Overhead

Version 3.1.5222.0

October 16, 2014

MyARM GmbH
Altkönigstraße 7
65830 Kriftel
Germany

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

Contents

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 for two different point of views:

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} (appliation load) can be easily calculated using the following formulas:

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

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 initialisation and termination of the application. The difference between the measured exeuction 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 Maximal through put

The maximal through put overhead measurement will show how the ARM instrumentation behaves on different hardware platforms and will highlight the performance impact of the different optional features.

The total execution time (R_{app}) of 1 million ARM measurements (start and stop calls) will be measured and the overhead for a single ARM transaction measurement can be calculated by the formula:

- $t_{arm} = R_{app}/N_{arm}$

1.3 ARM optional features

Not only these different kinds of ARM usages need to be highlighted regarding performance overhead also the different optional features (correlators, metrics and context values) of ARM influences the time spent by the ARM instrumentation. 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 Agent bindings

2.1 ARM 4.0 C Binding

2.1.1 Overhead

This section describes the transaction overhead of the ARM 4.0 C binding using the `arm_start_transaction()` and `arm_stop_transaction()` function calls. The following figure shows the transaction overhead for the simulated application and the ARM optional features as described in section *ARM optional features*.

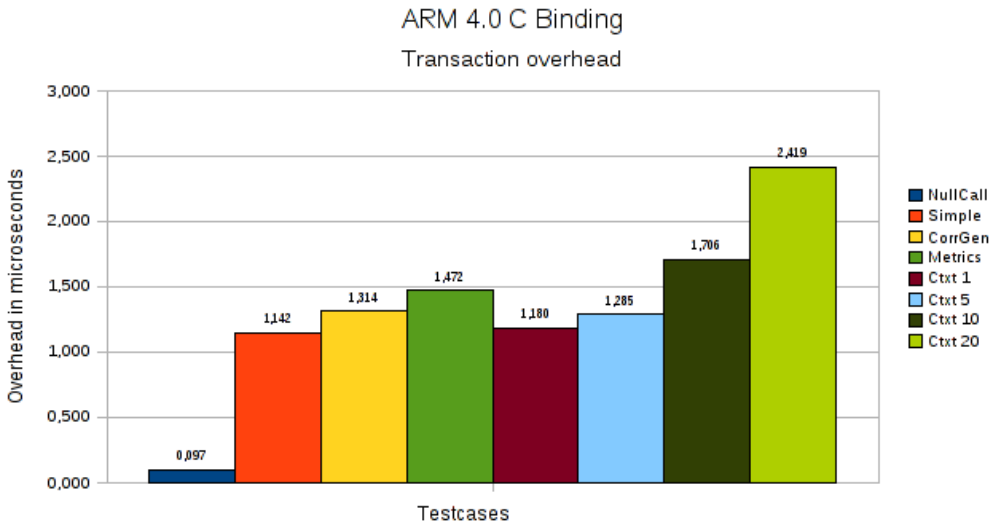

Figure 1: MyARM transaction overhead

BINDING	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
C	0.097	1.142	1.314	1.472	1.180	1.285	1.706	2.419

Table 1: **MyARM transaction overhead (in microseconds))**

The shown results are the average transaction overhead values from the last 10 MyARM builds of the version MyARM 1.3.x.5. See the following table for all detailed results:

2.1.1.1 Details

VERSION	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
1.3.2436.5	0.039	1.120	1.394	1.417	1.035	1.336	1.720	2.096
1.3.2435.5	0.276	1.497	1.837	1.389	1.305	1.777	1.718	2.355
1.3.2434.5	0.004	1.140	1.108	1.902	1.293	1.316	1.702	2.262
1.3.2433.5	0.045	1.035	1.059	1.332	1.020	1.046	1.829	2.454
1.3.2432.5	0.160	1.302	1.405	1.547	1.099	1.093	1.230	2.042
1.3.2431.5	0.019	1.141	1.401	1.320	1.010	1.369	1.706	2.243
1.3.2430.5	0.385	1.009	1.346	1.237	1.324	1.154	2.092	2.159
1.3.2429.5	0.027	1.026	1.226	1.398	1.293	1.211	1.767	2.199
1.3.2428.5	0.009	1.138	1.059	1.604	1.196	1.396	2.127	4.155
1.3.2427.5	0.008	1.012	1.304	1.570	1.221	1.152	1.169	2.220

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

2.1.2 Throughput

This section describes the maximal transaction throughput of the ARM 4.0 C binding using the `arm_start_transaction()` and `arm_stop_transaction()` function calls. The following figure shows the maximal throughput of ARM transactions with the ARM optional features as described in section *ARM optional features*.

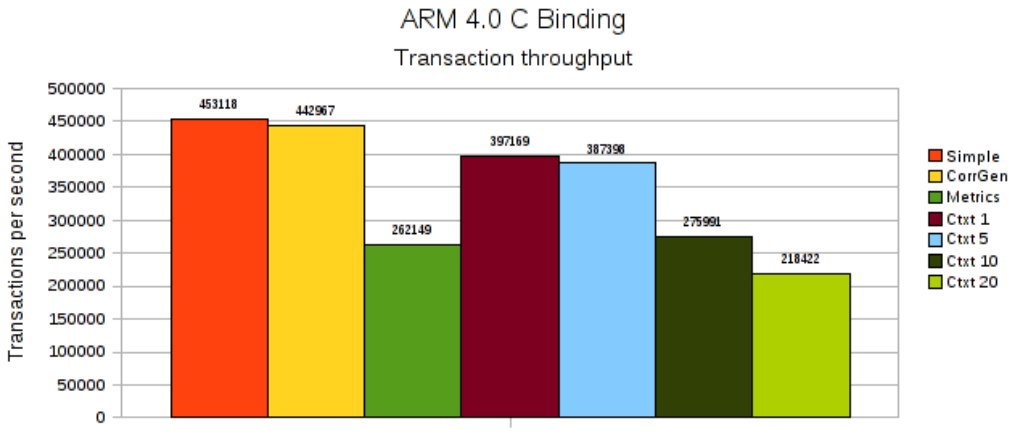

Figure 2: MyARM transaction throughput

BINDING	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
C	135704853	492139	451517	295732	460222	380294	297971	232023

Table 3: **MyARM transaction throughput (in transactions per second))**

2.1.2.1 Details

VERSION	NULLCALL	SIMPLE	CORRGEN	METRICS	CTXT 1	CTXT 5	CTXT 10	CTXT 20
1.3.2436.5	131908719	441419	418928	260771	400574	339927	281358	226295
1.3.2435.5	134336378	434547	642743	255158	390227	345244	278560	226252
1.3.2434.5	136351240	444127	418904	262082	404034	346496	263559	299808
1.3.2433.5	140252454	437004	404619	261227	846277	364531	283166	228092
1.3.2432.5	140765765	445751	558336	607339	405025	345861	276377	230377
1.3.2431.5	125391849	447376	409810	266721	405365	334759	272461	229785
1.3.2430.5	140944326	939412	417820	263800	550607	703113	244324	219787
1.3.2429.5	125407574	446474	409538	262803	404063	347708	281419	223370
1.3.2428.5	140924464	438766	415463	259654	393636	333178	273546	218759
1.3.2427.5	140765765	446512	419011	257765	402416	342125	524941	217703

Table 4: **MyARM transaction throughput for the ARM 4.0 C binding (in transactions per second)**