AMSI SUMMER RESEARCH SCHOLARSHIPS 2024–25

Get a taste for Research this Summer

Optimum Cloud Computing

Toby Mew

Supervised by Prof. Andreas Ernst

Monash University

Jane Street® AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

A computationally demanding program was executed on four high performance cluster computing servers. Measurable differences in performance of the program between servers were observed, which can be attributed to the differing specifications of each machine, although results were variable. Program execution time generally improved with access to a greater number of CPU cores, and optimal performance for this particular program is proposed to occur when tasks are allocated across six CPU cores. Execution time increased with the number of CPU cores available for processing when using than six cores, although the measured runtime was still better than using only one core. Only a small proportion of the program appeared to effectively utilise parallel processing, so further benchmark testing with a greater sample size and with a candidate program that is more parallelisable is recommended.

1 Introduction

Cloud computing services offers a server-based network of computational resources such as processing power and data storage for users to access on demand, enabled through virtualisation technologies. Public cloud providers such as AWS, Google and Microsoft offer these resources through a pay-as-you-go (PAYG) pricing model, where clients pay only for the time they use on the specific resources that they have recruited, with no additional upfront costs.

Parallel processing involves computational tasks being allocated to different CPU cores across one or multiple different virtual machines (VMs), which can then be handled simultaneously enabling many different possible task allocation scenarios. In order for a computational model or program to take advantage of parallel processing it must be composed of smaller tasks that are independent insofar as the outputs of one task are not required for other tasks to proceed [1].

The flexibility of task allocation and the potential to introduce more cost effective computation and data management options, together with relaxing the requirement for investment in their own IT infrastructure, makes cloud computing a highly attractive option to businesses. These capabilities may be highly beneficial to organisations that manage and model with large quantities of data, such as the Australian Bureau of Statistics (ABS). This research in particular is motivated by the problem of optimising cloud resource allocations for cost effective statistical production [2]. In this paper, the performance of a computationally intensive program when executed with up to ten CPU cores on four different cluster computing servers is evaluated, with a view towards proposing an informal definition of the optimisation problem of minimising cost of execution under hard time constraints.

1.1 Statement of Authorship

The motivation for this research is based on work done by Ryan Covey and Andreas Ernst in [2], which is summarised in Section 2. The Python scripts used for benchmarking were obtained from [3]. All other work not referenced in the bibliography is my own, however I particularly acknowledge and appreciate the assistance and direction that Andreas has offered me in producing it.

This research was supported in part by Monash eResearch and eSolutions-Research Support Services through the use of the MonARCH HPC Cluster.

2 The Problem of Optimal Cloud Computing

The problem of optimal cloud computing is well described by Covey in [2], and is summarised in this section.

Consider a computationally expensive model composed of multiple smaller independent tasks which are not necessarily homogenous in their dependencies, complexity or outputs. Consider also a variety of VMs with which we can compute this model, all of which differ in specifications and cost. We are presented with the problem of deciding which of the available VMs to recruit and how to allocate independent tasks across these VMs in order to minimise cost and runtime, keeping in mind the differences in performance and cost that are associated with each potential arrangement.

Evaluating the optimal task allocation by computing our model on each available VM would expend more time and money that it would save. Instead, one strategy is to use a related but cheaper-to-compute model. This model is executed on our available VMs under different task deployment scenarios to obtain a cost and runtime evaluation of each task, the results of which are extrapolated via regression to estimate the overall cost and runtime of our more expensive model on a variety of VMs.

An example of this would involve feeding the original target model a smaller subset of the dataset intended for modelling, which obtains benchmark tests that are cheaper-to-compute. If increasing the size of data input does not add additional burden to the individual computations being performed in the target model – that is, the model should use the same amount of computational work for each computation regardless of the amount of data to be processed – then the cost and execution time of the benchmark would be proportional to the size of the original dataset used, enabling easy extrapolation via regression to obtain overall cost and runtime.

Generally, an optimisation problem would seek to minimise cost under the hard constraints of a runtime deadline. In this study, only the latter is considered.

2.1 Methodology

Performance was taken as a measurement of CPU time and overall execution time. Computation time can also be affected independently of VM specifications by delays due to virtualisation and task allocation overheads; data transfer speeds; and network latency, thus affecting the price-per-use on public clouds [5]. Costs associated with data storage and data transfer between public and private clouds were not taken into account for this report.

