



Cloud Benchmarking

Chief Architect for Cloud Operations and Analytics Huawei Munich Research Centre Prof. Jorge Cardoso jorge.cardoso@huawei.com

16.01.2018

This document is Part C of the lectures on Cloud Benchmarking and contains a set of exercises to benchmark a public cloud platform. Part B was on OpenStack Operations and contained a set of exercises to operate OpenStack cloud platform. Part A was on OpenStack, Operations and Troubleshooting.





Table of Contents

1	Over	rview	3
2	ECS I	Benchmarking	3
	2.1	Preparing the VM	3
	2.1.1	VM Creation	3
	2.1.2	2 SSH Access	4
	2.1.3	B OS Update	4
	2.2	Benchmarking	4
	2.2.1	CPU Benchmark	4
	2.2.2	2 Disk IO Benchmark	7
	2.2.3	8 Network Benchmark	9
3	RDS	Benchmarking	11
	3.1	Preparation	11
	3.1.1	DB Preparation	11
	3.1.2	2 VM Preparation	12
	3.1.3	B Access the database	12
	3.2	DB Benchmarking	12
	3.3	Results	14
4	Web	app Deployment	14
	4.1	Overview	14
	4.2	Preparation	15
	4.3	Configure Wordpress	16





1 Overview

This document describes the benchmarking of AliCloud ECS and RDS. By no means, it should be considered a final study. Many more experiments and tests are needed to obtain a precise view of benchmarking. Nonetheless, it describes the main techniques and tools which are used for benchmarking exercises.

The benchmarking includes the following main activities:

- Create an ECS (Elastic Compute Service) instance
- Install performance analysis tools in VM (e.g., sysbench, netperf, etc.)
- Benchmark the CPU, the network, disk I/O, and file access
- Create RDS (Relational Database Service) instance
- Create database and accounts with privileges
- Run IOPS and connection to test the RDS
- Deploy Wordpress web application in VM
- Access Wordpress remotely

2 ECS Benchmarking

The performance benchmarking and reporting of ECS placed emphasis on:

- Benchmarking of ECS
- Focus on CPU, storage and networking performance, e.g., disk IO, networking IO.
- Benchmarking of RDS
- Focus on database IOPS, QPS, connection number and others relevant index.

Students can open an AliCloud account and use the resources for the exercise.

• http://intl.aliyun.com/campaign/freetrial

2.1 Preparing the VM

2.1.1 VM Creation

An ECS instance with 1 core and 1GB memory was created.

ECS Region: EU Central 1 (Frankfurt) I/O Optimized: IO optimized instance Instance type: 1-core, 1GB Network Type: VPC Bandwidth: 30Mbps (Data transfer usage) OS: Ubuntu 16.04 64bit System Disk: 40GB SSD Cloud Disk Auto-renew: No Server Guard: No Order Type: starter-package Prompt : If you select 1M bandwidth or above, a public IP will be distributed, which is unable to unbind.





Instance	List	China North	1 (Qingdao)	China North 2	? (Beijing)	China Nort	n 3 (Zhang	ijiakou)	China N	orth 5 (Hu	uhehaote)					S	,	Create	Instan	ice
		China East	1 (Hangzhou)	China East 2	(Shanghai)	China So	uth 1 (She	nzhen)	Hong Ko	ng As	ia Pacific N	E 1 (Japar	1)							
		Singapore	Asia Pacific	SE 2 (Sydney)	Asia Pacif	ic SE 3 (Kua	ala Lumpu	r) US	6 East 1 (Vin	ginia) l	JS West 1 (Silicon Va	lley)							
		Middle East	1 (Dubai)	Germany 1 (Fran	kfurt)															
Instance N	lame	\$ Ente	r instance nan	ne (fuzzy search) 5	Search	€¥Tag								Adva	anced Se	arch	M	٥	?
🗌 Insta	nce ID/N	√ame	Monit	or Zone	IP Address	5	Statu	ıs(All)	Network Type(All)	Configu	ration	Bi M	lling ethod(/	All)						Actions
i- gw8c AliCle	157egd5 oudEx	iwxcbv56vni	Q 🗠	Germany 1 Zone A	47.254.120 IP Address 172.25.200 IP Address	8.190 (interr s) 5.40 (Private s)	et O Runn	ing	VPC	CPU: 1 Memory Optimize 30 Mbps	Core(s) /: 1 GB (l/ ed) s (peak val	Si /O 18 17 Iue) So	ubscrip 3-01-26 7:00 (piring pon	tion M	anage	Connec	t C	Change (Re	Configu new	ration More ¬
S	tart	Stop	Restart	Reset Passwo	rd Re	new	Switch to \$	Subscri	ption	Release	Instance	More	•							
											Total: 1 it	tem(s), P	er Pag	e: 20	\$ item	(S) «	¢	1	>	>>

