Using MySQL

Version 4.0.5703.0

June 1, 2015

MyARM GmbH
Altkönigstraße 7
65830 Kriftel
Germany

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

Contents

1 Introduction

This document describes how to configure MyARM for using MySQL database with high transaction measurement numbers. For general information regarding MyARM or MySQL please consult the appropriate user manuals.

2 Motivation

MyARM is designed to measure and store (unlike other performance monitoring tools) all measured transactions of an ARM instrumented application into a connected database. This can result in high number of measurements which need to be stored into the database. This document describes how to configure MyARM using the MySQL database to achieve an optimal transaction `INSERT` throughput.

3 Scenarios

The ARM standard defines a variety number of optional features which influence the throughput of storing transactions into a database we need to define typical classes of scenarios. The following list defines the most common (items 1 to 5) used and worth case (items 6 and 7) ARM transaction measurements:

1. Simple transaction measurement without any additional data (Simple).
2. Simple transaction measurement with a parent correlator (ParentCorr).
3. Transaction measurement with 3 gauge metrics attached (Metrics).
4. Transaction measurement with 1 context value string with a size of 32 Bytes (Context 1).
5. Transaction measurement with 5 context value string with a size of 32 Bytes (Context 5).
6. Transaction measurement with 10 context value string with a size of 32 Bytes (Context 10).
7. Transaction measurement with 20 context value string with a size of 32 Bytes (Context 20).

Some of these scenarios are likely to be combined. For example it is quite usual that a transaction measurement has correlators as well as metrics and context properties. The tested scenarios should help to get an overview how additional data influence the overall database transaction throughput.

4 Test design

The following test design is used to get the transaction storing rate into a MySQL database. A simple C program is used to generate ARM transactions at a maximum rate. MyARM is configured to buffer all transaction measurements and write the buffered transactions into the MySQL database. When all transactions are written into the MySQL database the application terminates.

The elapsed real time divided by the number of transactions stored in the MySQL database gives an good overview of the maximal transaction throughput into the MySQL database.

To increase the maximal transaction throughput MyARM supports a so-called `thread` datasink which uses a defined number of threads to write data to a destination, here MySQL, datasink. Each thread open its own connection to the MySQL database. All test scenarios described above are executed with one, four, eight, ten or twelve MySQL datasink connections.

4.1 Hardware

All tests were executed on an Intel® Xeon® CPU E5-2660 v2 (10 cores with hyper threading) running Debian Linux 7.8 (amd64) with 128GB of memory and a 512GB Samsung SSD 840 PRO hard disk.

4.2 MySQL configuration

The MySQL version 5.5.43 (x86_64) is used and databases were created using the INNODB storage engine with the following parameters changes according to standard debian installation:

innodb_file_per_table – is set

max_binlog_size – is disabled

For a detailed description of the modified MySQL variables please consult the MySQL user manual.

4.3 MyARM configuration

The standard MyARM 4.0.x.0 configuration is used with the following property changes:

db_mysql.connections – is set to `1`, `4`, `8`, `12` or `16` (maximum allowed value).

basic.armdata.buffer.size – is set to `196608`.

basic.armdata.buffer.pool.max – is set to `1024`.

agent.transaction.pool.max – is set to `2048`.

agent.metric.pool.max – is set to `8192`.

These property changes are made to support very high transaction measurement rates of the MyARM ARM 4.0 C agent. For a detailed description of the modified MyARM properties please consult the MyARM user manual.

5 Results

The Figure 1 shows the results of all tests as described above. First of all the number of MySQL database connections influence the overall transaction throughput in any scenario. Thus if high transaction rates are expected configure MyARM to use more MySQL database connections (`db_mysql.connections`).

Within this test setup `12` MySQL database connections performs best. As a rule of thumb the number of MySQL database connections should be around the number of physical CPU cores to get best results. Within our test scenario 10-14 MySQL database connections produced the highest transaction rates per second. Above these numbers the transaction rate per second descreases slightly.