A machine learning program based on the PyTorch library was used to produce benchmarking results [3]. This program contained the functions generate_sine_wave.py, which generates sine waves under different phases; and train.py, which predicts future signal values from these initialised signals. This program was selected for its ability to utilise parallel processing, with training steps theoretically being executable as independent computational tasks on different CPU cores; and for its computational demands on the hardware which would allow for measurable differences in performance to occur.

This program was executed using four Monash Advanced Research Computing Hybrid (MonARCH) servers, each of which were supported by one of Intel's Xeon-Gold-5220R, Xeon-Gold-6150, Xeon-Gold-6338 or Xeon-Platinum-8260 processors (Appendix 1), which are second, first, third and second generation processor respectively. For these reasons, the servers are hereafter referred to in text as G5220R, G6150, G6338 and P8260. The G5220R server is baremetal, and the others are VMs. The program was executed within one server using a specific number of available CPU cores for processing multiple threads. This was repeated for between one and ten CPU cores on all four servers in five separate batch submissions; i.e. obtaining five separate data points for each of the 40 specific combinations of server type and number of cores, or 200 data points in total. The first three batches were executed in succession, and the next two batches were executed in succession in a separate session. Data with respect to execution runtime, CPU time, memory usage, page faults and context switches were recorded from which the results are derived (Appendix 2).

3 Results

Peak memory usage was higher for the first three batch submissions of the program (median=1753728kB, n=120) than for the next two batch submissions (median=1566088kB, n=80) (Fig. 1). The peak memory usage of the program was marginally higher when using three to seven cores (median=1761212kB for batch 1, 1573718kB for batch 2) than for one, two, eight, nine or ten cores (median=1745652kB for batch 1, 1564014kB for batch 2) with a difference of 15 560kB for batch one and 9 704kB for batch 2.

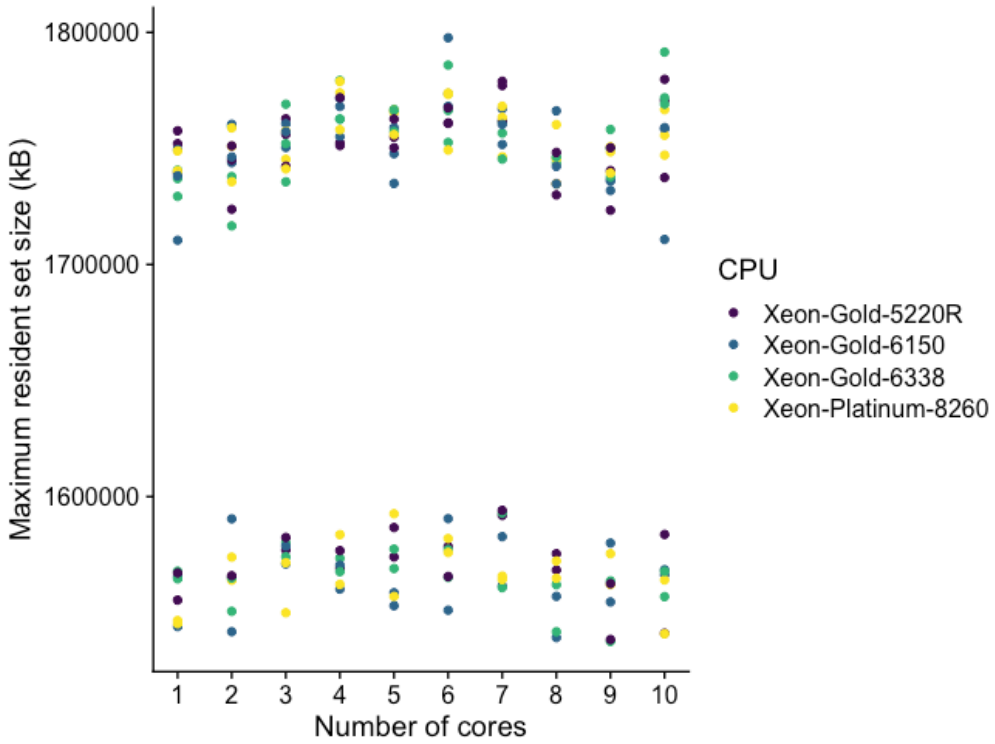

Figure 1: Peak memory usage by number of cores used by the program (n=200).

