Search & Sort Algorithm Results

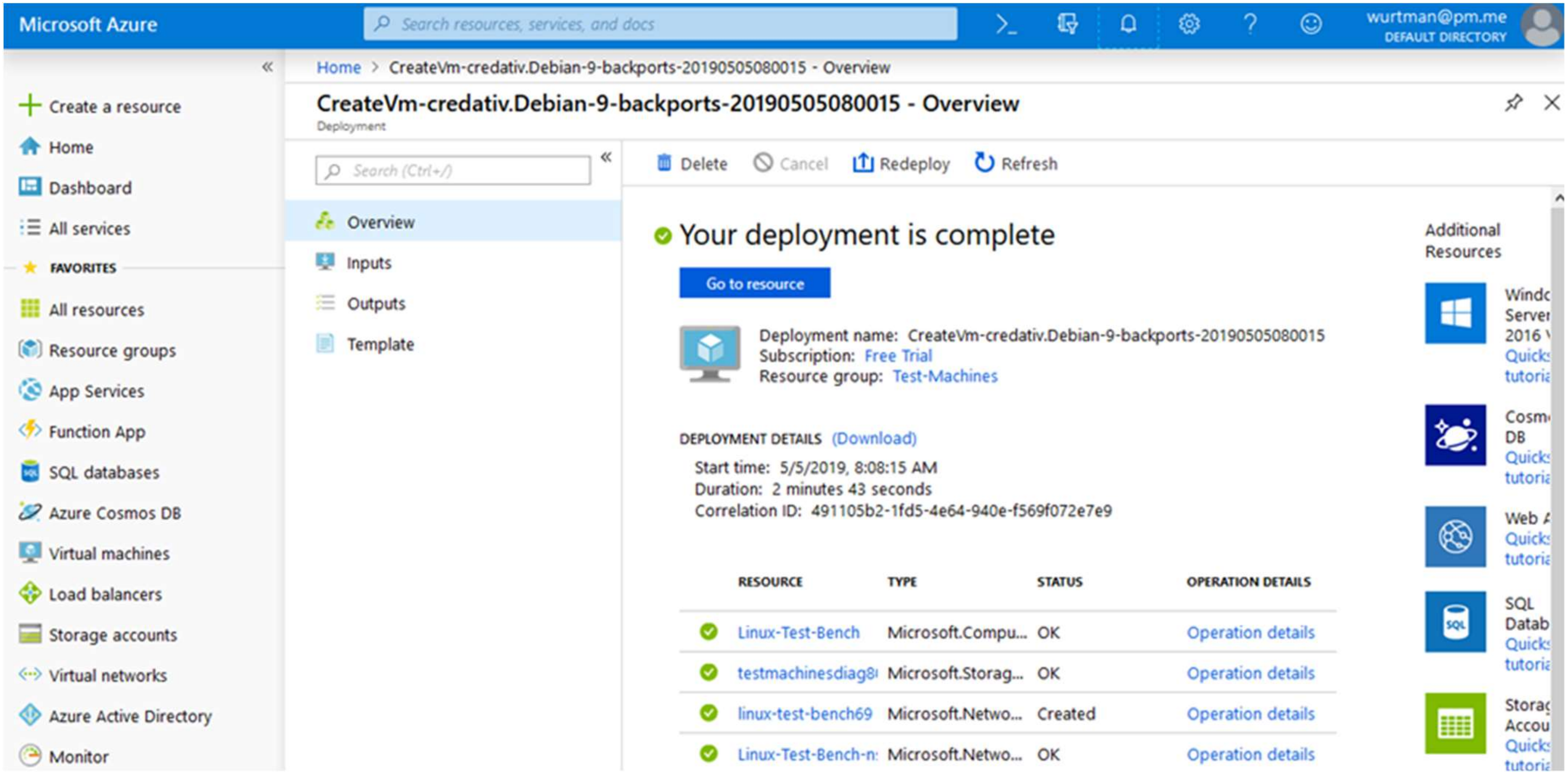
BY: STUART WURTMAN & KYLE SANDRIDGE

Methodology

- We wrote a program to run all the searching and sorting tests.
- The program outputs the results into two csv files: searchResults.csv and sortResults.csv.
- The program can be told to iterate the tests any number of times.
- To run the tests, we started by deploying a compute optimized Linux VM in Azure
- Fetched project files from github, installed g++, compiled program.
- Run program.