The shown results are the average transaction per second values from the last 10 MyARM builds of the version MyARM 4.0.x.0.

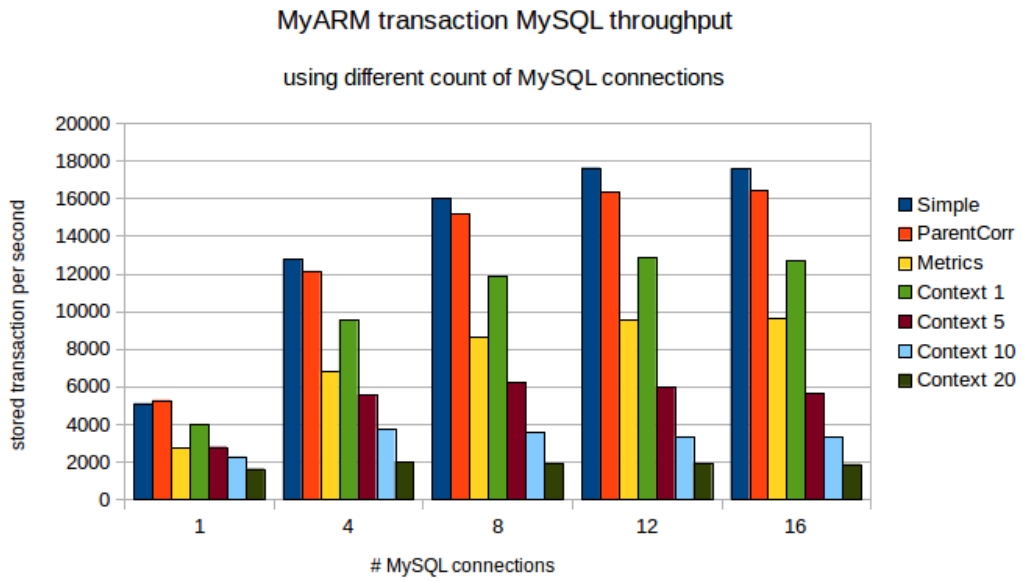

Figure 1: MyARM transaction MySQL throughput

In real world ARM instrumented application a mixture of the outlined scenarios are present. However the following rules should be taken into account for instrumenting applications with ARM:

- Add only information (metrics, context properties) which are of real interest
- Generate only a correlator if its passed to a sub-transaction.
- Prefer context properties instead of metrics if possible.
- Use diagnostic detail for failed or aborted transactions instead of repeating information in any transaction for diagnosing errors.

The following table lists all average transaction throughput rates in detail:

THREADS	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
1	5102	5275	2722	4010	2794	2283	1629
4	12760	12138	6821	9553	5562	3729	1994
8	16010	15176	8641	11857	6208	3571	1916
12	17624	16330	9525	12841	5963	3352	1925
16	17578	16404	9609	12686	5627	3329	1871

Table 1: **MyARM transaction MySQL average throughput (transactions per seconds)**

5.1 One MySQL database connections result details

VERSION	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
4.0.5685.0	4659	5462	2767	3720	2828	2266	1626
4.0.5684.0	5261	5190	2679	3866	2876	2267	1600
4.0.5683.0	5086	5165	2739	4127	2820	2328	1636
4.0.5682.0	5031	5260	2708	4084	2780	2284	1657
4.0.5679.0	5343	5448	2659	4177	2850	2267	1625
4.0.5677.0	5056	5083	2770	4008	2716	2231	1635
4.0.5676.0	5220	5291	2673	3911	2833	2263	1668
4.0.5675.0	4980	5286	2680	4139	2713	2310	1639
4.0.5674.0	5330	5272	2895	3989	2655	2288	1589
4.0.5672.0	5054	5288	2653	4081	2865	2321	1618

