

Table of Contents

List of Tables	IX
List of Figures	X
Acronyms	XII
1 Introduction	1
1.1 Context: High Performance Computing, Artificial Intelligence and Java	1
1.1.1 Java-powered AI and Data Mining	2
1.1.2 Distributed Java in HPC	3
1.1.3 Performance analysis	4
1.2 MareNostrum Tools Environment	5
1.2.1 Extrae	6
1.2.2 Paraver	6
1.3 Problem Statement and Goal	7
1.4 Materials and Methods	7
2 Extrae for JAVA: State of the Art	9
2.1 The example program	9
2.2 Generate the traces	11
2.3 Pthread instrumentation	13

2.4	Traces analysis	13
2.5	Extrae Java API through JNI implementations	15
2.6	Experimental features	15
2.6.1	Java Virtual Machine Tool Interface	15
2.6.2	AspectJ for User Functions	16
2.7	Where to go from here	17
3	Java Tracing Methodologies	18
3.1	Linker Preload approach	18
3.2	Event-driven instrumentation	19
3.3	Bytecode and Native Instrumentation	20
3.3.1	Bytecode manipulation in C and Java	21
3.3.2	Native methods instrumentation	21
3.4	Aspect Oriented Programming approach	22
3.5	Discussion on the methodology to adopt	22
4	Tracing threads on a single JVM	24
4.1	JVM Tool Interface library	24
4.2	Generate the Aspects	24
5	Tracing in a distributed environment	25
5.1	HPC and Distributed Systems	25
5.2	Singularity containers	25
5.3	Shared resources for tracing	25
6	Case Study: Hadoop MapReduce	26
6.1	What methods to instrument	26
6.2	Probes implementation	26
6.3	Aspects generation	26

6.4	Distributed execution	26
7	Discussion of Results	27
7.1	What to Look for	27
7.2	Tracing overhead analysis	27
7.3	Further Improvements	27
8	Conclusions	28
A	Environment set-up	29
B	Extrae State of the Art complete code (with the Example)	30
	Bibliography	31