Using only one CPU core, there were observable differences in performance between servers, with the G6338 server exhibiting both a median CPU time (470.24s, range 465-478.78s) and median total execution runtime (497.1s, range 477-528.11s) lower than for all other processors (Figs. 2 and 3). G5220R exhibited the highest median CPU time (647.5s, range 611.6-676.43s) and median total execution time (664.44s, range 627.2-738.8s) with one core, although this was not significantly different to the G6150 or the P8260 servers. The median CPU and execution times for G6338 were significantly lower than that for all other processors for all number of cores.

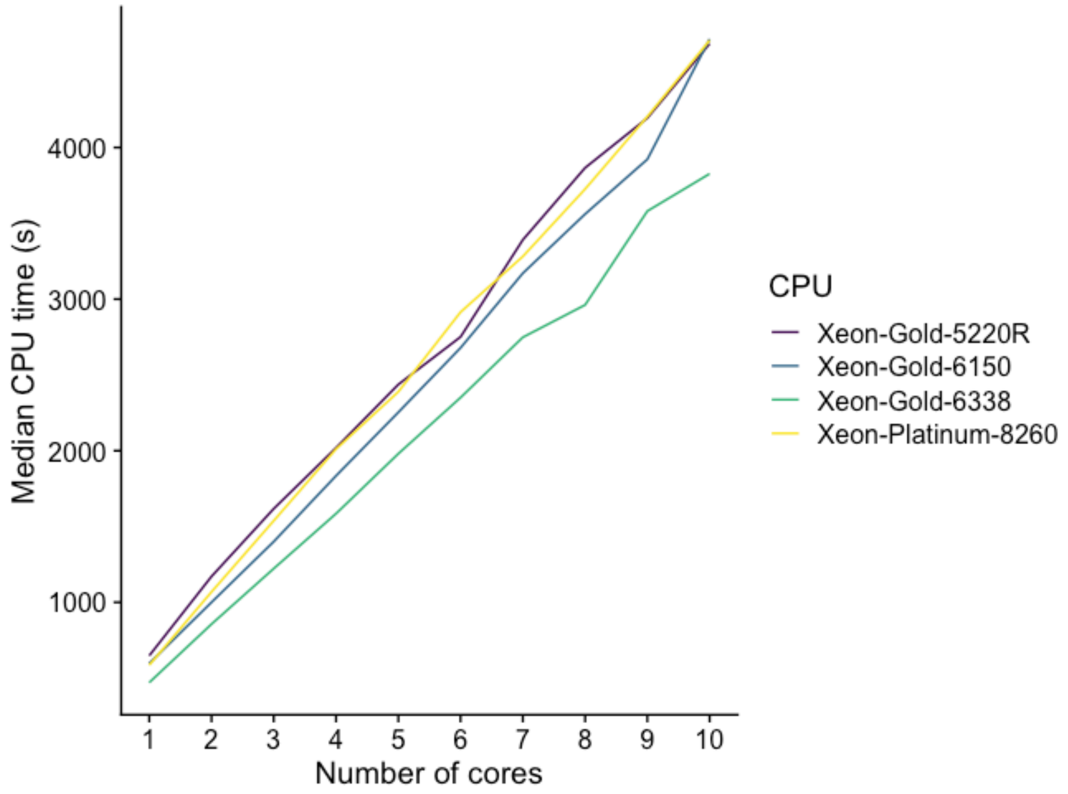

Figure 2: Median CPU time by number of cores used by the program.