Figure 1. Console and instance panel

2.1.2 SSH Access

To gain access to the VM created, the private key **alicloud.pem** generated using the option Key Pairs of the console was used along with ssh:

\$ Jorges-iMac:.ssh jcardoso\$ ssh -i alicloud.pem root@47.254.128.190

The output is:

```
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com
```

```
* Management: https://landscape.canonical.com
```

```
* Support: https://ubuntu.com/advantage
```

```
Welcome to Alibaba Cloud Elastic Compute Service !
```

2.1.3 OS Update

The VM was updated with the latest Ubuntu packages.

```
$ sudo apt-get update
$ sudo apt-get install unzip
$ sudo apt-get install software-properties-common python-software-
properties
$ sudo add-apt-repository main
$ sudo add-apt-repository universe
$ sudo add-apt-repository restricted
$ sudo add-apt-repository multiverse
```

2.2 Benchmarking

2.2.1 CPU Benchmark

To benchmark CPU, the byte-unixbench tools was used:

```
$ wget https://github.com/kdlucas/byte-unixbench/archive/master.zip
```

```
$ unzip ./master.zip
```

```
$ cd ./byte-unixbench-master/UnixBench
```

```
$ ./Run
```





After running the tool, the output obtained was: _____ BYTE UNIX Benchmarks (Version 5.1.3) System: AliCloudEx: GNU/Linux OS: GNU/Linux -- 4.4.0-62-generic -- #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 Machine: x86 64 (x86 64) Language: en US.utf8 (charmap="UTF-8", collate="UTF-8") CPU 0: Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz (5000.0 bogomips) x86-64, MMX, Physical Address Ext, SYSENTER/SYSEXIT, SYSCALL/SYSRET 20:02:01 up 4 days, 20:17, 1 user, load average: 0.00, 0.00, 0.20; runlevel 2017-12-26 _____ Benchmark Run: Sun Dec 31 2017 20:02:01 - 20:30:05 1 CPU in system; running 1 parallel copy of tests Dhrystone 2 using register variables Double-Precision Whetstone Execl Throughput File Copy 1024 bufsize 2000 maxblocks File Copy 256 bufsize 500 maxblocks File Copy 4096 bufsize 8000 maxblocks Direc Throughput Dhrystone 2 using register variables 34438786.9 lps (10.0 s, 7 samples) 340783.6 lps (10.0 s, 7 samples) 17191.1 lps(30.0 s, 2 samples)10418.7 lpm(60.0 s, 2 samples)1349.6 lpm(60.0 s, 2 samples) Process Creation Shell Scripts (1 concurrent) Shell Scripts (8 concurrent) 1349.6 lpm (00.0 0, 2 1) 4089066.5 lps (10.0 s, 7 samples) System Call Overhead System Benchmarks Index ValuesBASELINERESULTINDEXDhrystone 2 using register variables116700.034438786.92951.1Double-Precision Whetstone55.04186.1761.1Exect Throughput43.05661.11316.5 File Copy 1024 bufsize 2000 maxblocks

