



Overhead

Version 4.0.5703.0

June 1, 2015

Copyright 2005-2015 MyARM GmbH

Copyright © 2005-2015 MyARM GmbH

MyARM GmbH
Altkönigstraße 7
65830 Kriftel
Germany

Web: <http://myarm.com/>
Mail: info@myarm.com

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Contents

1	Introduction	2
1.1	Performance impact	2
1.2	Overhead measurement	2
2	Test environment	3
3	Agent bindings	3
3.1	ARM 4.0 C Binding	3
3.2	ARM 4.0 Java Binding	4
3.3	ARM 4.0 C# Binding	5

1 Introduction

This document describes overhead produced by the MyARM ARM implementation and the methods how the overhead was measured.

1.1 Performance impact

Measuring any business or technical transactions needs to execute itself. Therefore getting measurement information is not just for free. In order to decide if the instrumentation does not significantly interfere with the real application tasks the performance impact of the instrumentation is good to know. As of its nature it is highly hardware and platform dependent with higher performing processors the performance impact of the measurement decreases expressed in real time.

One key question about instrumentation for measuring performance is the performance overhead of the measurement itself. The interference with the application to measure should be reduced to a minimum. The MyARM ARM implementation is written with this requirement in mind.

1.2 Overhead measurement

ARM can be used in a variety of use cases therefore we measured the overhead of MyARM as described in the following section.

1.2.1 Simulating a real application

Simulation of a real application is done using busy waiting for a defined number of microseconds (application load, t_{app}). Each such task is measured using normal ARM start and stop calls (instrumentation overhead, t_{arm}). The simulation application runs a defined number of seconds. Therefore the total number of ARM transaction measurements N_{arm} and the total time of the application T_{app} (application load) can be easily calculated using the following formulas:

- $T_{app} = 60seconds$
- $t_{app} = 1millisecond$
- $N_{arm} = T_{app}/t_{app} * 1000$

When the ARM instrumentation ideally needs no time the application would execute exactly in 60 seconds. This is for sure not true, therefore we measure the overall execution time (R_{app}) of the running application without initialization and termination of the application. The difference between the measured execution time and the 60 seconds are the overhead for the ARM instrumentation. This can be broken down to a single transaction measurement overhead by the following formula:

- $t_{arm} = (R_{app} - T_{app})/N_{arm}$

1.2.2 ARM optional features

One major factor influencing the performance impact is how many and which context information is associated with a transaction measurement. The different optional features (correlators, metrics and context values) of ARM influences the time spent by the ARM instrumentation in manifold ways. To reflect these options we measured the overhead of MyARM for the following typical ARM instrumentation patterns:

1. Simple transaction measurement using null call library (e.g. `libarm4null.so` for the ARM 4.0 C binding) (`NullCall`).

2. Simple transaction measurement without any additional data (*Simple*).
3. Simple transaction measurement returning a correlator (*CorrGen*).
4. Transaction measurement with 3 gauge metrics attached (*Metrics*).
5. Transaction measurement with 1 context value string of size 100 Bytes (*Ctxt 1*).
6. Transaction measurement with 5 context value string of size 100 Bytes (*Ctxt 5*).
7. Transaction measurement with 10 context value string of size 100 Bytes (*Ctxt 10*).
8. Transaction measurement with 20 context value string of size 100 Bytes (*Ctxt 20*).

2 Test environment

All tests were executed on an Intel® Core™ i7-3517UE CPU (CompuLab Intense PC) running under Linux (Ubuntu-based) in 64bit using kernel 3.13.0-24-generic.

3 Agent bindings

The following figure shows the transaction measurement overhead for the simulated application and the ARM optional features as described in section *ARM optional features* for the ARM 4.0 C-, Java- and C#-Agent.

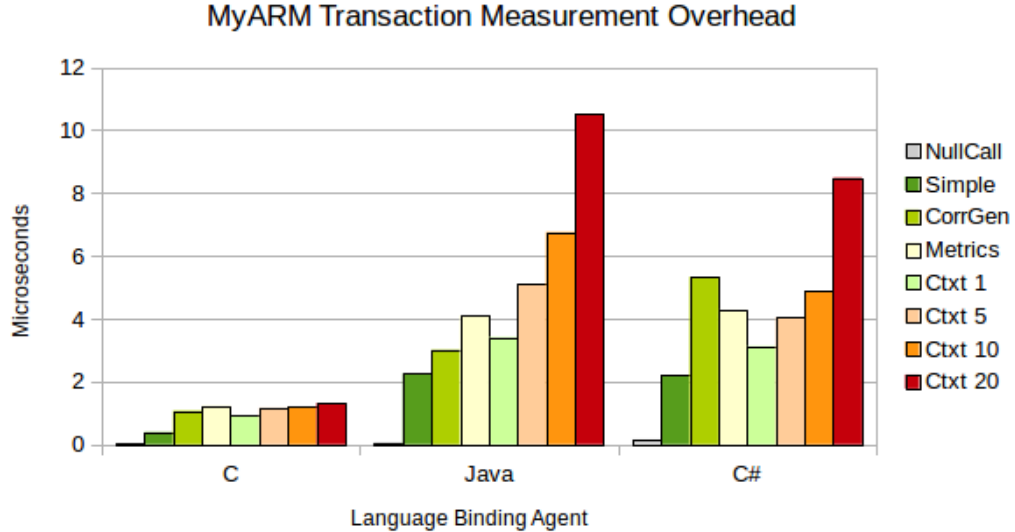


Figure 1: MyARM ARM 4.0 transaction measurement overhead

3.1 ARM 4.0 C Binding

3.1.1 Overhead

This section describes the transaction measurement overhead of the ARM 4.0 C binding using the `arm_start_transaction()` and `arm_stop_transaction()` function calls.

