Virtual Machine Performance over the Cloud

Fangzhou Liu and Shugo Tanaka
Team Iota

Final Presentation - December 14, 2013

How Our Project Evolved

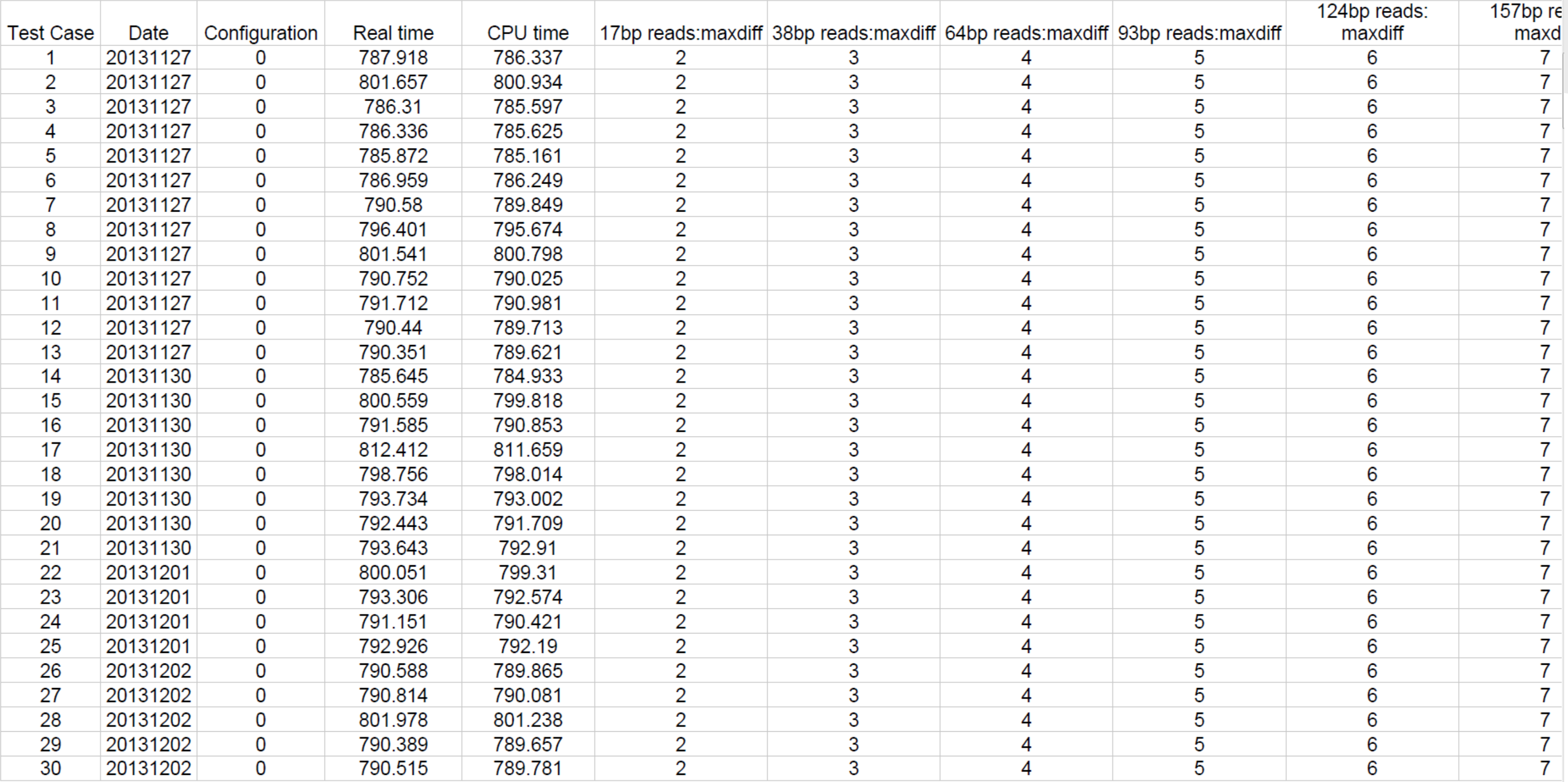
- Began by running a genomic sequencing script over three environments:
  - Baremetal
  - Kernel-based Virtual Machine
  - Linux Containers

