

1 Introdução

A máquina virtual Java é bastante difundida nos dias de hoje. Ela está presente em servidores, computadores de mesa, navegadores da web, e dispositivos móveis como celulares. Ela é a chave para a portabilidade da linguagem: o compilador Java gera um código intermediário (doravante denominado *bytecode* Java), independente de plataforma ou sistema operacional, que é interpretado pela máquina virtual.

A linguagem Java, criada pela Sun Microsystems, é atualmente controlada por um comitê gestor, denominado JCP (*Java Community Process*). Isso faz com que novas funcionalidades tenham que passar por um crivo para serem incorporadas à linguagem. Na época de criação do Java, o JCP definiu que a máquina virtual seria baseada na manipulação dos operandos através de uma pilha. Um fator determinante da escolha foi a maior simplicidade para a implementação do compilador e da máquina virtual baseada em pilha, comparativamente ao modo tradicional usado nos processadores, que é a manipulação de operandos através de registradores.

Diversos estudos (Davis et al. 2003; Ertl et al., 2005) indicam que máquinas virtuais baseadas em registradores podem ser mais rápidas que as baseadas em pilha. Para exemplificar, vejamos o código abaixo de pseudo-instruções para a expressão “ $a = b + c$ ”, em uma máquina virtual baseada em pilhas (coluna da esquerda) e outra baseada em registradores (coluna da direita):

```
    iload b           iadd a, b, c
    iload c
    iadd
    istore a
```

A máquina virtual baseada em pilha gasta mais instruções, pois requer que cada variável seja colocada na pilha, que é de onde a instrução *iadd* irá retirar os operandos para efetuar a soma. O resultado é então guardado de volta na pilha e então armazenado na variável destino. Já na baseada em registradores não é preciso mover os operandos para nenhum local temporário, e consegue-se fazer o mesmo com menos instruções.

Visto que o desempenho de Java era bem inferior ao das linguagens tradicionais como C, uma solução foi adotada: a criação dos compiladores em tempo real (JIT - *Just In-Time compiler*). Os compiladores JIT transformam os bytecodes Java em código de máquina do processador-alvo no momento da execução; com isso consegue-se um ganho de desempenho na ordem de 10 vezes. Porém, a criação destes compiladores é difícil e deve ser feita para cada processador-destino. Além disso, o JIT aumenta o uso de memória, e também faz com que ocorra um atraso na execução do programa, pois na primeira vez que o método ou classe é carregado, devem ser compilados e convertidos para código de máquina.

Os computadores móveis atuais, conhecidos como assistentes pessoais digitais (*PDA*, da sigla em inglês), têm poder de processamento cerca de 10 a 30 vezes menor que os computadores de mesa¹, e memória dezenas de vezes menor². Nesses equipamentos, o atraso inicial provocado por um JIT é bastante perceptível.

Sendo o nosso foco equipamentos móveis, como PDAs e *Smartphones*³, decidimos criar uma nova especificação de bytecodes, específicos para máquinas virtuais baseadas em registradores. Esperamos com isso obter um aumento no desempenho das aplicações nestes dispositivos, sem requerer o uso de compiladores JIT.

Tendo sempre o cuidado de minimizar os requisitos de memória da nova especificação, iremos apresentar, no capítulo 2, resumos de artigos que sugere modificações em aspectos da linguagem que permitam uma redução no tamanho do código gerado, assim como um aumento na eficiência do mesmo. Em seguida, no capítulo 3, iremos apresentar as principais características da nova especificação, além de descrever os resultados obtidos por um analisador estático de código Java que usamos para validar a escolha das instruções. No capítulo 4, mostraremos resultados de código gerado com a nova arquitetura, e, finalmente, no capítulo 5 apresentaremos as conclusões relativas ao trabalho.

¹ Muitos computadores de mesa atuais têm um *clock* de 3 GHz, enquanto que os PDAs mais comuns têm de 100 a 300 MHz.

² Os computadores de mesa possuem entre 512MB e 2GB mais o armazenamento em disco rígido (80GB), enquanto que os PDAs possuem entre 32MB e 128MB, que devem ser compartilhados entre memória RAM e armazenamento de programas.

³ Celulares com uma maior capacidade de processamento.