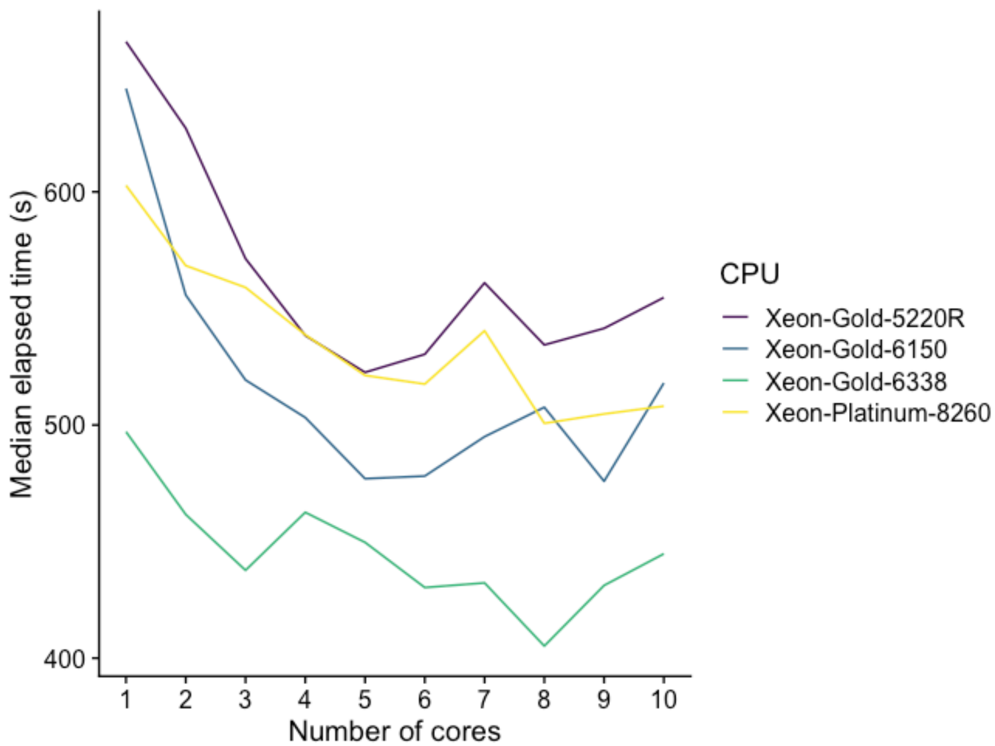

Figure 3: Median execution time by number of cores used by the program.

An increase in the number of cores corresponded with an increase in CPU time. On average, adding an additional core increased the measured CPU time by 373 seconds on the G6338 server; 448 seconds for G5220R; and 458 seconds for both G6150 and P8260 (Fig. 2).

Increasing the number of available cores for up to and including six cores corresponded with a decrease in execution time, however continuing to add to more than six cores resulted in a slight increase in runtime (Fig. 3). These measurements were fairly variable for more than six cores (Figs. 4 and 5), so it is difficult to assess the relationship for a large number of cores. The lowest median execution time for the G5220R server was 523 seconds with five cores; 475.89 seconds for G6150 with nine cores; 405.2 seconds for G6338 with eight cores; and 500.72 seconds for P8260 with eight cores. For all servers, the highest median runtime was obtained using only one core.