 43.0
 30000
 1295671.2
 3271.9

 1655.0
 352375.9
 2129.2

 5800.0
 2915577.3
 5026.9

 12440.0
 2344275.2
 1884.5

 4000.0
 340783.6
 852.0

 File Copy 256 bufsize 500 maxblocks File Copy 4096 bufsize 8000 maxblocks Pipe Throughput 340783.6 Pipe-based Context Switching Process Creation 126.0 17191.1 1364.4 10418.7 42.4 Shell Scripts (1 concurrent) 2457.2 6.0 1349.6 15000.0 4089066.5 1349.6 2249.4 Shell Scripts (8 concurrent) System Call Overhead 2726.0 _____ System Benchmarks Index Score 1970.1

The CPU has a score of **1970.1** for a single task. The AliCloud VM has a similar performance with an Intel Xeon Processor E5-2680 v2 made available in 2013. To compare results with other platforms, see <u>http://linux-</u>

<u>bench.com/results/?table=unixbench_single</u>. The comparison table is not maintained and verified. Thus, entries may not reflect real performance and may be in some cases incorrect. Therefore, the tool has more utility when used to directly compare systems.

Figure 2 shows the CPU performance using the AliCloud console.







Figure 2. CPU benchmark using AliCloud console

Figure 3 shows the **top** unix utility to observe the performance of the CPU. The values shown by AliCloud console and the top utility generally differ in 10%-20%.

top Task	- 20:3	4:58 up 4 total.	4 day 2 r	s, 20:50 unning.	, 2 user 108 sleer	rs, 1 Dina.	loa	ad av	erage: opped.	0.27, 1.8 0 zomb:	32 , 1. 72 ie
%Cpu	(s):10	0.0 us,	0.0	sy, 0.0	ni, 0.0) id,		0.0 w	a, 0.	0 hi, 0.0) si, 0.0 st
KiB	Mem :	1016096	tota	l , 647	104 free,	19	986	504 u	sed,	170388 bi	uff/cache
KiB	Swap:	0	tota	ι,	0 free,			0 u	sed.	653432 av	/ail Mem
PI	D USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2881	3 root	20	0	4372	688	616	R	99.9	0.1	0:06.24	dhry2reg
	1 root	20	0	37884	2504	540	S	0.0	0.2	0:03.69	systemd
	2 root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
	3 root	20	0	0	0	0	S	0.0	0.0	0:09.47	ksoftirqd/0
	5 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
	7 root	20	0	0	0	0	S	0.0	0.0	0:14.43	rcu_sched
	8 root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
	9 root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
1	0 root	rt	0	0	0	0	S	0.0	0.0	0:01.14	watchdog/0
1	1 root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
1	2 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
1	.3 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	perf
1	4 root	20	0	0	0	0	S	0.0	0.0	0:00.08	khungtaskd
1	5 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
1	6 root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
1	7 root	39	19	0	0	0	S	0.0	0.0	0:00.56	khugepaged
1	8 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	crypto
1	9 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
2	0 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioset
2	1 root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kblockd

Figure 3. CPU benchmark using top unix utility

Another tool which can be used for CPU benchmarking is **sysbench**. For example, the following command enables to study how threaded process scales for the VM.

\$ for each in 1 2 4 8 16 32 64; do sysbench --test=cpu --cpu-maxprime=20000 --num-threads=\$each run; done

The results were close to 26 seconds. Naturally, since the VM we are benchmarking has only one core (run: \$ cat /proc/cpuinfo), increasing the number of threads can in some cases even slowdown the computation time. But for multicore CPUs, the true performance will be measured when running the benchmark with the correct amount





of threads. A CPU with 8 cores should run an 8 thread CPU benchmark to get the true performance number.

Ideally the benchmark should also be done for Amazon AWS, MS Azure, and Google GCP to establish a baseline and calculate an offset of performance.

