

# Context Threading: A flexible and efficient dispatch technique for virtual machine interpreters - CGO'05

Marc Berndt  
Benjamin Vitale  
Mathew Zaleski  
Angela Demke Brown

# Interpreter performance

- Why not JIT?
  - High performance JVM still interpret
  - People use interpreted languages that don't yet have JITs
  - They still want performance
- 30-40% of execution time is due to stalls caused by branch misprediction.
- This technique eliminates 95% of branch misprediction

# Switched Interpreter

```
while(1) {  
  opcode = *vPC++;  
  switch(opcode) {
```

```
    case iload_1:  
      ..  
      break;
```

```
    case iadd:  
      ..  
      break;
```

```
    //and many more..
```

```
  }  
};
```

# Threading Dispatch

```
0:   iconst_0
1:   istore_1
2:   iload_1
3:   iload_1
4:   iadd
5:   istore_1
6:   iload_1
7:   bipush 64
9:   if_icmplt 2
12:  return
```

```
iload_1:
  ..
  goto *vPC++;
```

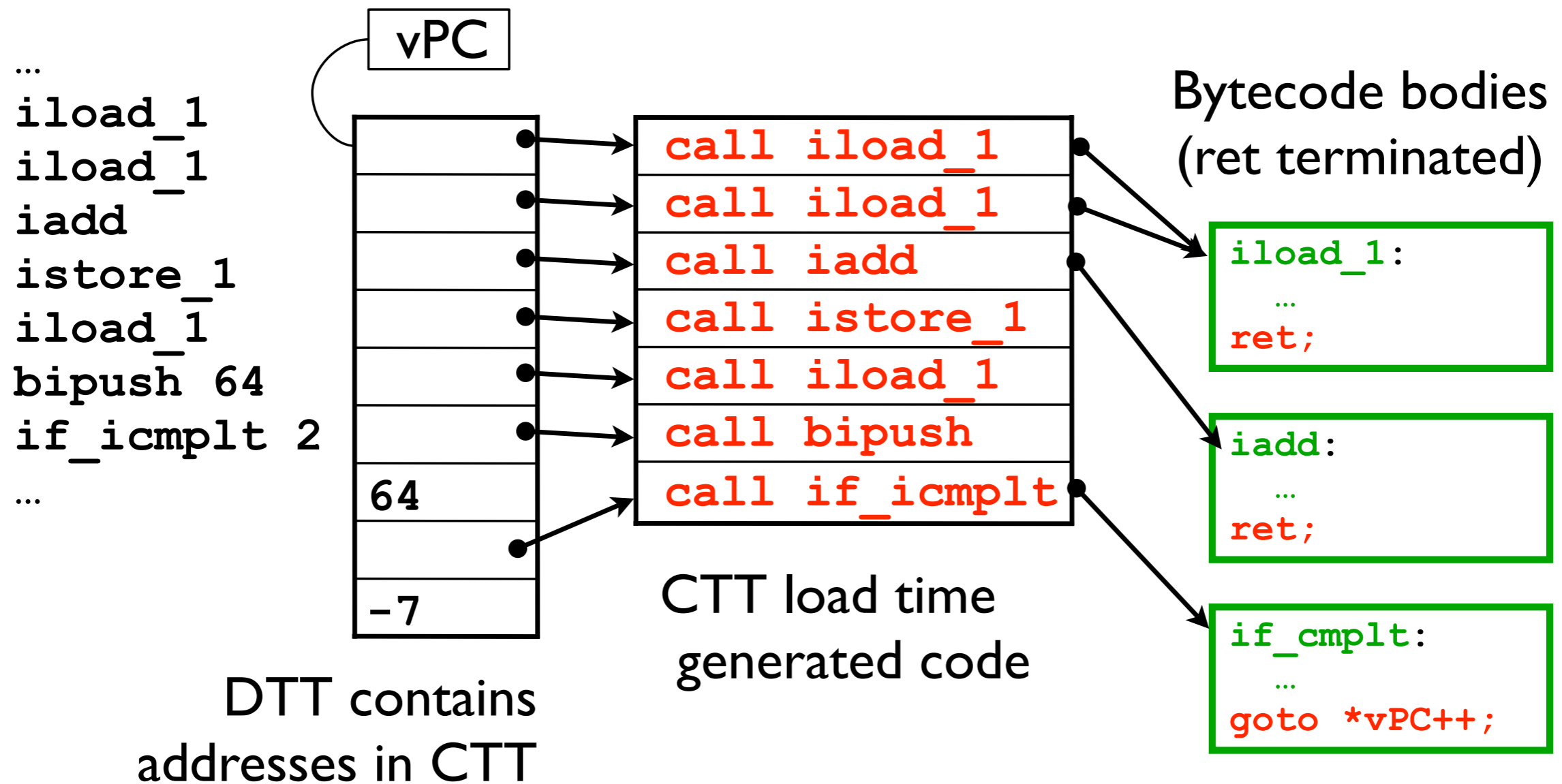
```
iadd:
  ..
  goto *vPC++;
```

```
istore:
  ..
  goto *vPC++;
```