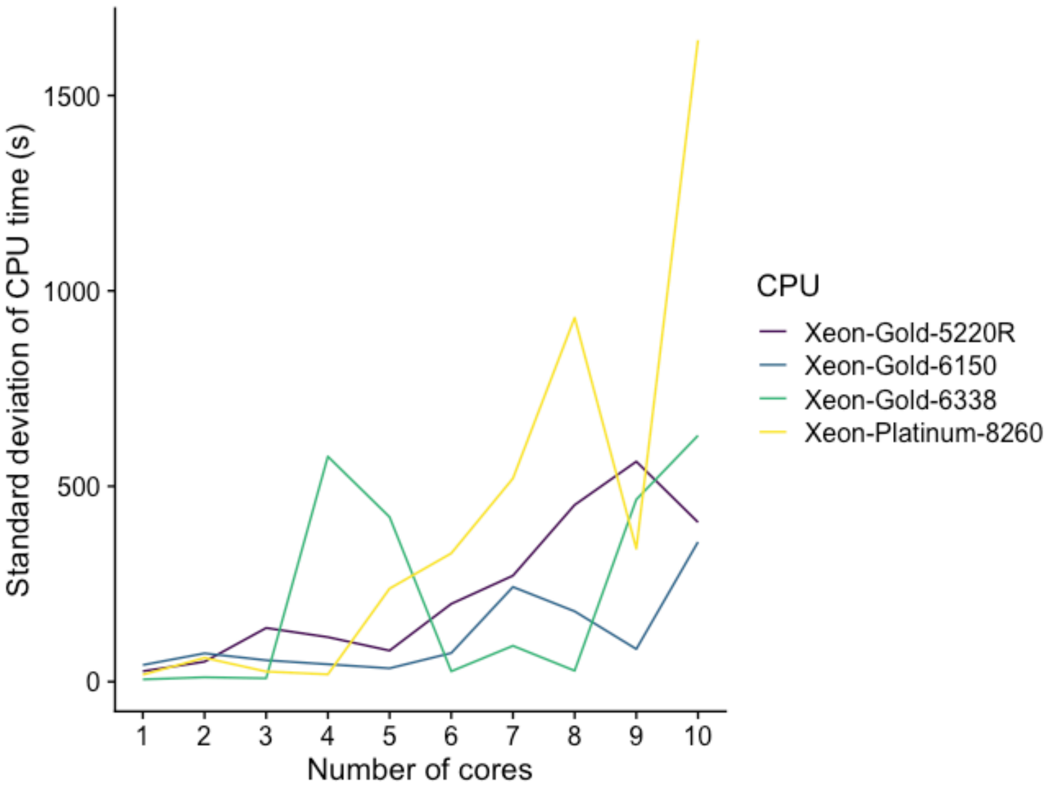

Figure 4: Standard deviation of CPU time by number of cores used by the program.

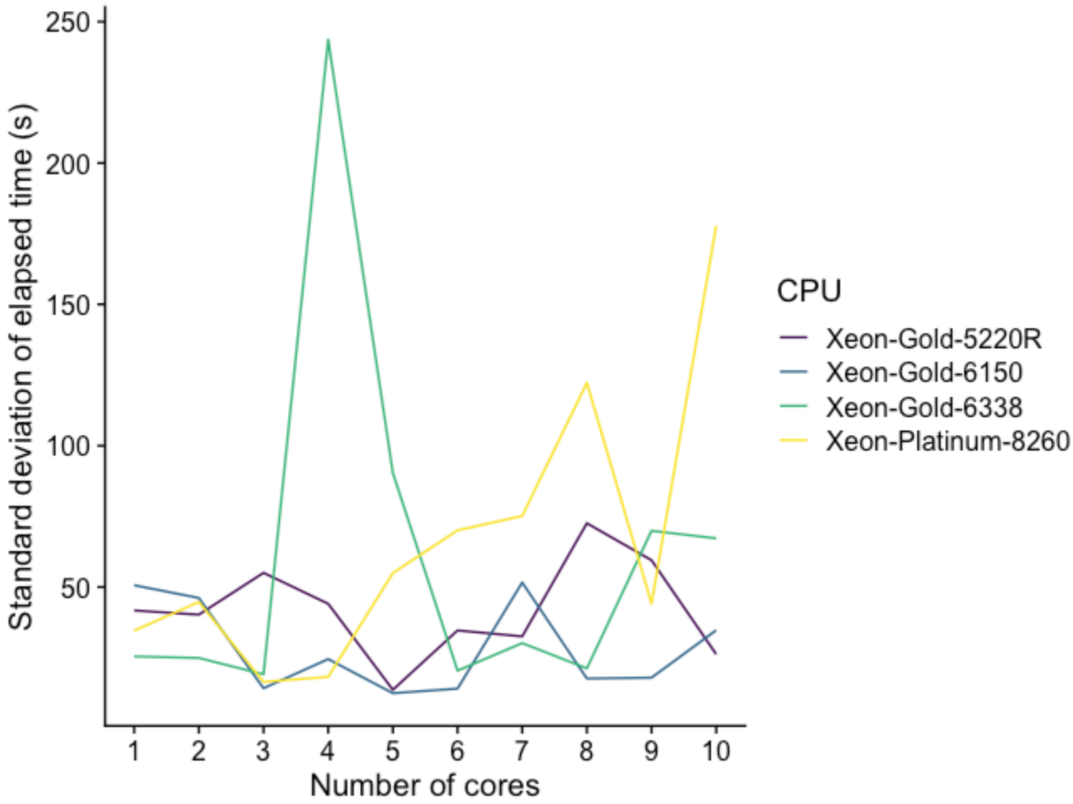

Figure 5: Standard deviation of execution time by number of cores used by the program.

3 Discussion and Conclusion

This study demonstrated that running a particular machine learning program on different VMs produces observable differences in performance, likely due to the differing hardware specifications of each VM. The G6338 server equipped with the Xeon-Gold-6338 processor performed significantly better than all other servers when using the same number of CPU cores. This particular VM was equipped with the most recent generation of processor, so this improvement in performance was expected. The baremetal server G5220R performed minimally worse than the G6150 and G8260 VMs, despite housing a more recent generation of processor than the G6150.