Other programs to benchmark a VM can be found at:

https://wiki.mikejung.biz/Benchmarking

2.2.2 Disk IO Benchmark

To measure disk IO performance, **sysbench** was used. First, a test file of 30GB (we are using a SSD with 40GB) was created. The file needs to be much bigger than the RAM, otherwise the system will use RAM for caching which tampers with the benchmark results.

\$ sysbench --test=fileio --file-total-size=30G prepare

The output was:

sysbench 0.4.12: multi-threaded system evaluation benchmark

128 files, 245760Kb each, 30720Mb total Creating files for the test...

Afterwards, benchmark itself can be run:

```
$ sysbench --test=fileio --file-total-size=30G --file-test-mode=rndrw -
-init-rng=on --max-time=300 --max-requests=0 run
```

The output was:

sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options: Number of threads: 1 Initializing random number generator from timer.

```
Extra file open flags: 0

128 files, 240Mb each

30Gb total file size

Block size 16Kb

Number of random requests for random IO: 0

Read/Write ratio for combined random IO test: 1.50

Periodic FSYNC enabled, calling fsync() each 100 requests.

Calling fsync() at the end of test, Enabled.

Using synchronous I/O mode

Doing random r/w test

Threads started!

Time limit exceeded, exiting...

Done.
```

Operations performed: 428100 Read, 285400 Write, 913193 Other = 1626693 Total Read 6.5323Gb Written 4.3549Gb Total transferred 10.887Gb (37.161Mb/sec) 2378.33 Requests/sec executed

Test execution summary: total time: 300.0002s total number of events: 713500 total time taken by event execution: 157.0585





per-request statistics:	
min:	0.00ms
avg:	0.22ms
max:	33.67ms
approx. 95 percentile:	0.32ms
Threads fairness:	
events (avg/stddev):	713500.0000/0.00
execution time (avg/stddev):	157.0585/0.00

The important number is the *Kb/sec* value:

- Operations performed: ... 1626693. (i.e., 1626693/300s=5422 IOPS)
- Total transferred 10.887Gb (37.161Kb/sec)

Analysing the disk monitoring interface of the AliCloud console, the following results for IOPS were observed (Figure 4).

🚊 d-gw8d57e	gd5wxcbv4i	n1md		C	Create Snapsho	ot Autom	atic Snapsho	ot Policy
Monitoring Information				20	17-12-31 12:25	5 — 2017-1	2-31 12:40	
IOPS							_	
6000					2017-12	-31 12:35:00	-	
4000					Total	IOPS 4075		
0							-	
12:26:00 12:27:00 1	2:28:00 12:29:00	12:30:00	12:31:00	12:32:00) 12:33:00	12:34:00	12:35:00	12:36:00
		Read IOPS	6 – Write IOP	6 🔵 Total I	OPS			

Figure 4. Disk IO benchmark (IOPS)

It can be seen that the benchmarking done using sysbench provides similar values (5422 vs 5477) when compared to the results of the console.

Analysing the disk monitoring interface of the console, the following results for the throughput can be observed (Figure 5).



Figure 5. Disk IO benchmark (throughput)





It can be seen that the benchmarking done provides similar values (36Mb/s vs 39Mb/s) when compared to the results of the console. The difference is most like due to buffering being ignored by sysbench.

After the benchmark, the 30GB test file can be deleted from the system. \$ sysbench --test=fileio --file-total-size=30G cleanup

A proper performance evaluation would require to run the benchmark with different block sizes (e.g., 4, 8, 12, and 16Kb) and with different number of threads (e.g., 1, 2, 3, and 4) assuming the use of a VM with a multicore CPU.