# Key Observation

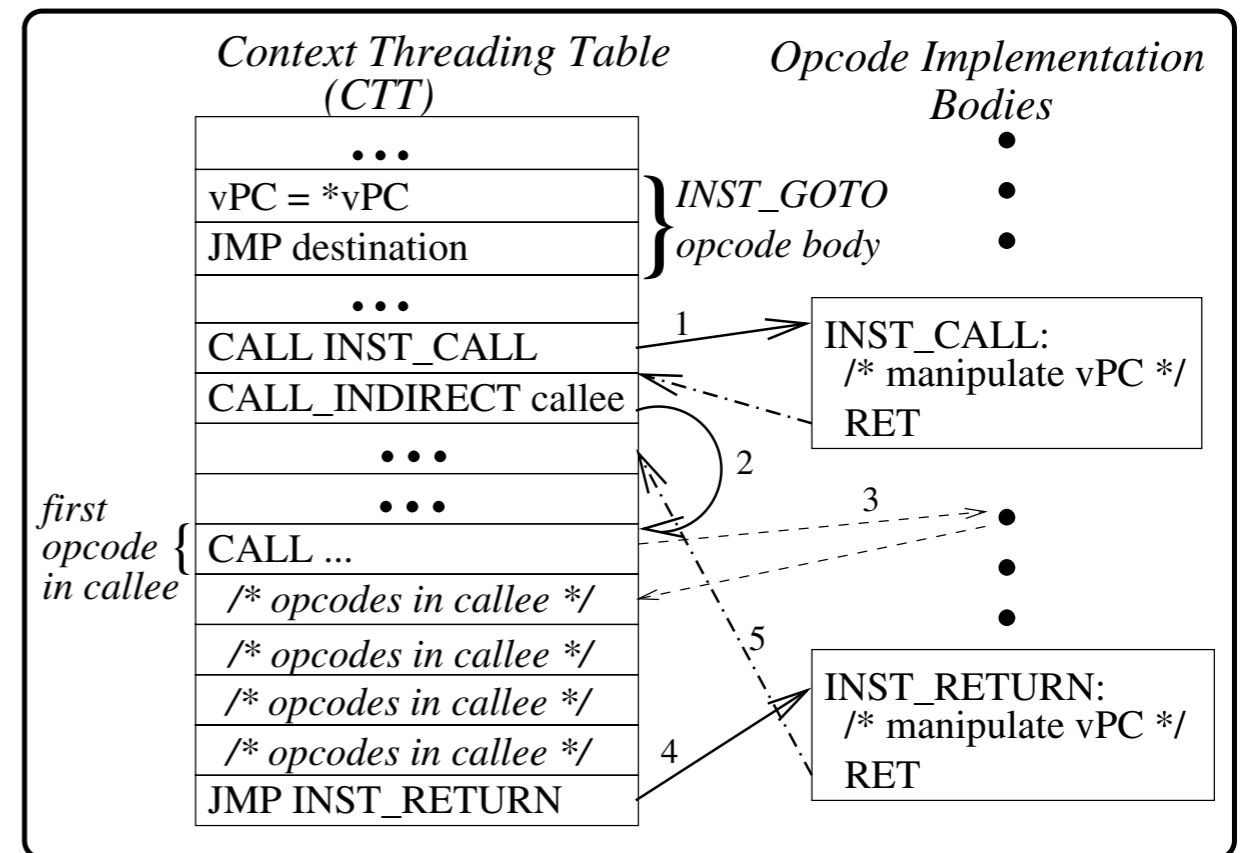
- Virtual and native control flow similar
  - Linear or straight-line code
  - Conditional branches
  - Calls and Returns
  - Indirect branches
- Hardware has predictors for each type
  - Direct uses indirect branch for everything!
- Solution: Leverage hardware predictors