Increasing the number of cores utilised by the program increased the total CPU time at an unacceptably high rate for all VMs, likely due to high CPU idle time indicating that the program used was inefficient in performing parallel computations across multiple cores simultaneously. Many of the computational tasks may not have been independent but were instead inherently sequential in nature, emphasising the importance of thorough benchmarking of the desired application to determine that it does indeed take advantage of parallel processing to reduce execution time [1].

This is further supported by the relatively small improvements in overall program runtime when given more CPU cores with which to perform computations. Simplistically, an optimally parallelisable program would halve execution time with a doubling of available processors. In reality, this is difficult to achieve [4]. The G6338 VM exhibited only an 18.5% best improvement in runtime when using eight cores compared to one, hardly close to even halving the overall runtime. In the added context of a large statistical model which may take many days to run, these improvements in performance would come at an unacceptably high additional cost in wasted CPU time. The trade-off of large additional cost for relatively minimal gains in performance would rarely be an option for most clients unless in cases of extreme time deadline constraints, and would generally not be a cost-effective option.

Performance was improved with an increase in CPU cores, and generally did not improve further beyond six cores for all VMs, indicating deployment of this particular program across six CPU cores in total over one or more VMs would likely yield the best performance. The overhead computational costs of managing multiple processing threads and load imbalances may be offsetting any gains made in using more processors, as supported by the slight increase in execution times when using greater than six cores (Fig. 3) [4]. As execution of all tasks within the program was contained to one VM for each test, it is possible that deployment of tasks across multiple VMs may yield a lower CPU and/or execution time depending on the requirements of the specific computational task and its suitability to a particular VM's capabilities.

3.1 Further Study

This study was successful in evaluating the performance of the chosen computational program on a variety of VMs. However, the lack of major improvements in performance when making extra processing power available suggests that the program utilised mostly linear processing, and as such is not a good candidate for further investigation. More interesting results could be yielded by using a Monte Carlo simulation, as this type of model is composed of many independent iterations [8]. As this research was motivated by statistical production, direction for future work could also involve creating a 'metadataset' from publicly available survey data and applying a small area estimation model. The Rao-Yu spatio-temporal model is a candidate example [6].

Further, a greater sample size of tests than what was completed in this research is needed to accurately assess the performance of different VMs. For example, RAM size differs between VMs offered by public cloud providers, thus the uncertainty in the true memory requirements of the program in Fig. 1 presents issues in formulating the memory constraint of the optimisation problem.

This study evaluated the performance of several VMs on a private cluster computing network which is not commercially available, hence commercial cost estimations could not be obtained. After obtaining benchmark measurements from executing a candidate program on a private cloud, cost estimations should be obtained by executing the program on a public cloud to provide input data for a developed optimisation problem.

The ideal candidate program should satisfy the scalability requirements discussed in Section 1.1.

4 References

[1] Adams NM, Kirby SPJ, Harris P and Clegg DB (1996) *A Review of Parallel Processing for Statistical Computation*, Statistics and Computing, 6:37-49, doi:10.1007/BF00161572

[2] Covey, R (3 September 2024) *The Problem of Optimal Cloud Computing for Statistical Production* [unpublished manuscript], received 2 October 2024.

[3] Fu Z and Chintala S (6 April 2017) *Time Sequence Prediction* [source code], GitHub, accessed 27 November 2024. https://github.com/pytorch/examples/tree/main/time_sequence_prediction

[4] Li K, Pan Y, Shen H and Zheng SQ (1999) *A Study of Average Case Speedup and Scalability of Parallel Computations on Static Networks*, Mathematical and Computer Modelling, 29(9):83-94, doi:10.1016/S0895-7177(99)00083-7

[5] Malawski M, Figiela K and Nabrzyski J (2013) *Cost Minimization for Computational Applications on Hybrid Cloud Infrastructures,* Future Generation Computer Systems, 29(7):1786-1794, doi:10.1016/j.future.2013.01.004

[6] Rao JNK and Yu M (1994) *Small-Area Estimation by Combining Time-Series and Cross-Sectional Data*, The Canadian Journal of Statistics, 22(4):511-528, doi:10.2307/3315407

[7] Thai LT (2017) *Optimising the Usage of Cloud Resources for Executing Bag-of-Tasks Applications* [PhD thesis], University of St Andrews. Accessed 13 January 2025 from http://hdl.handle.net/10023/15642

[8] Tricomi G, Merlino G, Panarello A and Puliafito A (2020) *Optimal Selection Techniques for Cloud Service Providers*, IEEE Access, 8:203591-203618, doi:10.1109/ACCESS.2020.3035816

5 Appendix

Some additional hardware specifications and results are provided below.