2.2.3 Network Benchmark

2.2.3.1 Point-to-point Connection

The network was tested using the unix utility netperf.

```
$ wget
http://archive.ubuntu.com/ubuntu/pool/multiverse/n/netperf/netperf_2.6.
0-2_i386.deb
$ sudo dpkg -i netperf 2.6.0-2 i386.deb
```

Since netperf works with a client and a server, the server run in the AliCloud VM: \$ netserver -D -4 -L 0.0.0.0 -p 12865

The client run in a remote MacOS computer:

```
Jorges-iMac:osk jcardoso$ sudo netperf -H 47.254.128.190 -1 60 -t
TCP STREAM -p 12865
```

The results show that the network has a throughput of 1.43 *10^6 bits /sec when communicating between ECS and the remote MacOS.

MIGRATED TCP STREAM TEST from (null) (0.0.0.0) port 0 AF_INET to (null) () port 0 AF_INET Recv Send Send Socket Socket Message Elapsed Size Size Size Time Throughput bytes bytes bytes secs. 10^6bits/sec 87380 131072 131072 61.05 **1.43**

Analysing the network monitoring interface of the console, the following results for the throughput can be observed (Figure 6).







Figure 6. Network benchmark (client-server throughput)

Both values are roughly similar (a thorough investigation would be needed to understand why the difference exists). The throughput is mainly limited by the speed of the internet connection on the client side, i.e., the MacOS system. The same benchmark was reexecuted using a remote client located in another cloud provider (https://www.cloud.mwn.de). The results were higher with an AliCloud VM network throughput of 104.33 *10^6 bits /sec.

```
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF INET to 47.254.128.190 ()
port 0 AF INET : demo
Recv Send
              Send
Socket Socket
             Message Elapsed
                       Time
                               Throughput
Size Size
              Size
            bytes
bytes bytes
                      secs.
                               10^6bits/sec
 87380 16384 16384
                       60.23
                                104.33
```

2.2.3.2 Internet Connection

I also tested the upload and download network speed to the internet connection of the AliCloud VM. I used speedtest.

```
$ wget https://raw.githubusercontent.com/sivel/speedtest-
cli/master/speedtest.py
$ python speedtest.py
```

The results of speedtest.py were:

Upload: 35.15 Mbit/s

The results show that the VM network has a throughput of:

- Download 414.37 Mbits/s
- Upload 35.15 Mbit/s





The tests would need to be complemented by varying the number of concurrent connections.

3 RDS Benchmarking

To test the performance of the RDS service, the focus was on database IOPS, QPS, connection number and others relevant index.

3.1 Preparation

3.1.1 DB Preparation

The first step is to start launching a new DBMS.

	Home	Products	•	Message 8	Billing Management	Support ICP	Documentation jorge	e.ca******@gmail.c	com English
•	RDS	RDS Management	China North 1 (Qingdao)	China East 1 (Hangzhou)	China North 2 (Beijing)	China East 2 (Shanghai)		□ Refresh	Create Instance
	Instances		China South 1 (Shenzhen)	China North 3 (Zhangjial	kou) Asia Pacific NE 1 (J	apan) Germany 1 (Frankfurt)	I		
•	Trash		Asia Pacific SE 2 (Sydney) US East 1 (Virginia) Asia	Middle East 1 (Dubai)	Ir) China North 5 (Huheh	US West 1 (Silicon Valley)			
		Basic Information	Tags						
		Instance Name	Search by Instance	ID S	Search 🗆 Tag				
		Instance Name	Sta	atus(All) Created Time	Instance Data Type(All) - Engi	base ne(All) - Availability Zone	Network Type(Network Type) 👻	Billing Method Tags	Action
		m-4xol16x	c2ni7s1rig Cn c2ni7s	eating 2017-12-27 02:5	6 Regular MyS	Germany 1 QL 5.6 (Frankfurt) ZoneA	VPC (VPC:vpc- gw8vdcdj52tlbl7sh9dm9)	Pay-As- You-Go	Manage More 🗸
		C Edit Tags	Instance Authorization				Total: 1 item(s), Per Page	c 30 item(s) «	< 1 > 30