# Subroutine Threading

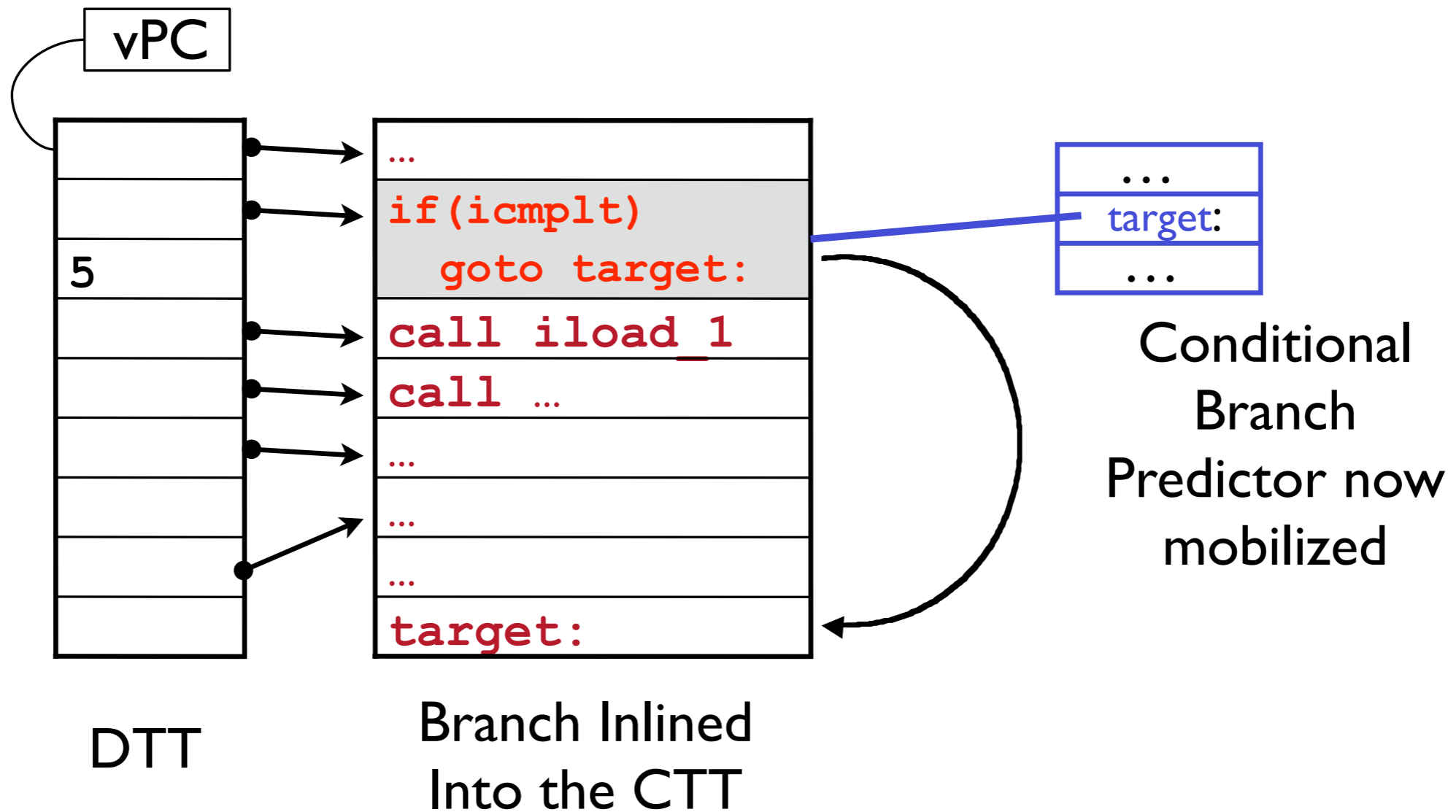


# Virtual Call and Return

- Virtual call
- not so problem
- Virtual return
- It may go back to multiple call site
- Using native return addr



