FERMILAB-POSTER-22-162-SCD-V

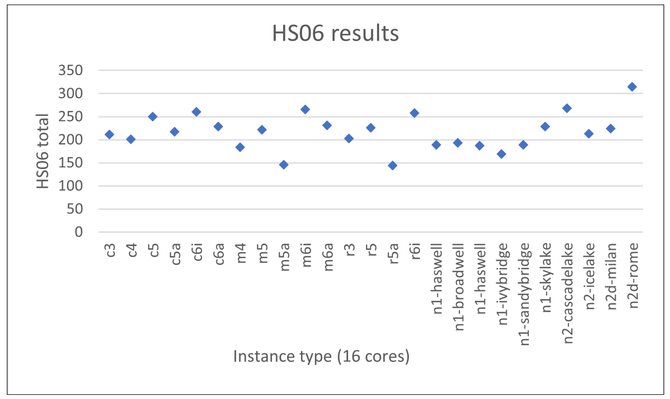
Evaluation and Optimization of Cloud Resources

Shrijan Swaminathan, Purdue University, DOE OMNI Intern with HEPCloud

Background

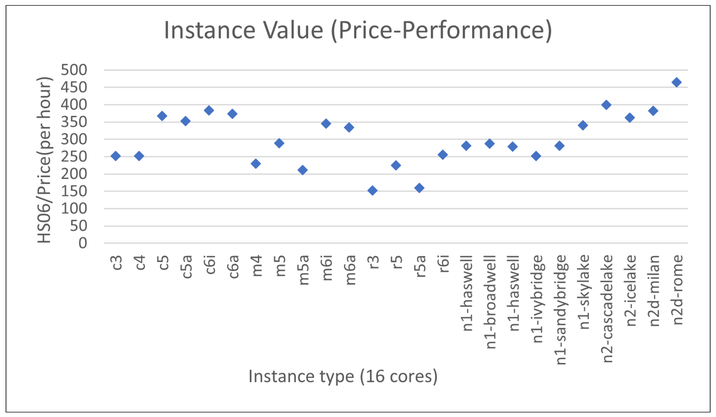
- HEPCloud uses resource provisioning systems to allow experiments, such as DUNE or CMS access to a vast amount of computing resources in an efficient way.
- Amazon and Google auctions virtual machines as instances and HEPCloud takes advantage of this to lower computing costs for scientists.
- HEPCloud's strategy relies on the benchmarking of the available machine types and on heuristics to optimize the bidding on the spot price marketplace.
- This project aims to update the current strategy by benchmarking the new resource types and determining improvements
- The project uses distributed computing systems, deploy services using virtual machines (OpenStack), run benchmarks on these Google Cloud and AWS EC2 machines, and research setups to optimize costs.

Results



Raw HS06 results (on top) and HS06 Compared with On-Demand pricing (on the right)

- The benchmark used is the HepSpec06 benchmark (HS06 for short)
- I ran these tests on the raw computing power of the virtual machines.
- The graph on left shows the HS06 data that computes raw data and the graph below is the results for the price to performance ratio of the instances



- The results overall trend higher towards higher CPU generations (indicated by the number in the instance names)
- This means that it is more efficient to purchase newer instance types

PURDUE UNIVERSITY College of Engineering

HEPCloud

Spot Instances

- Due to the way Spot Instances work, the way to choose flavors of virtual machines differs due to demand
- As a result, we formerly used an algorithm known as Demandx25 where we would compare prices to earlier timings to “predict” when the instance would be pre-empted or stopped by AWS or Google Cloud.
- We updated this algorithm to Demandx50 to adjust for increased pricing over the years.

Amazon	N_COR	CORE TYPE	Speed(GHz	$ per hou	HS06 per cor	HS06 tota	HS06 per $/
c3	16	Xeon E5-2680	2.80	0.840	13.2	212	252
c4	16	Xeon E5-2666	2.90	0.796	12.6	201	252
c5	16	Xeon Platinum 8275CL	3.00	0.680	15.6	250	368
c5a	16	AMD EPYC 7R32	3.3*	0.616	13.6	217	353
c6i	16	Xeon Platinum 8375C	2.90	0.680	16.3	260	383
c6a	16	AMD EPYC 7R13	2.0*	0.612	14.3	229	374
m4	16	Xeon E5-2686	2.30	0.800	11.5	184	229
m5	16	Xeon Platinum 8259CL	2.50	0.768	13.9	222	289
m5a	16	AMD EPYC 7571	2.1*	0.688	9.1	146	212
m6i	16	Xeon Platinum 8375C	2.90	0.768	16.6	266	346
m6a	16	AMD EPYC 7R13	2.0*	0.691	14.4	231	334
r3	16	Xeon E5-2670	2.50	1.330	12.7	203	153
r5	16	Xeon Platinum 8259CL	2.50	1.008	14.2	226	225
r5a	16	AMD EPYC 7571	2.1*	0.904	9.0	144	160
r6i	16	Xeon Platinum 8375C	2.90	1.008	16.1	258	256

Google	N_COR	CORE TYPE	Speed(GHz	$ per hou	HS06 per cor	HS06 tota	HS06 per $/
n1-broadwell	16	Intel Xeon CPU	2.20	0.672	11.8	189	281
n1-haswell	16	Intel Xeon CPU	2.30	0.672	12.1	193	287
n1-ivybridge	16	Intel Xeon CPU	2.50	0.672	11.7	187	279
n1-sandybridge	16	Intel Xeon CPU	2.60	0.672	10.6	169	252
n1-skylake	16	Intel Xeon CPU	2.00	0.672	11.8	189	281
n2-cascadelake	16	Intel Xeon CPU	2.80	0.672	14.3	229	340
n2-icelake	16	Intel Xeon CPU	2.60	0.672	16.8	268	399
n2d-milan	16	AMD EPYC 7B13	2.50	0.587	13.3	213	363
n2d-rome	16	AMD EPYC 7B12	2.25	0.587	14.0	224	382
t2d-milan	16	AMD EPYC 7B13	2.50	0.677	19.7	315	465

Raw data for each instance type with AWS(on top) and Google Cloud (on the bottom)

Acknowledgements

Thank you to all my mentors, supervisors, and other staff for creating a collaborative and productive environment possible for this summer.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.