Methodology

- Test bench VM deployed in Azure.



CreateVm-credativ.Debian-9-backports-20190505080015 - Overview

Deployment

Your deployment is complete

Deployment name: CreateVm-credativ.Debian-9-backports-20190505080015
Subscription: Free Trial
Resource group: Test-Machines

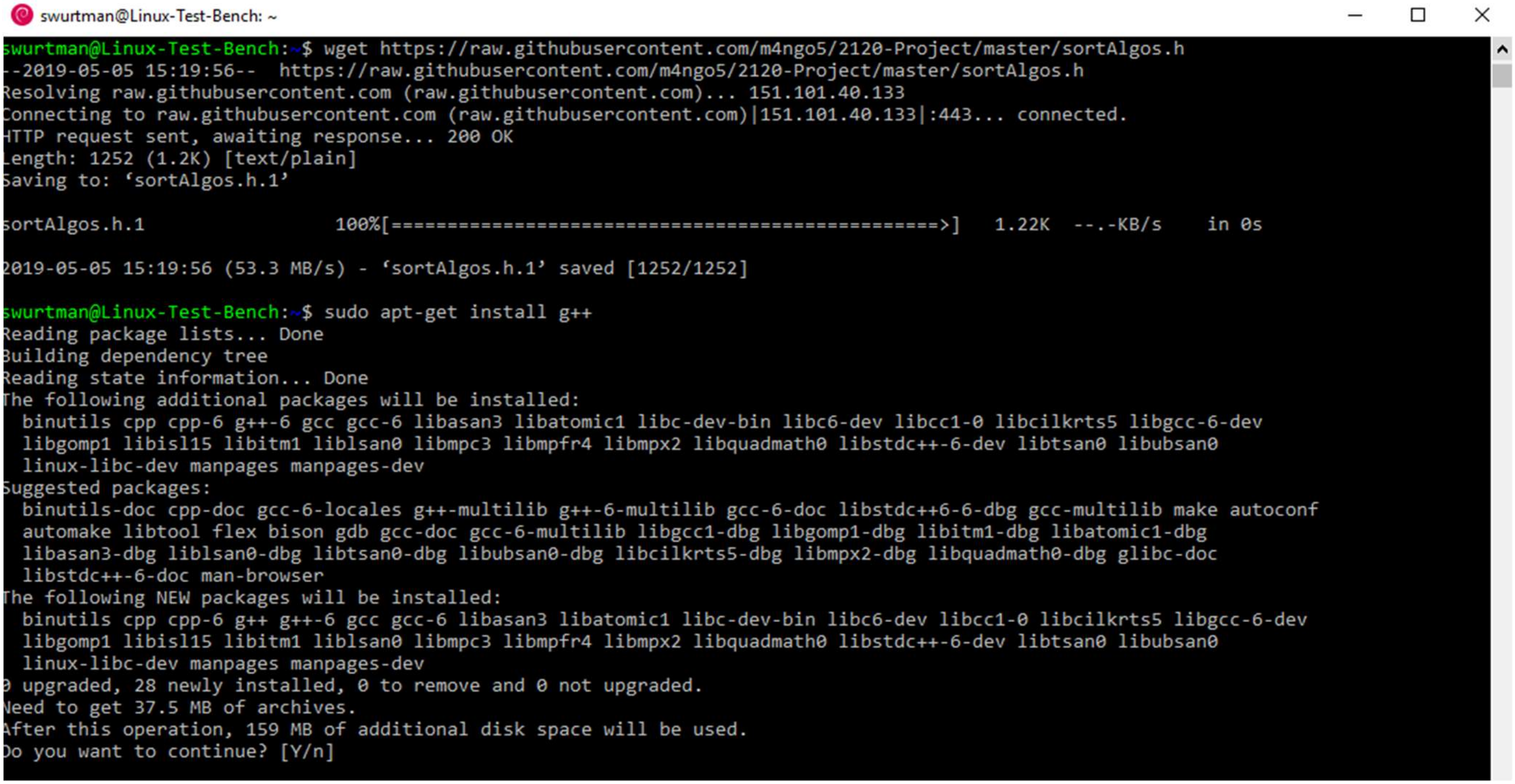
DEPLOYMENT DETAILS (Download)