# Specialized Branch Inlining



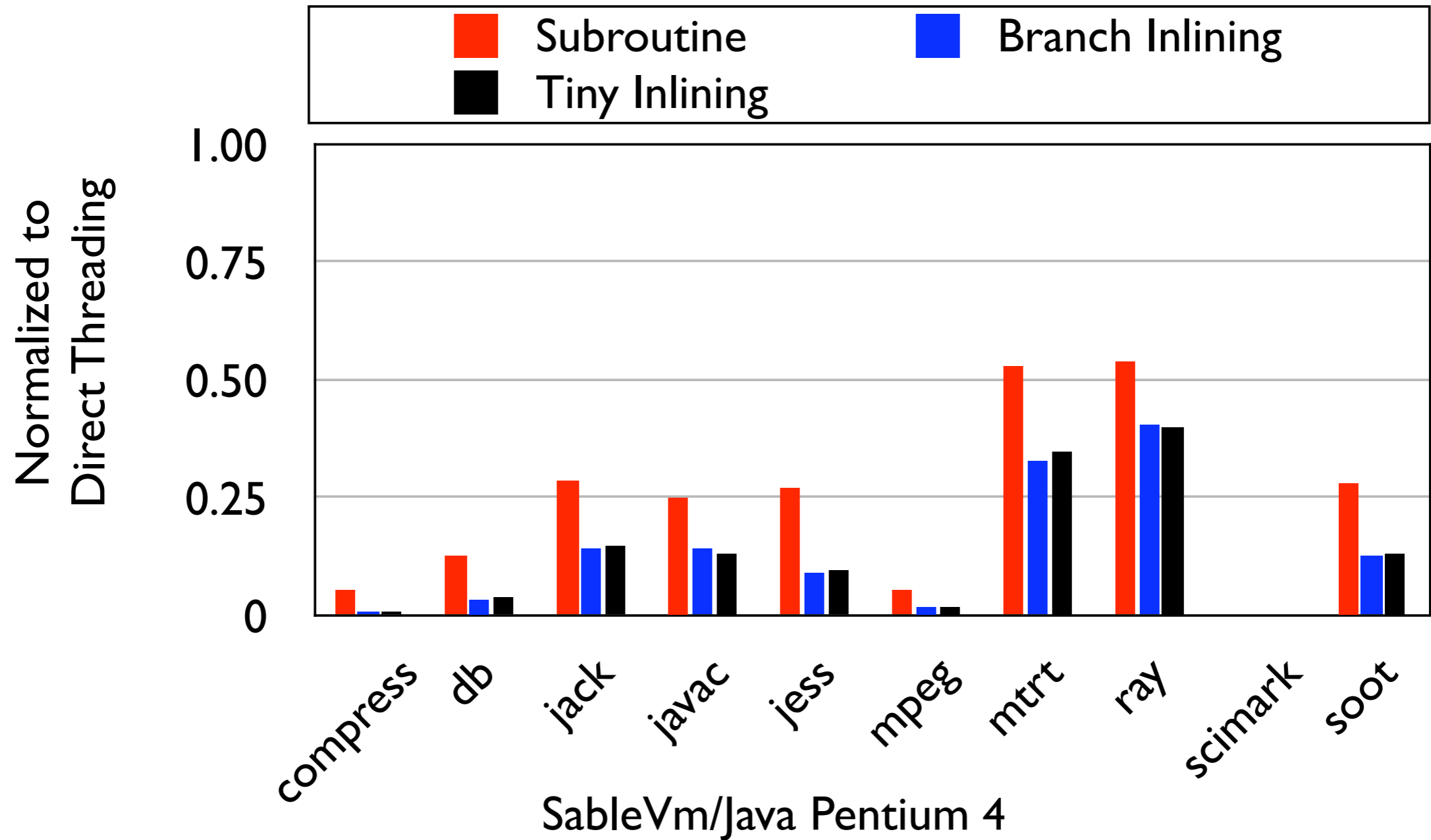


# Experimental Setup

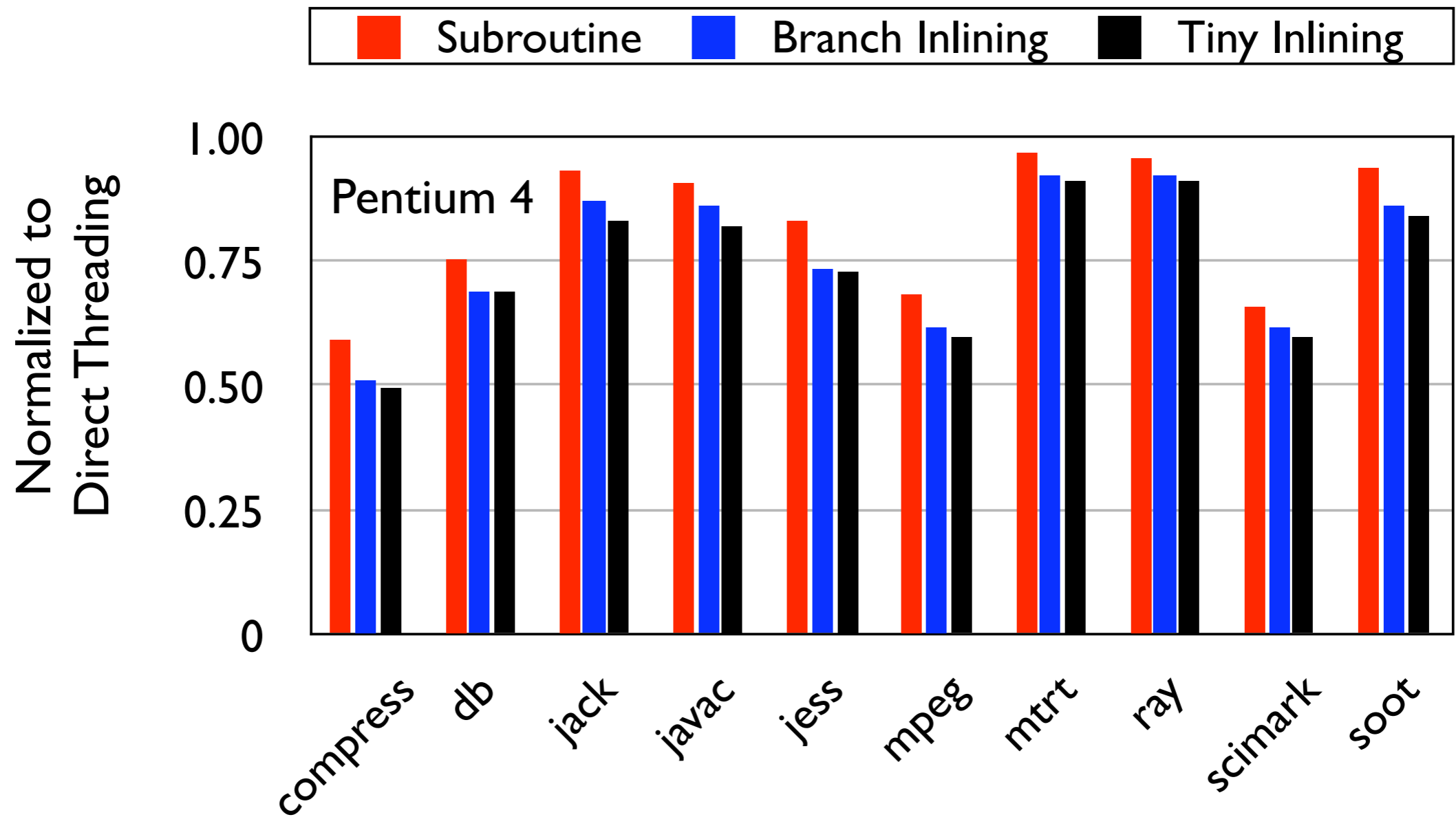
- Two VM on two hardware architecture
  - VM: Java/SableVM, Ocaml interpreter
    - Compare against direct threaded SableVM
    - SableVM use selective inlining
  - Arch: P4, PPC
- Mispredicted Taken Branches(MPT)\*
- Execution Time

\* MPT events are counted with performance counter 8 by setting the P4 CCCR to 0x0003b000 and the ESCR to value 0xc001004 [2]