Name	CPU	Number of Cores / Server	Usable Memory / Server
G6150	Xeon-Gold 6150 @ 2.70GHz	36	158893MB
G5220R	Xeon(R) Gold 5220R @ 2.2GHz	48	735000MB
P8260	Xeon-Platinum-8260 @ 2.50GHz	48	342000MB
G6338	Xeon-Gold-6338 @ 2.00GHz	64	505700MB

Appendix 1: Available servers in the MonARCH high performance computing cluster

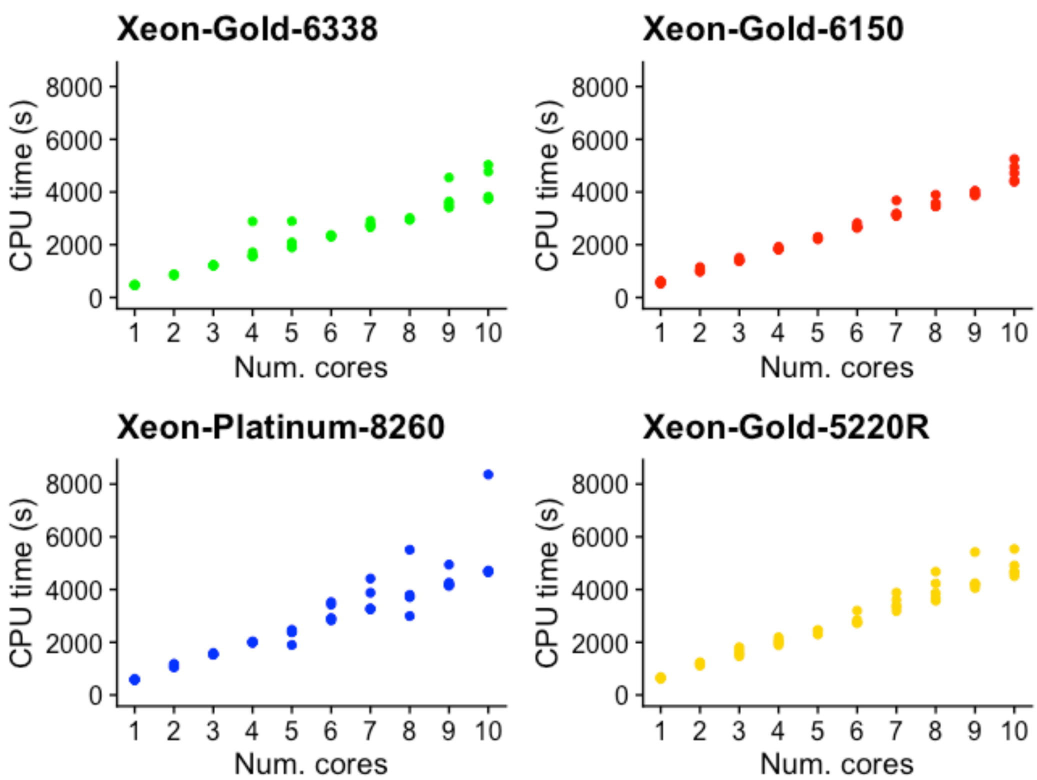

Figure 6: Total CPU time when using up to ten CPU cores for four Intel processors