Start time: 5/5/2019, 8:08:15 AM
Duration: 2 minutes 43 seconds
Correlation ID: 491105b2-1fd5-4e64-940e-f569f072e7e9

	RESOURCE	TYPE	STATUS	OPERATION DETAILS
✓	Linux-Test-Bench	Microsoft.Compu...	OK	Operation details
✓	testmachinesdiag8	Microsoft.Storag...	OK	Operation details
✓	linux-test-bench69	Microsoft.Netwo...	Created	Operation details
✓	Linux-Test-Bench-n	Microsoft.Netwo...	OK	Operation details

Methodology

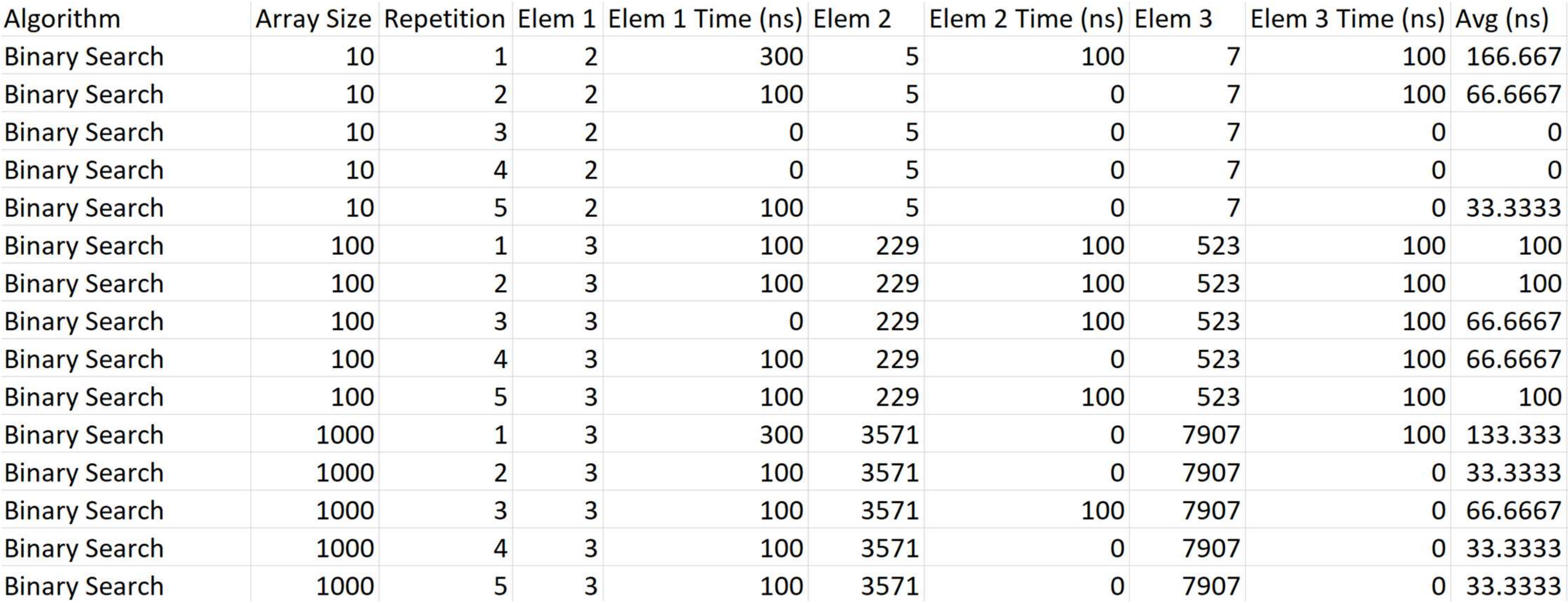
- SSH into VM, download repo, install g++, compile, run



```
swurtman@Linux-Test-Bench:~$ wget https://raw.githubusercontent.com/m4ngo5/2120-Project/master/sortAlgos.h
--2019-05-05 15:19:56--  https://raw.githubusercontent.com/m4ngo5/2120-Project/master/sortAlgos.h
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.40.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.40.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1252 (1.2K) [text/plain]
Saving to: ‘sortAlgos.h.1’

