

io-port 05133839

Venstermans, Kris; Eeckhout, Lieven; De Bosschere, Koen

64-bit versus 32-bit Virtual Machines for Java.

Softw., Pract. Exper. 36, No. 1, 1-26 (2006).

Summary: The Java language is popular because of its platform independence, making it useful in a lot of technologies ranging from embedded devices to high-performance systems. The platform-independent property of Java, which is visible at the Java bytecode level, is only made possible thanks to the availability of a Virtual Machine (VM), which needs to be designed specifically for each underlying hardware platform. More specifically, the same Java bytecode should run properly on a 32-bit or a 64-bit VM. In this paper, we compare the behavioral characteristics of 32-bit and 64-bit VMs using a large set of Java benchmarks. This is done using the Jikes Research VM as well as the IBM JDK 1.4.0 production VM on a PowerPC-based IBM machine. By running the PowerPC machine in both 32-bit and 64-bit mode we are able to compare 32-bit and 64-bit VMs. We conclude that the space an object takes in the heap in 64-bit mode is 39.3% larger on average than in 32-bit mode. We identify three reasons for this: (i) the larger pointer size, (ii) the increased header and (iii) the increased alignment. The minimally required heap size is 51.1% larger on average in 64-bit than in 32-bit mode. From our experimental setup using hardware performance monitors, we observe that 64-bit computing typically results in a significantly larger number of data cache misses at all levels of the memory hierarchy. In addition, we observe that when a sufficiently large heap is available, the IBM JDK 1.4.0 VM is 1.7% slower on average in 64-bit mode than in 32-bit mode.

Keywords: Java; Virtual Machine; 64-bit versus 32-bit computing; performance evaluation; PowerPC
doi:10.1002/spe.679