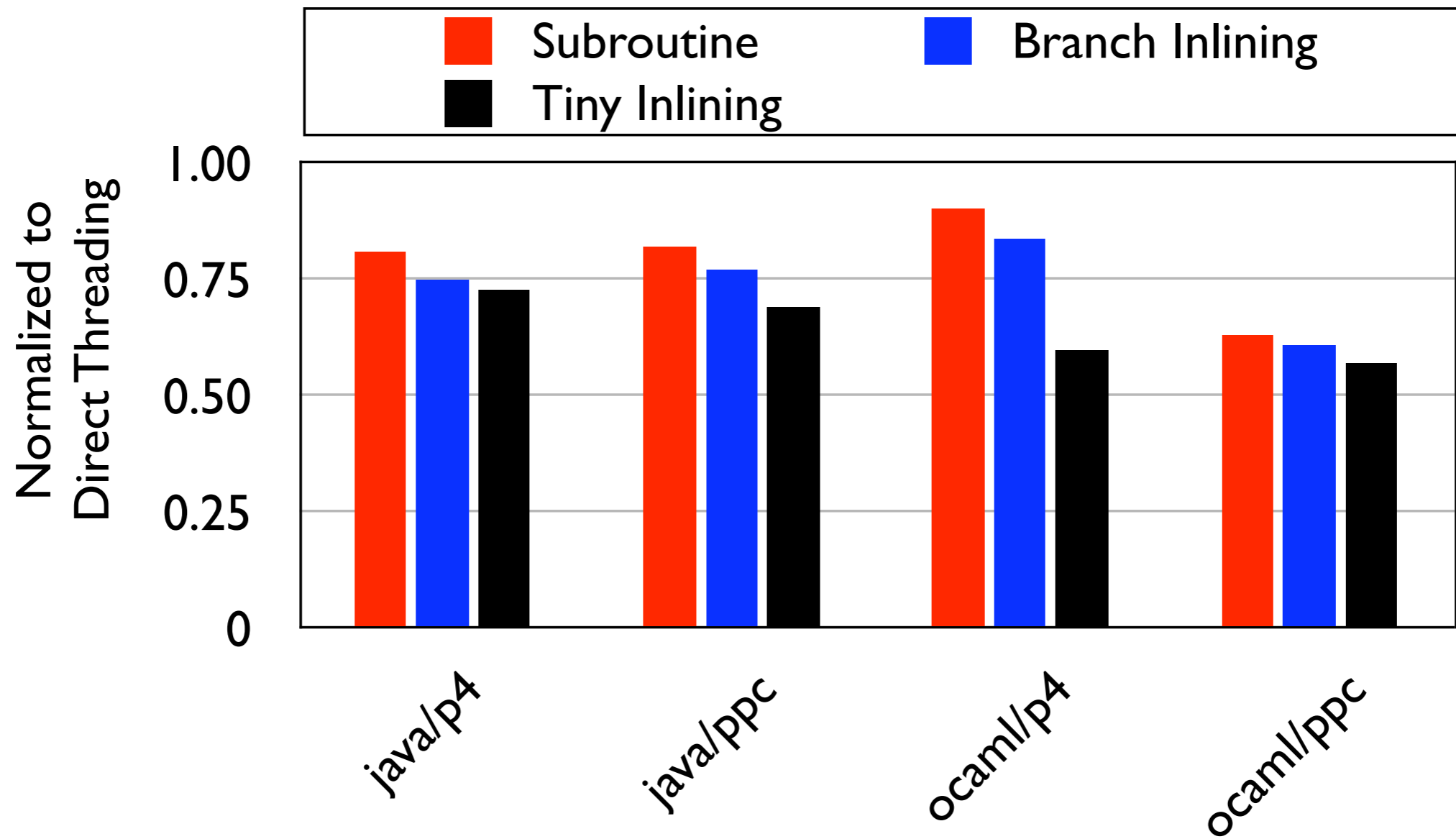
# Mispredicted Taken Branches



# Execution Time



# Execution Time



# Conclusion

- **Context Problem: branch mispredictions due to mismatch between native and virtual control flow**
- **Solution: Generate control flow code into the Context Threading Table**
- **Results**
  - **Eliminate 95% of branch misprediction**
  - **Reduce execution time by 30-40%**