Workshop on the Future of Virtual Execution Environments

Wrapup and discussion

September 17, 2004

Main themes of the workshop in hindsight

- **The convergence of virtual machines and operating systems**
  - In many cases, we're solving the same problems
  - But it's not clear that we're making the right design choices ("where to draw the lines?", "how many levels of virtualization?")
  - In the absence of a technical basis for factoring functionality, it will be (has been?) decided by economics
- **The need for hardware/software co-design in virtual machines**
  - Azul and Intel both make the point that the time is right to do more serious co-design of VMs and hardware
  - This happens to coincide with the predicted microprocessor frequency scaling crisis in a few years due to power leakage and cooling limits
  - As with any such effort, we need to strive for high impact and minimally disruptive additions to hardware that will be generally usable and have a long lifetime

Main themes of the workshop in hindsight

- **Language and VM diversity is good but not if the seams show through to the developer**
  - CLR and scripting communities are paying more attention to this than Java but the real interoperability solution must include every widely used VM (even the ones that don't call themselves VMs) and support reasonably foreseeable future types of VM
  - We must nurture innovation in programming language and virtual machine design
- **Software quality depends on widespread use of virtualization and metadata**
  - Fairly strong safety and security assurances seem just around the corner if we can take a bit more care in constructing our virtual machines
  - Can extremely reliable programs be built without strong safety guarantees from the execution environment?
- **The promise and reality of virtual execution environments are far apart**
  - Practitioners can see the light at the end of the tunnel (no, not the train!)
  - But .. VEE implementations have to be much better (faster, more reliable, more predictable) and the supporting tools have to get a lot better to deliver on the promise

Main themes of the workshop in hindsight

- **The interoperability of virtual and native code is a subtle but important problem**
  - ABI differences can be a major source of performance and usability problems
  - Virtual machines need “backdoor” access to native code to assist in library design and enable high performance but we need to take care to design usable and checkable interface mechanisms
- **New opportunities for virtual machines come from surprising places**
  - Database engines
  - Massively parallel systems
  - Real time systems
- **Type-safe languages and runtimes are mainstream**
  - Open source communities all straining under the constraints/evils of C++ programming but no obvious unified runtime solution
  - The rubber is hitting the road in small devices, clients and large servers

Stuff I missed …

Next Steps

- **Fill in your feedback forms**
  - We've had a lot of feedback over the last couple of days but it helps to write it down
  - We can provide everyone a summary of the feedback
- **What about producing a report?**
  - We've noted some interesting challenges and opportunities for VEEs this week
  - Why not write them down in a report and use it to communicate with our students and colleagues?
  - If nothing else, it gives us a forum to work out some of these problems in a bit more detail in the coming weeks
- **Stay connected**
  - Our key goal for the workshop has been to bring these diverse communities together, if only for a few days
  - I hope we can keep the relationships alive .. shared goals can only be achieved through personal collaboration
  - Feel free to use IBM if its useful to make these connections for you

Thank you!