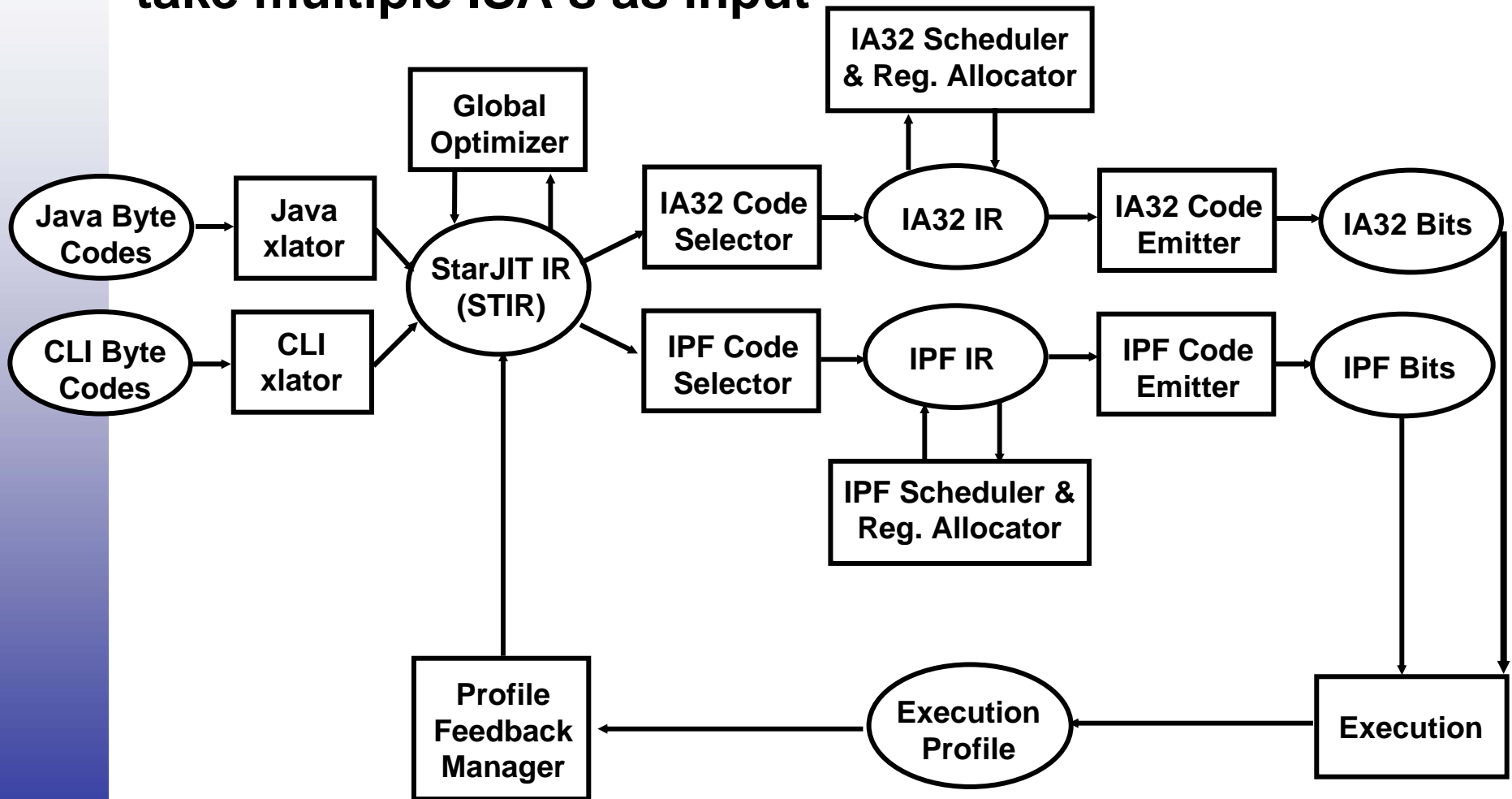
- **Edge profiling mechanisms**
 - PMU-based edge profiling
- **Profile-guided optimizations**
 - Inlining & guarded devirtualization
 - Loop peeling
 - Hot path splitting
 - Block layout & method splitting
 - Trace scheduling
- **Profile consistency assurance**
 - Checksums on CFG
 - Propagation of the profile during optimizations
 - Profile consistency check & smoothing
- **Profile-guided recompilation techniques**
 - IR rematerialization
 - Optimization phase ordering strategies

Executing Multiple ISA's in Managed Runtimes

- **Issue:** Currently shipping Production Runtimes provide execution of one virtual ISA (bytecode) interfacing to one host ISA
- **Opportunity:**
 - Interoperability of host ISA's (IA-64 and IA-32) within same process
 - Would allow mixed Mode Applications (eg Java on 32-bit JVM calling IA-32 library) to run unmodified on IPF
 - Interoperability of virtual ISA's (Java bytecode and CLR bytecode)
- **Intel has been researching Multiple ISA execution in Managed Runtimes**