- Focused on hypothesis testing bandwidths to buffer size
- And bandwidths and environments to buffer size

Data Collection Methods

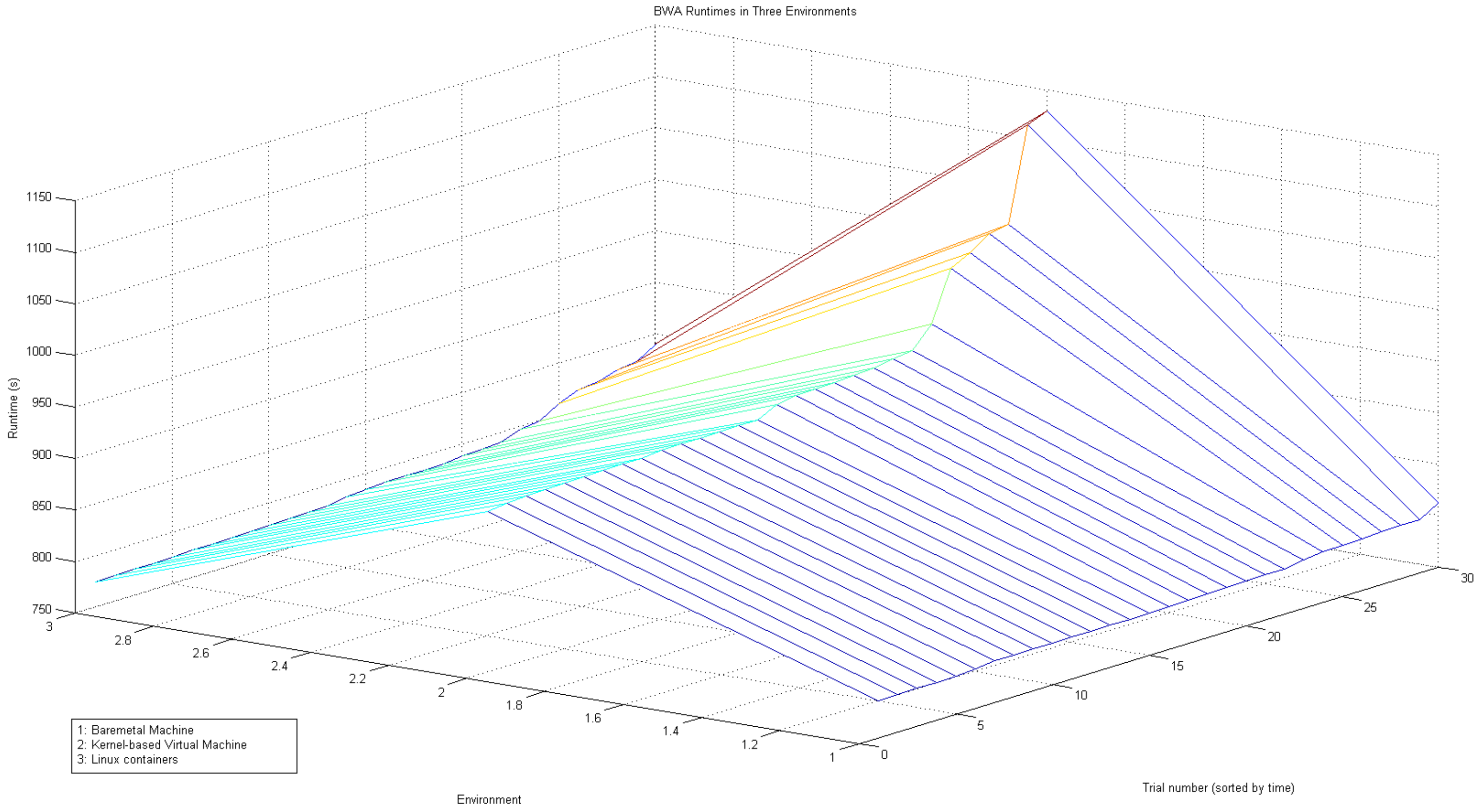
- Burrows-Wheeler Aligner (BWA) for 30 trials on 3 environments
- fstime (a benchmark test from UnixBench)
  - Read, write, and copy on a fixed array
- stream (another benchmark software)
  - copy, scale, add, and triad operations
- fstime with increasing buffer sizes
  - Starting from 256 incrementing by 64 until 8196