sortAlgos.h.1                100%[=================================================>]   1.22K  --.-KB/s    in 0s

2019-05-05 15:19:56 (53.3 MB/s) - ‘sortAlgos.h.1’ saved [1252/1252]

swurtman@Linux-Test-Bench:~$ sudo apt-get install g++
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils cpp cpp-6 g++-6 gcc gcc-6 libasan3 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-6-dev
  libgomp1 libisl15 libitm1 liblsan0 libmpc3 libmpfr4 libmpx2 libquadmath0 libstdc++-6-dev libtsan0 libubsan0
  linux-libc-dev manpages manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-6-locales g++-multilib g++-6-multilib gcc-6-doc libstdc++6-6-dbg gcc-multilib make autoconf
  automake libtool flex bison gdb gcc-doc gcc-6-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan3-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx2-dbg libquadmath0-dbg glibc-doc
  libstdc++-6-doc man-browser
The following NEW packages will be installed:
  binutils cpp cpp-6 g++ g++-6 gcc gcc-6 libasan3 libatomic1 libc-dev-bin libc6-dev libcc1-0 libcilkrts5 libgcc-6-dev
  libgomp1 libisl15 libitm1 liblsan0 libmpc3 libmpfr4 libmpx2 libquadmath0 libstdc++-6-dev libtsan0 libubsan0
  linux-libc-dev manpages manpages-dev
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 37.5 MB of archives.
After this operation, 159 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Methodology

- Problem: the VM system time resolution is 100 ns



Algorithm	Array Size	Repetition	Elem 1	Elem 1 Time (ns)	Elem 2	Elem 2 Time (ns)	Elem 3	Elem 3 Time (ns)	Avg (ns)
Binary Search	10	1	2	300	5	100	7	100	166.667
Binary Search	10	2	2	100	5	0	7	100	66.6667
Binary Search	10	3	2	0	5	0	7	0	0
Binary Search	10	4	2	0	5	0	7	0	0
Binary Search	10	5	2	100	5	0	7	0	33.3333
Binary Search	100	1	3	100	229	100	523	100	100
Binary Search	100	2	3	100	229	100	523	100	100
Binary Search	100	3	3	0	229	100	523	100	66.6667
Binary Search	100	4	3	100	229	0	523	100	66.6667