BINDING	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
C	0.016	0.388	1.073	1.229	0.936	1.164	1.212	1.320

Table 1: MyARM average transaction measurement overhead for the ARM 4.0 C binding (in microseconds))

The shown results are the average transaction measurement overhead values from the last 10 MyARM builds of the version MyARM 4.0.x.0. See the following table for all detailed results:

3.1.1.1 Details

VERSION	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
4.0.5685.0	0.009	0.434	1.090	1.194	0.900	1.171	1.205	1.327
4.0.5684.0	0.012	0.325	1.042	1.153	0.893	1.102	1.156	1.321
4.0.5683.0	0.011	0.336	1.005	1.196	0.897	1.095	1.165	1.257
4.0.5682.0	0.018	0.485	1.126	1.314	0.957	1.162	1.217	1.302
4.0.5679.0	0.019	0.420	1.082	1.234	0.961	1.195	1.249	1.336
4.0.5677.0	0.017	0.399	1.081	1.259	0.967	1.187	1.214	1.315
4.0.5676.0	0.022	0.343	1.116	1.262	0.957	1.191	1.233	1.317
4.0.5675.0	0.017	0.372	1.068	1.249	0.937	1.199	1.239	1.375
4.0.5674.0	0.016	0.380	1.037	1.222	0.923	1.186	1.220	1.324
4.0.5672.0	0.021	0.382	1.083	1.207	0.971	1.151	1.219	1.324

Table 2: MyARM transaction measurement overhead for the ARM 4.0 C binding (in microseconds)

3.2 ARM 4.0 Java Binding

3.2.1 Overhead

This section describes the transaction measurement overhead of the ARM 4.0 Java binding using the `ArmTransaction.start()` and `ArmTransaction.stop()` method calls.

BINDING	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
Java	0.059	2.255	3.012	4.103	3.408	5.110	6.756	10.524

Table 3: MyARM average transaction measurement overhead for the ARM 4.0 Java binding (in microseconds))

The shown results are the average transaction measurement overhead values from the last 10 MyARM builds of the version MyARM 4.0.x.0. See the following table for all detailed results:

3.2.1.1 Details

VERSION	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
4.0.5685.0	0.059	2.327	2.961	3.952	3.687	5.043	6.399	10.466
4.0.5684.0	0.059	2.159	2.721	4.030	3.149	4.967	6.507	10.362
4.0.5683.0	0.041	2.122	2.899	3.788	3.197	5.013	6.509	10.400
4.0.5682.0	0.069	2.291	3.248	4.166	3.527	5.261	7.043	10.578
4.0.5679.0	0.059	2.346	3.189	4.111	3.440	5.280	6.836	10.529
4.0.5677.0	0.070	2.264	2.997	4.240	3.382	5.037	7.076	10.368
4.0.5676.0	0.047	2.236	2.921	4.204	3.419	5.165	6.882	10.619
4.0.5675.0	0.052	2.269	3.212	4.199	3.352	5.077	6.746	10.381
4.0.5674.0	0.063	2.274	3.008	4.264	3.428	5.053	6.749	10.764
4.0.5672.0	0.066	2.263	2.967	4.075	3.503	5.200	6.812	10.770

Table 4: MyARM transaction measurement overhead for the ARM 4.0 Java binding (in microseconds)

3.3 ARM 4.0 C# Binding

3.3.1 Overhead

This section describes the transaction measurement overhead of the ARM 4.0 C# binding using the `IArmTransaction.start()` and `IArmTransaction.stop()` method calls.

BINDING	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
C#	0.159	2.208	5.330	4.282	3.124	4.048	4.890	8.488

Table 5: MyARM average transaction measurement overhead for the ARM 4.0 C# binding (in microseconds)

The shown results are the average transaction measurement overhead values from the last 10 MyARM builds of the version MyARM 4.0.x.0. See the following table for all detailed results:

3.3.1.1 Details

VERSION	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
4.0.5685.0	0.128	2.056	5.219	5.510	4.083	4.876	4.664	7.844
4.0.5684.0	0.129	1.791	4.832	3.708	3.994	3.085	4.695	9.915
4.0.5683.0	0.133	1.914	5.084	5.238	2.384	3.437	4.727	8.159
4.0.5682.0	0.169	2.256	5.428	4.401	2.761	5.635	5.195	8.021
4.0.5679.0	0.177	2.403	5.589	4.296	2.732	3.719	5.025	8.505
4.0.5677.0	0.180	2.365	5.727	4.078	2.724	3.979	4.903	10.188
4.0.5676.0	0.176	2.394	5.418	4.213	2.780	3.362	4.842	8.477
4.0.5675.0	0.219	2.993	5.908	4.078	2.786	5.436	5.205	7.983
4.0.5674.0	0.141	1.935	5.130	3.691	2.421	3.353	4.603	7.778
4.0.5672.0	0.140	1.969	4.969	3.602	4.576	3.598	5.036	8.009

Table 6: MyARM transaction measurement overhead for the ARM 4.0 C binding (in microseconds)