Raw Data

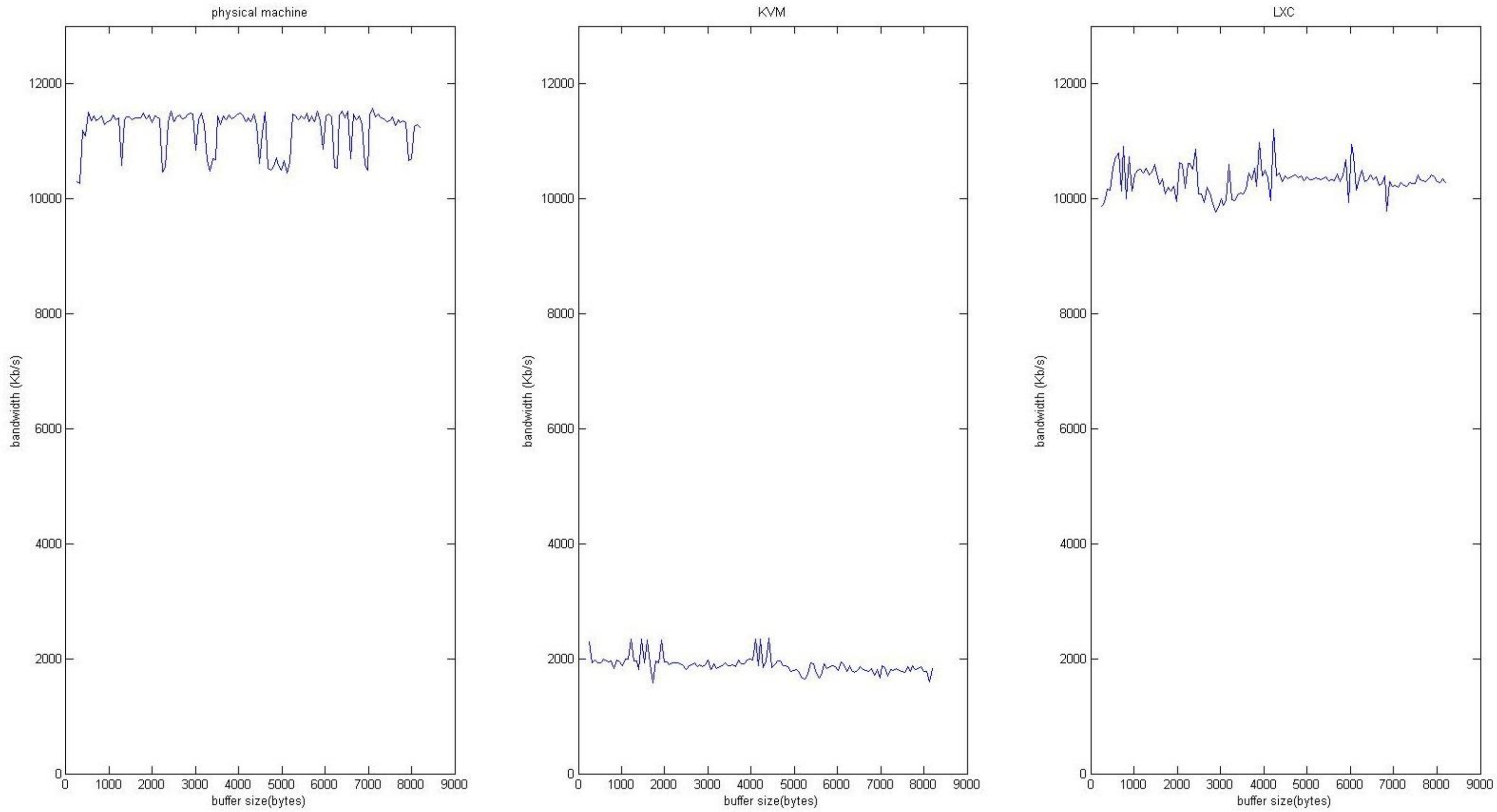


Test Case	Date	Configuration	Real time	CPU time	17bp reads:maxdiff	38bp reads:maxdiff	64bp reads:maxdiff	93bp reads:maxdiff	124bp reads: maxdiff	157bp re maxd
1	20131127	0	787.918	786.337	2	3	4	5	6	7
2	20131127	0	801.657	800.934	2	3	4	5	6	7
3	20131127	0	786.31	785.597	2	3	4	5	6	7
4	20131127	0	786.336	785.625	2	3	4	5	6	7
5	20131127	0	785.872	785.161	2	3	4	5	6	7
6	20131127	0	786.959	786.249	2	3	4	5	6	7
7	20131127	0	790.58	789.849	2	3	4	5	6	7
8	20131127	0	796.401	795.674	2	3	4	5	6	7
9	20131127	0	801.541	800.798	2	3	4	5	6	7
10	20131127	0	790.752	790.025	2	3	4	5	6	7
11	20131127	0	791.712	790.981	2	3	4	5	6	7
12	20131127	0	790.44	789.713	2	3	4	5	6	7
13	20131127	0	790.351	789.621	2	3	4	5	6	7
14	20131130	0	785.645	784.933	2	3	4	5	6	7
15	20131130	0	800.559	799.818	2	3	4	5	6	7
16	20131130	0	791.585	790.853	2	3	4	5	6	7
17	20131130	0	812.412	811.659	2	3	4	5	6	7
18	20131130	0	798.756	798.014	2	3	4	5	6	7
19	20131130	0	793.734	793.002	2	3	4	5	6	7
20	20131130	0	792.443	791.709	2	3	4	5	6	7
21	20131130	0	793.643	792.91	2	3	4	5	6	7
22	20131201	0	800.051	799.31	2	3	4	5	6	7
23	20131201	0	793.306	792.574	2	3	4	5	6	7
24	20131201	0	791.151	790.421	2	3	4	5	6	7
25	20131201	0	792.926	792.19	2	3	4	5	6	7
26	20131202	0	790.588	789.865	2	3	4	5	6	7
27	20131202	0	790.814	790.081	2	3	4	5	6	7
28	20131202	0	801.978	801.238	2	3	4	5	6	7
29	20131202	0	790.389	789.657	2	3	4	5	6	7
30	20131202	0	790.515	789.781	2	3	4	5	6	7