Binary Search	100	5	3	100	229	100	523	100	100
Binary Search	1000	1	3	300	3571	0	7907	100	133.333
Binary Search	1000	2	3	100	3571	0	7907	0	33.3333
Binary Search	1000	3	3	100	3571	100	7907	0	66.6667
Binary Search	1000	4	3	100	3571	0	7907	0	33.3333
Binary Search	1000	5	3	100	3571	0	7907	0	33.3333

Revised Methodology

- Solution: Abandon the VM and use Debian on WSL on my laptop

Computer Specs

Razer Blade Stealth 13.3” (2018)

- Intel Core i7-8550OU (4 Core)
- Processor Speed (1.8 GHz)
- 16GB Ram
- Windows 10 Home
- Tests run in Debian on the Windows Subsystem for Linux (WSL)

Algorithms Used

SEARCHING

- Binary Search
- Linear Search
- Interpolation Search

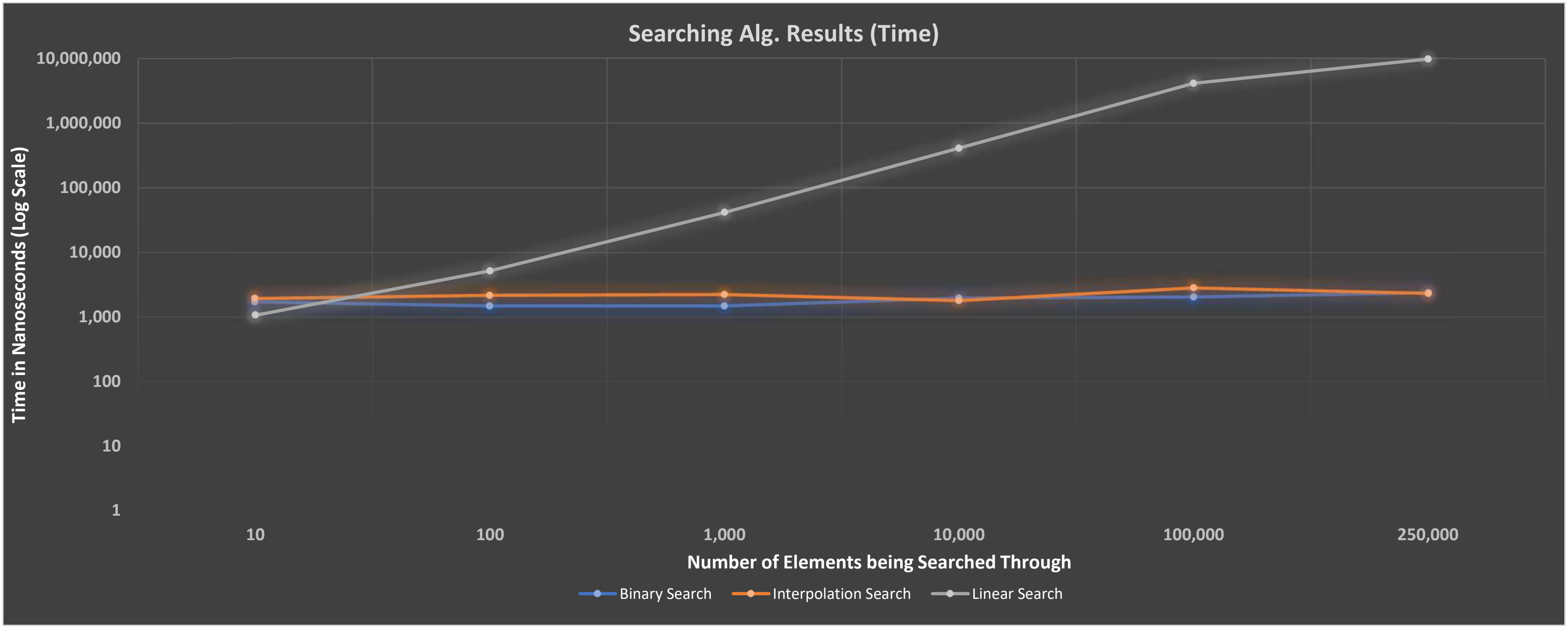
SORTING

- Selection Sort
- Heap Sort
- Quick Sort

Search Algorithm Table

Average of Time (Nanosec.)	Algorithm			
# of Elements	**Binary Search**	**Interpolation Search**	**Linear Search**	**Overall Avg**
10	115	130	71	105
100	100	145	343	196