Figure 7. RDS creation

	Home P	roduct	s 🗆	□ Message	e 🤨 🛛 Billing N	Management	Support I	CP Do	ocumentation	jorge.ca*****	***@gmail.com	English		
•			rm-4xol16xc2 (Creatin	g) Back to Instance	25		Operation	Guide	Restart Instance	Back Up Inst	ance D Refresh	n I ≣		
	Basic Information													
•	Accounts		Basic Information			Set Whitelist	Migrate Zone		Instance D	istribution				
•	Databases		Instance ID: rm-4xol16xc2ni7s1rig			Name: m-4	xol16xc2ni7s1rig		Bead	only 🗆	Temporary	,		
•	Connection Options		Instance Region and Zone: Germany 1 (Fran	nkfurt)ZoneA		Instance Type (rds.status.ca	e: Standard ategory.HighAvailabili	ity)	0		0			
	Monitoring and Alarm		Intranet Address: Set Whitelist and then add	fress will be shown.		Intranet Port:	3306		Add Read-o	only Instance	Add Temporary In:	stance		
	Security		Apply for Internet Address Apply for Internet	t Address										
	Instance Availabilit													
	Log Management		Status								Release Instance			
	Backup and Recovery		Status: Creating		Billing Method	Pay-As-You-Go			Created Time: 2	017-12-26 19:56	:03			
	Parameters		Configuration Information											
			Class Family: General		Database Engi	ne: MySQL 5.6			CPU: 1Core					
			Database Memory: 1024MB		Maximum IOP	S: 600			Maximum Numb	er of Connection	s: 300			
			Maintenance Period: 19:00-23:00 Modify		Instance Class	rds.mysql.t1.sm	all							
			Resource Information											
			Instance Space: Total Space5.00G	e: Total Space5.00G				Backup Size: 0.00K (Contains data and log backups. The total amount below 2560M is free) 5 Details						

Figure 8. RDS view

Afterwards, create a new database called mydb.





	Home P	roducts 🗆	۵	Message 🔒	Billing Management	Support ICP	Documentation	jorge.ca******@	gmail.com English
•		😵 rm-4xol	16xc2 (Running)	Back to Instances		Operation Guide	Restart Instance	Back Up Instance	Refresh
	Basic Information	Databases						□Refres	h Create Database
	Accounts								
•	Databases	Database Name	Database Status	Character Set	Bound Accounts	Description			Action
۰	Connection Options	mydb	Running	utf8	admin1234	None		Delete Migrat	te backup files from OSS
	Monitoring and Alarm								
	Security								
	Instance Availabilit								
	Log Management								
	Backup and Recovery								
	Parameters								

Figure 9. The mydb database

3.1.2 VM Preparation

Install the required packages to get access to the RDS:

\$ sudo apt install mysql-client-core-5.7
\$ sudo apt install mariadb-client-core-10.0

3.1.3 Access the database

Use the Intranet ip of database to access the database:

rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com

\$ mysql -u admin1234 -p -h rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com mydb

3.2 DB Benchmarking

To measure MySQL performance, sysbench was used.

First, a test table in the database *test* with 1,000,000 rows of data was created:

```
$ root@AliCloudEx:~# sysbench --test=oltp --oltp-table-size=1000000 --
mysql-db=test --mysql-user=admin1 --mysql-host=rm-
4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com --mysql-
password=H****2#$ --db-driver=mysql prepare
```

Afterwards, launch a read test with 8 threads using the database test:

```
$ sysbench --test=oltp --oltp-table-size=1000000 --mysql-host=rm-
4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com --mysql-db=test --
mysql-user=admin1 --mysql-password=H****2#$ --db-driver=mysql --max-
time=60 --oltp-read-only=on --max-requests=0 --num-threads=8 run
```

The output was:

sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options: Number of threads: 8