BWA Runtimes



Bandwidth to Buffersize Plots



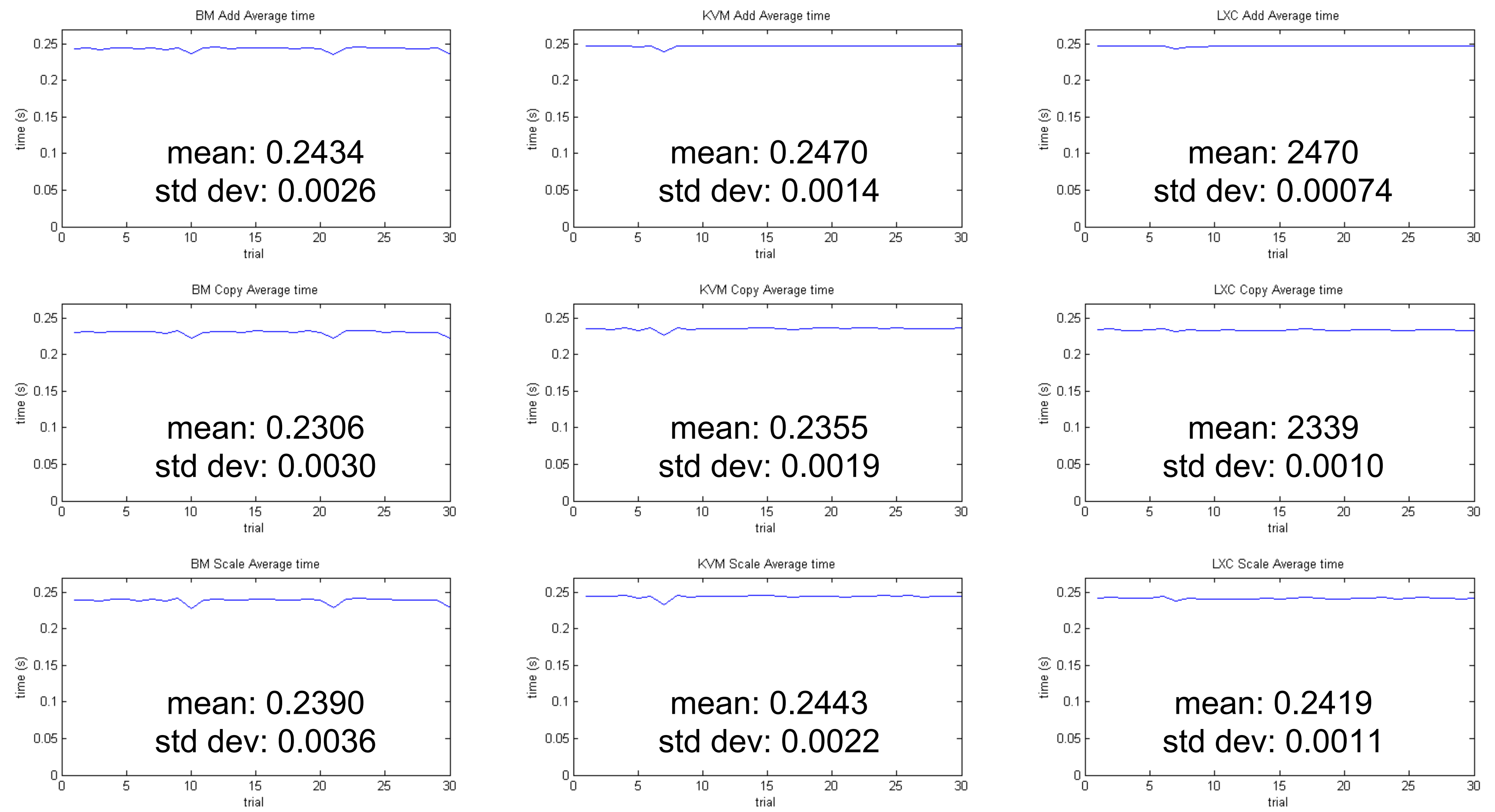
Covariance

- Comparing the buffer size vs. bandwidth

Environment	Covariance	Correlation
Physical Machine	$-2.68*10^4$	Negative
KVM	$-1.61*10^5$	Negative
LXC	587	Positive

$$\sigma(x,y) = \mathrm{E}\left[(x-\mathrm{E}[x])(y-\mathrm{E}[y])\right],$$

Stream: Add, Copy, Scale



BM Add Average time

mean: 0.2434
std dev: 0.0026

KVM Add Average time

mean: 0.2470
std dev: 0.0014

LXC Add Average time

mean: 2470
std dev: 0.00074

BM Copy Average time

mean: 0.2306
std dev: 0.0030

KVM Copy Average time

mean: 0.2355
std dev: 0.0019

LXC Copy Average time

mean: 2339
std dev: 0.0010

BM Scale Average time

mean: 0.2390
std dev: 0.0036

KVM Scale Average time

mean: 0.2443
std dev: 0.0022

LXC Scale Average time

mean: 0.2419
std dev: 0.0011

Hypothesis Testing

- Separated buffers into 8 intervals.
- Sorted our buffers randomly and divided into two groups: training and testing.
- H1: bandwidth is above a certain threshold (mean bandwidth)
- H0: bandwidth is below that threshold
- Found false alarms and missed detections
- Successes and probability of error

Hypothesis Testing 1

Given bandwidth, determine the probability of a buffer size.

Success Percentage	64.52%
Probability of false alarms	38.46%
Probability of missed detections	36.11%
Probability of Error	37.10%

Hypothesis Testing 2

Given bandwidth, predict probability of a given environment and buffer size.

Success Percentage	100%
Probability of false alarms	0%
Probability of missed detections	0%
Probability of Error	0%

Conclusions

- Baremetal machines had the fastest average runtime in BWA
- Computational performance across environments were fairly constant
- KVM had roughly six times lower bandwidth across all buffer sizes
- Given the environment, you can know with certainty if the bandwidth is above or below a mean threshold.

Suggestions for Future Projects

1) Better tutorials on R and Matlab

2) Mini-Projects were pretty well structured. But better datasets or tools for data collection would be helpful for the final project.

Thank you!