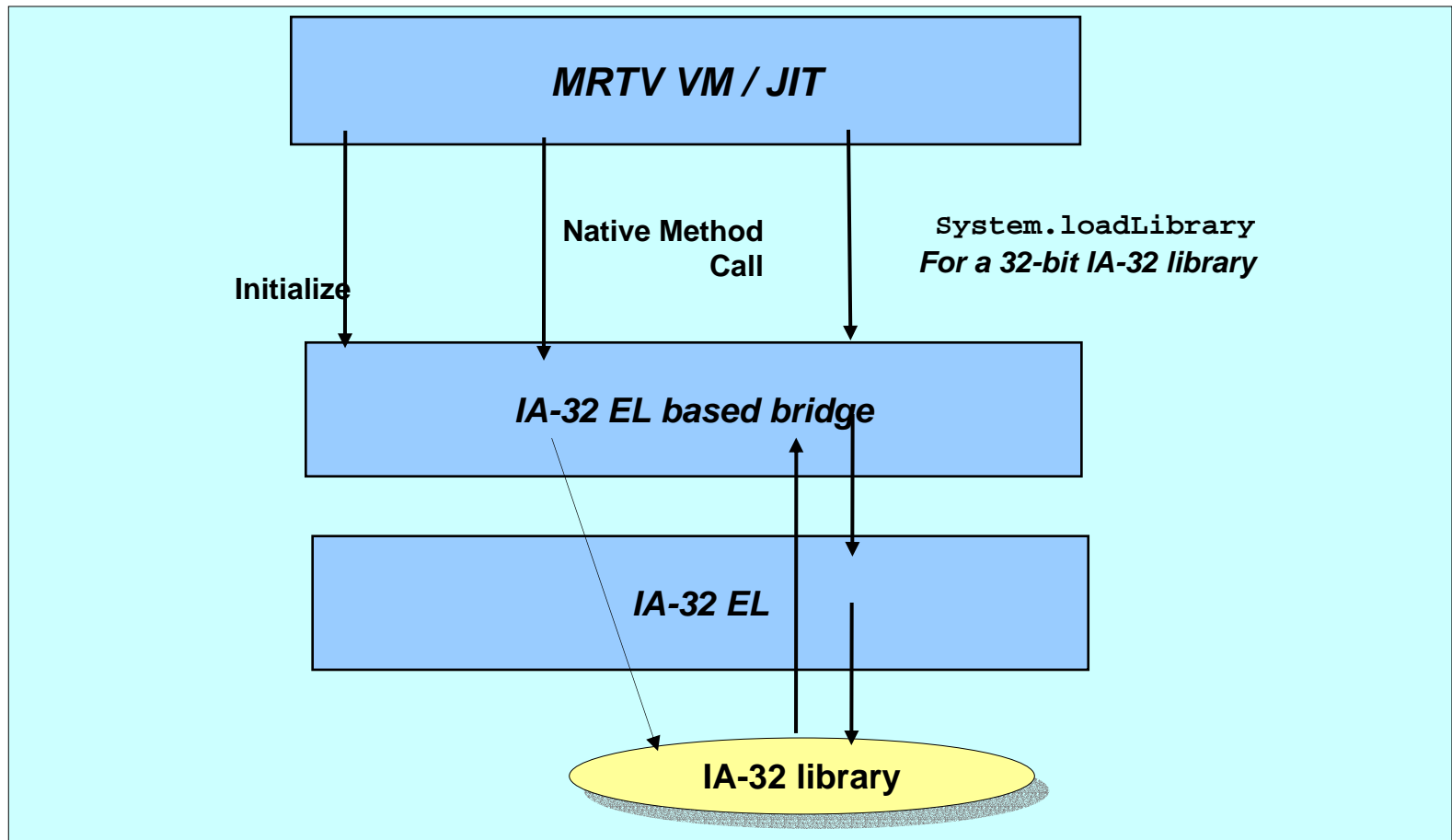
StarJIT Architecture

- StarJIT has several front-ends and is architected to take multiple ISA's as input



Loading and Executing IA-32 Libraries in 64-bit IPF MRTE

- MRTV VM loads the bridge and interfaces to it
- Bridge based on IA-32 EL



Future of Virtual Execution Environments

- **HW support for OS Virtualization now in IA**
- **HW support for MRTEs provide next boost in**
 - Performance, Security, Reliability
- **Dual-core/Multi-core coming in HW**
 - MRTE Core Libraries
 - Could be more concurrent
 - MRTE redesign to dedicate cores for certain activities