Table 2: **MyARM transaction MySQL throughput with one MySQL database connection (transactions per seconds)**

5.2 Four MySQL database connections result details

VERSION	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
4.0.5685.0	12755	12040	6675	9680	5829	3567	1990
4.0.5684.0	12666	11730	6763	9606	5889	3578	1904
4.0.5683.0	12755	12128	6856	9661	5788	4085	2034
4.0.5682.0	12634	12300	6856	9610	5211	3954	2060
4.0.5679.0	12399	12269	6870	9541	5817	3454	2032
4.0.5677.0	12642	12307	6718	9124	4948	3533	2014
4.0.5676.0	12738	12121	6772	9551	5827	3648	1986
4.0.5675.0	12861	11968	6922	9732	5223	3814	1976
4.0.5674.0	13114	12437	6879	9451	5224	3693	1969
4.0.5672.0	13037	12077	6903	9573	5865	3965	1973

Table 3: **MyARM transaction MySQL throughput with four MySQL database connections (transactions per seconds)**

5.3 Eight MySQL database connections result details

VERSION	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
4.0.5685.0	16051	15408	8631	11997	6222	3522	1858
4.0.5684.0	16000	14716	8650	11918	6626	3429	1813
4.0.5683.0	16051	15174	8602	11689	6688	3916	2009
4.0.5682.0	15974	15302	8673	11778	6563	3570	1963
4.0.5679.0	15910	15255	8550	11947	6618	3658	2069
4.0.5677.0	15748	14958	8650	11890	5973	3462	1965
4.0.5676.0	16366	15432	8485	11954	5750	3615	1945
4.0.5675.0	15835	15060	8654	11661	5595	3727	1841
4.0.5674.0	16025	15232	8833	12106	5360	3317	1828
4.0.5672.0	16142	15220	8680	11634	6684	3490	1864

Table 4: **MyARM transaction MySQL throughput with eight MySQL database connections (transactions per seconds)**

5.4 Twelfe MySQL database connections result details

VERSION	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
4.0.5685.0	17746	16501	9505	12936	5524	3245	2048
4.0.5684.0	17761	16488	9350	12845	6321	3231	1983
4.0.5683.0	17559	16326	9620	12936	6731	3482	1938
4.0.5682.0	17889	16194	9610	12820	6711	3594	1974
4.0.5679.0	17543	16207	9429	12787	5810	3321	1878
4.0.5677.0	17256	16420	9564	12953	5370	3390	1939
4.0.5676.0	17873	16406	9601	12944	6049	3349	1881
4.0.5675.0	17376	16142	9478	12706	5488	3431	1880
4.0.5674.0	17574	16460	9451	12812	6025	3290	1878
4.0.5672.0	17667	16155	9638	12666	5605	3191	1853

Table 5: **MyARM transaction MySQL throughput with twelfe MySQL database connections (transactions per seconds)**

5.5 Sixteen MySQL database connections result details

VERSION	SIMPLE	PARENTCORR	METRICS	CONTEXT 1	CONTEXT 5	CONTEXT 10	CONTEXT 20
4.0.5685.0	17746	16366	9689	12787	5586	3179	1925
4.0.5684.0	17182	16339	9671	13089	5701	3334	2030
4.0.5683.0	17889	16597	9661	12795	6361	3424	1809
4.0.5682.0	17497	16406	9510	12730	5465	3287	1933
4.0.5679.0	17636	16064	9578	12995	5585	3350	1904
4.0.5677.0	17346	16420	9478	11363	5768	3245	1817
4.0.5676.0	17809	16542	9573	12903	5534	3343	1820
4.0.5675.0	17452	16339	9633	12453	5360	3460	1803
4.0.5674.0	17652	16583	9666	12928	4961	3366	1742
4.0.5672.0	17574	16380	9629	12812	5947	3301	1929

Table 6: **MyARM transaction MySQL throughput with ten MySQL database connections (transactions per seconds)**