1,000	100	150	2,755	1,001
10,000	132	121	27,225	9,159
100,000	137	189	273,876	91,401
250,000	159	155	656,168	218,827
Overall Avg	**124**	**148**	**160,073**	**53,448**

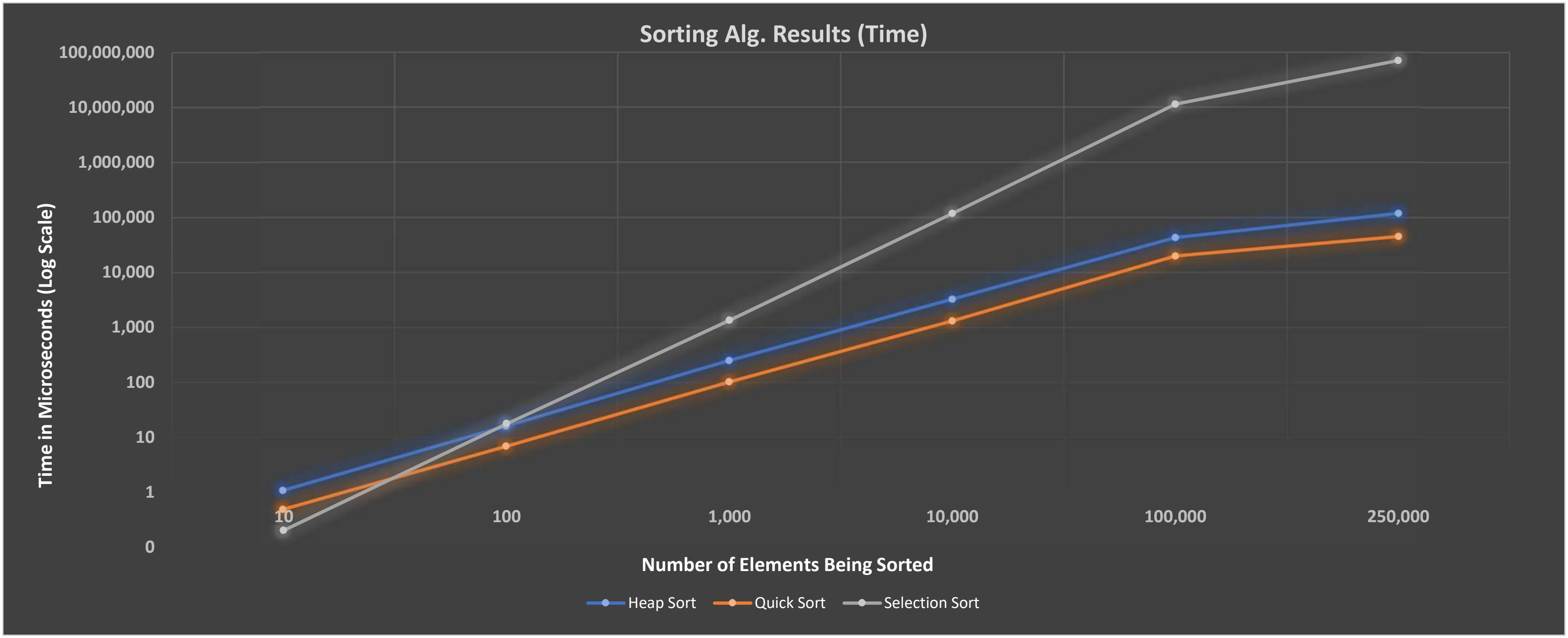
Search Alg. Graph



Sort Algorithm Table

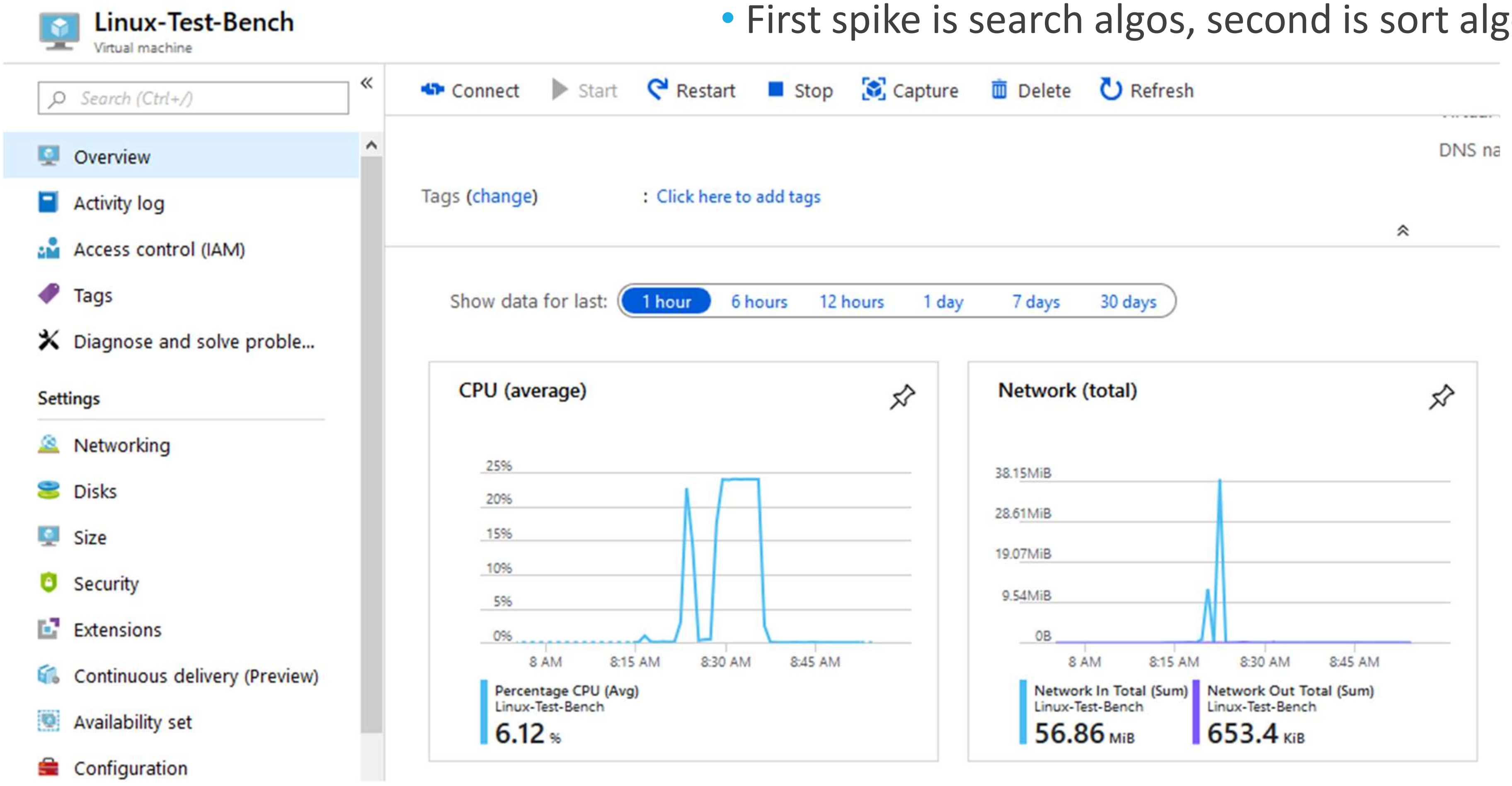
Avg Time (Microsec.)	Algorithm			
# of Elements	Heap Sort	Quick Sort	Selection Sort	Overall Avg
10	1	0	0	1
100	16	7	18	14
1,000	249	103	1,344	565
10,000	3,250	1,314	117,860	40,808
100,000	42,768	19,838	11,462,696	3,841,767
250,000	118,077	44,803	71,953,702	24,038,861
Overall Avg	**27,394**	**11,011**	**13,922,603**	**4,653,669**

Sort Algorithm Graph



Something Interesting

- 25 % CPU = 1 of 4 cores at 100%
- First spike is search algos, second is sort algos



Tags (change) : Click here to add tags

Percentage CPU (Avg)
Linux-Test-Bench
6.12 %

Network In Total (Sum)
Linux-Test-Bench
56.86 MiB

Network Out Total (Sum)
Linux-Test-Bench
653.4 KiB