Evaluation of the runtime performance of control flow structures for dynamic dispatch in Java
Olivier Zendra, Karel Driesen, Feng Qian, Laurie Hendren

To cite this version:
Olivier Zendra, Karel Driesen, Feng Qian, Laurie Hendren. Evaluation of the runtime performance of control flow structures for dynamic dispatch in Java. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), 2001, Tampa, United States. inria-00563339

HAL Id: inria-00563339
https://inria.hal.science/inria-00563339
Submitted on 4 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Evaluation of the runtime performance of control flow structures for dynamic dispatch in Java

Olivier Zendra & Karol Driesen (INRIA - McGill University, ACL group) - Feng Qian & Laurie Hendren (McGill University, Sable group)

{ozendrakarel,fqian,hendren}@cs.mcgill.ca

Some results:

1. Virtual calls tend to be more expensive than other structures.
2. Virtual calls on constant patterns tend to be less efficient than other virtual calls.
3. Virtual calls on constant patterns behave like if sequences.
4. Virtual calls on constant patterns are faster on U3 than on P3.
5. Virtual calls on constant patterns are faster than if sequences.
6. Virtual calls on U3 behave like if sequences.
7. Virtual calls on HSC behave like if sequences.
8. Virtual calls on HSS behave like if sequences.
9. Virtual calls on IBM are sensitive to execution patterns.
10. Virtual calls on IBM are sensitive to execution patterns.

Type ID Patterns:

- Static Call
- Virtual Call
- If Sequence
- Switch Table
- Binary Tree
- No Call

Control structures:

- We generate different control structures for polymorphic calls.
- Different control structures are generated for different execution environments.
- Different control structures are generated for different execution environments.
- Different control structures are generated for different execution environments.
- Different control structures are generated for different execution environments.
- Different control structures are generated for different execution environments.

Conclusion:

- Virtual calls are significantly improved at compile time.
- Virtual calls are significantly improved at compile time.
- Virtual calls are significantly improved at compile time.
- Virtual calls are significantly improved at compile time.
- Virtual calls are significantly improved at compile time.
- Virtual calls are significantly improved at compile time.

Methodology:

- A benchmark suite is used to test the performance of different control flow structures.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite is run on different execution environments.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.

Assumptions:

- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.
- The benchmark suite includes different types of control structures and different numbers of possible receiver types.

Platforms:

- SUN UltraSparc III
- IBM
- Intel Pentium III

Hardware platforms:

- Intel Pentium III
- SUN UltraSparc III
- Intel Celeron
- AMD Athlon

Virtual calls tend to be more expensive than other structures.

Dynamic dispatch in current VMs can be significantly improved at compile time.

Motivation:

- Frequent dynamic dispatch in Java programs
- Efficient implementation crucial
- Which control structure is the most efficient for dynamic dispatch?

For comparison we also run benchmarks without any call!