Doing OLTP test. Running mixed OLTP test





```
Doing read-only test
Using Special distribution (12 iterations, 1 pct of values are returned in 75 pct
cases)
Using "BEGIN" for starting transactions
Using auto inc on the id column
Threads started!
Time limit exceeded, exiting...
(last message repeated 7 times)
Done.
OLTP test statistics:
    queries performed:
                                         322364
       read:
       write:
                                         0
       other:
                                         46052
        total:
                                         368416
    transactions:
                                         23026 (383.72 per sec.)
                                                (0.00 per sec.)
    deadlocks:
                                         0
    read/write requests:
                                         322364 (5372.11 per sec.)
    other operations:
                                         46052 (767.44 per sec.)
Test execution summary:
    total time:
                                         60.0070s
    total number of events:
                                         23026
    total time taken by event execution: 479.9380
    per-request statistics:
        min:
                                               6.00ms
         avg:
                                              20.84ms
                                              88.00ms
        max:
        approx. 95 percentile:
                                              43.15ms
Threads fairness:
   events (avg/stddev):
                                  2878.2500/74.80
    execution time (avg/stddev): 59.9922/0.00
```

The important number is the transactions per second value:

• transactions: 23026 (383.72 per sec.)

Afterwards, launch a read/write test with 8 threads using the database test.

\$ sysbench --test=oltp --oltp-table-size=1000000 --mysql-host=rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com --mysql-db=test -mysql-user=admin1 --mysql-password=H****2#\$ --db-driver=mysql --maxtime=60 --oltp-read-only=off --max-requests=0 --num-threads=8 run

The output was:

```
sysbench 0.4.12: multi-threaded system evaluation benchmark
```

```
Running the test with following options:
Number of threads: 8
Doing OLTP test.
Running mixed OLTP test
Using Special distribution (12 iterations, 1 pct of values are returned in 75 pct
cases)
Using "BEGIN" for starting transactions
Using auto_inc on the id column
Threads started!
Time limit exceeded, exiting...
(last message repeated 7 times)
Done.
```





OLTP test statistics:		
queries performed:		
read:	174454	
write:	62305	
other:	24922	
total:	261681	
transactions:	12461	(207.64 per sec.)
deadlocks:	0	(0.00 per sec.)
read/write requests:	236759	(3945.11 per sec.)
other operations:	24922	(415.27 per sec.)
Test execution summary:		
total time:	60.0134	ls
total number of events:	12461	
total time taken by event execution:	480.024	11
per-request statistics:		
min:	8	8.67ms
avg:	38	3.52ms
max:	108	3.30ms
approx. 95 percentile:	91	.47ms
Threads fairness:		
events (avg/stddev): 1557.	6250/24.	. 65
execution time (avg/stddev): 60.003	30/0.00	

The important number is the transactions per second value:

• transactions: **12461** (**207.64** per sec.)

To clean up the system afterwards (i.e., remove the test table), run:

```
sysbench --test=oltp --mysql-db=test --mysql-user=admin1 --mysql-
host=rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com --mysql-
password=H****2#$ cleanup
```

3.3 Results

- Using 1 threads (results not shown)
 - The number of transactions is almost 140 for read only operations
 - The number of transactions is close to 100 for r/w operations
- Using 4 threads (results not shown)
 - The number of transactions is almost 400 for read only operations
 - The number of transactions is close to 200 for r/w operations
- Using 8 threads
 - The number of transactions is almost 400 for read only operations
 - The number of transactions is close to 200 for r/w operations

The database reaches maximum number of transactions supported using 4 threads. The IOPs were not measured, only the query performance. For short tests (e.g., a few seconds to up to a few minutes), the monitoring charts of the console did not show results.

4 Webapp Deployment

4.1 Overview

Application deployment and reporting involved deploying one web application to AliCloud. The reporting includes the deployment procedure and the problems





experienced. The deployment is minimal and involved using of one ECS and one RDS instances.

