Using VEEs for Standard Business Applications

Hans-Christoph Rohland
VP Java Server Technology, SAP AG
SAP and VEEs

SAP is a long-term provider of a VEE

- R/3, mySAP and the Web Application Server all incorporate the ABAP environment
- ABAP was dedicately developed to enable business applications

SAP is also a user of standard VEEs

- SAP NetWeaver is to the greater part developed in Java
- We also integrate with .NET technology
Requirements to run business applications

Target group is not technical

- Both development and customers focus on business content
- Reproducibility and standard behavior is key for the customers
- Complexity of the scenarios drives technology needs

The Platform needs to provide the technological progress in the easiest way possible
The ABAP value proposition

Provide an environment which is

- Provides very easy access to business related functionality
  - Transactions
  - Dialog oriented
  - Very high robustness and fault tolerance
  - Hide the platform

- Disables most of the technological features
  - Strict resource control
  - No explicit parallelism – but very good scalability
  - No direct networking
The Java Value Proposition

You get a generic machine
- You get access to all resources available
- It provides fine grained operations
- It can be used from embedded devices up to application servers
- It hides the platform details
- One size fits all
Expectation

ABAP:

Physical Environment
- Preconfigured environment
- Isolation
- Efficiency
- Freedom

Specialized Virtual
- Restricted
- Easy
- Powerful operation

Java:

General VEE
- One environment for all layers
Recap:

Some observed benefits of application-level virtualization

- Application-level virtualization provides *portable, well-specified behaviour* for program execution and selected, high-level operating system interfaces.
  - Virtualization represents a factoring of the runtime into an API friendly to compilers and programmers and an implementation that can vary and be optimized for each platform

- Access to critical operating system or application resources can be controlled through VEE-enforced *security* mechanisms.

- Programmers can achieve higher levels of *productivity* through
  - VEE safety assurances
  - automated memory management
  - sophisticated analysis and debugging tools
  - automatic support for distribution and persistence

- High levels of *performance* are possible through dynamic and adaptive program transformation.
  - But we are just on the frontier of exploiting this capability
Real life technical problems

Portable and well specified
- No build in support to manage the VM instance
- Runtime behaviour differs across platforms

Security
- Per default all resources are exposed

Performance
- Concurrency is misused
- JIT is causing stability problems
- GC leads to unpredictable response times
Real life technical problems

Memory Management
- The notion of „GC takes care of it“ leads to bloat
- GC is a big problem area
- No resource monitoring
- No resource alerting

Productivity
- Tool support under par
Problems from another angle

There is no border line between core and application

- People tend to reinvent the wheel
- People overestimate their skills
  - Concurrency

Never assume some part is perfect

- The VM is a very complex beast
- It is far from perfect

Basic operating system mechanisms are missing

- Protection and sharing
- Ready to use environment
- Resource control and monitoring

Nonfunctional aspects dominate in real world scenarios!
Some ideas

Even for one size fits all we need different levels of access

- We need to get more isolation
- If we have isolation we need sharing and messaging

Nonfunctional aspects should be better specified

- Performance
- GC behaviour

Tool support

- We need to have ways to look into the innerts
- Provide access to resource control
  - Even simple alerting facilities make a tremendous difference

The VM itself is not controllable
What’s left?

One tool for all layers is still a valuable goal
- But needs strong support and governance

Dynamic extensibility is a needed feature

The rich environment is breaking boundaries

We have to make the promises reality!

→ The value proposition is still there