The application deployment was tested using WordPress web application on the cloud. The simplest deployment was carried out involving one compute instance ECS and one database service RDS. The following actions were executed:

- Apache, PHP, modules, and WordPress were installed in the ECS VM
- Create WordPress database in RDS
- Configure WordPress to use the RDS database created
- The database used was the RDS benchmarked in the previous tasks

4.2 Preparation

To install Apache web server:

```
$ sudo apt-get install apache2 apache2-utils
```

Enable Apache2 web server to start at system boot time:

```
$ sudo systemctl enable apache2
$ sudo systemctl start apache2
```

To test that the server is running. Open the web browser and enter:

• http://47.254.128.19.

In case the web server is up and running, the Apache2 default index page is displayed.

Install PHP and modules for it to work with the web and database servers using the command below:

```
$ sudo apt-get install php7.0 php7.0-mysql libapache2-mod-php7.0
php7.0-cli php7.0-cgi php7.0-gd
```

Test if php is working with the web server, create a **info.php** file inside /var/www/html with the content:

<?php phpinfo(); ?>

\$ sudo vi /var/www/html/info.php

To test that the server is running with php support, open the web browser and enter: • http://47.254.128.19/info.php

Install WordPress CMS by download the latest package and extract:

\$ wget -c http://wordpress.org/latest.tar.gz
\$ tar -xzvf latest.tar.gz

Move the WordPress files from the extracted folder to the Apache default root directory:

\$ sudo rsync -av wordpress/* /var/www/html/





Set the permissions on the website directory to give ownership of the WordPress files to the web server as follows:

\$ sudo chown -R www-data:www-data /var/www/html/ \$ sudo chmod -R 755 /var/www/html/

Create the WordPress database within RDS:

\$ mysql -u admin1 -p -h rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com

At the mysql shell, type the following commands:

mysql> CREATE DATABASE wp_myblog; mysql> GRANT ALL PRIVILEGES ON wp_myblog.* TO admin1@'47.254.128.190' IDENTIFIED BY 'H***2#\$'; mysql> FLUSH PRIVILEGES; mysql> EXIT;

Rename the file **wp-config-sample.php** to **wp-config.php**:

\$ sudo mv /var/www/html/wp-config-sample.php /var/www/html/wpconfig.php

Update the file with database information:

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wp_myblog'); /** MySQL database username */
define('DB_USER', 'admin1'); /** MySQL database password /
define('DB_PASSWORD', 'H***2#$'); /** MySQL hostname */
define('DB_HOST', 'rm-4xol16xc2ni7s1rig.mysql.germany.rds.aliyuncs.com
'); /** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8'); /** The Database Collate type. Don't
change this if in doubt. */
define('DB_COLLATE', '');
```

Restart the web server and mysql service:

```
$ sudo systemctl restart apache2.service
$ sudo systemctl restart mysql.service
```

4.3 Configure Wordpress

Open web browser, enter the server address to get the welcome page to configure the system:

• http://47.254.128.190/index.php



(um



limitations of	PC-algorith	https://www	ECS Manage	"log templat	Enterprise Cl	Cloud Per
		1				
	_					
vveicom	е					
Welcome to the you'll be on you	ne famous five-n our way to using	ninute WordPress ins the most extendable	stallation process! and powerful per	Just fill in the info sonal publishing p	rmation below and platform in the world.	
Informat	ion neede	d				
Please provid	e the following ir	nformation. Don't wo	rry, you can alway	s change these se	ttings later.	
Site Title						
Username						
	Usern the @	ames can have only alpha symbol.	anumeric characters, s	paces, underscores, h	yphens, periods, and	
Password	2yQ	Y4etWUV1AErSy1f	🔊 Hide	е		
		Strong				
	Impo	rtant: You will need this	s password to log in.	Please store it in a se	ecure location.	
Your Email						
	Double	e-check your email addre	ss before continuing.			
Search Engin Visibility	ne ⊡Di Itisup	scourage search engo to search engines to ho	gines from indexin nor this request.	g this site		

Figure 10. WordPress Welcome page