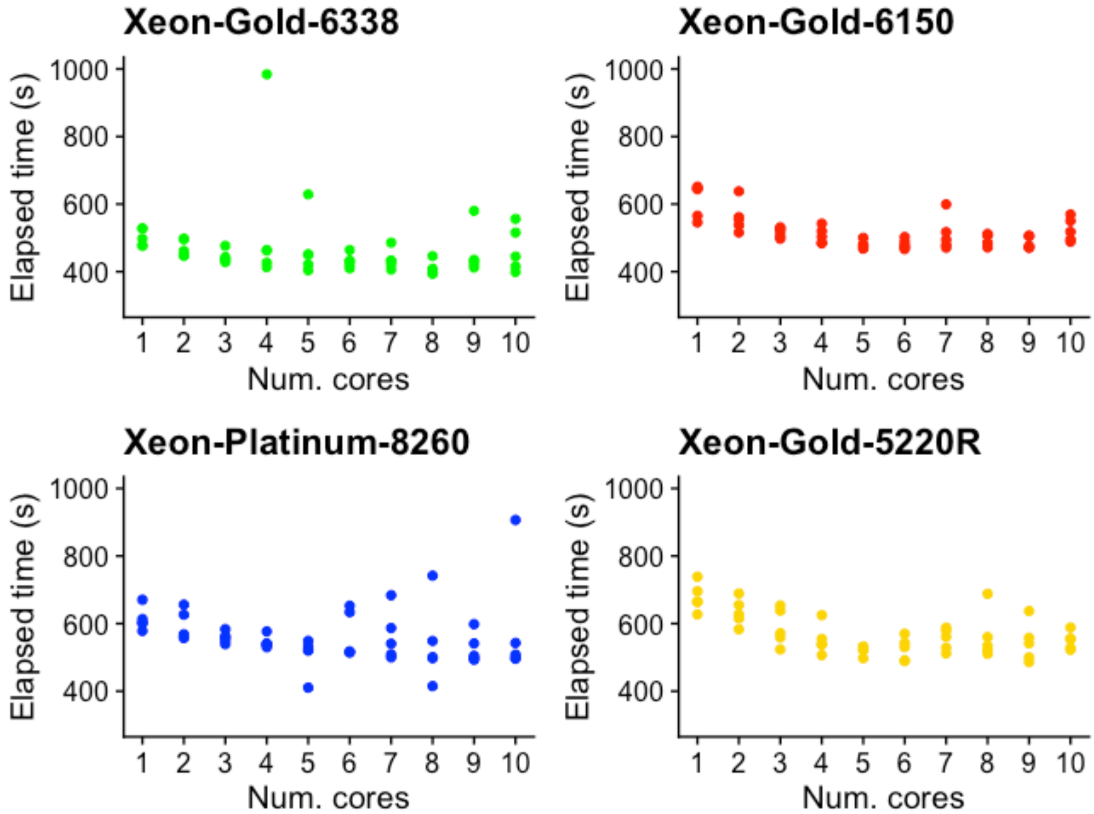

Figure 7: Total elapsed time when using up to ten CPU cores for four Intel processors

Node	Num. cores	CPU time (s)	Sys. Time (s)	CPU utilisation	Max. resident set size (kB)	Major page faults	Minor page faults	Voluntary context switches	Involuntary context switches
Xeon-Gold-6150	1	551	3.1	98%	1710420	16315	458296	26492	5984
Xeon-Gold-6150	2	1059.42	13.15	193%	1743856	15831	5429452	26313	14116
Xeon-Gold-6150	3	1512.1	12.63	287%	1750396	16489	5462551	29872	1894
Xeon-Gold-6150	4	1834.2	12.59	381%	1768068	16070	5498751	29882	1785
Xeon-Gold-6150	5	2252.94	12.69	475%	1747696	16936	5615735	31352	2010
Xeon-Gold-6150	6	2643.72	13.48	569%	1773580	16423	5487194	32302	5839
Xeon-Gold-6150	7	3135.41	11.77	668%	1767028	1351	4995204	12940	2383
Xeon-Gold-6150	8	3896.19	20.63	765%	1766188	2223	8515612	60844	7530
Xeon-Gold-6150	9	4036.8	12.63	857%	1736528	1426	5405838	14260	5651
Xeon-Gold-6150	10	5250.27	21.04	958%	1758296	2217	8886230	31568	8715
Xeon-Gold-6338	1	465.29	2.65	98%	1740644	16990	586051	31865	1290
Xeon-Gold-6338	2	849.98	7.19	192%	1759408	16832	5067452	31988	1358
Xeon-Gold-6338	3	1211.17	9.12	285%	1769040	16438	5262324	79588	2038
Xeon-Gold-6338	4	1558.89	7.67	379%	1762600	16822	5015214	32808	2213

Xeon-Gold-6338	5	1899.31	7.48	472%	1766816	16419	4838838	33650	1833
Xeon-Gold-6338	6	2299.71	11.54	564%	1752588	17069	6367203	123289	8462
Xeon-Gold-6338	7	2663.98	7.79	658%	1761464	17465	5004581	36792	4086
Xeon-Gold-6338	8	2961.52	8.95	751%	1745860	16421	5595362	38069	3181
Xeon-Gold-6338	9	3466.22	8.71	843%	1758156	16319	5323173	34447	8100
Xeon-Gold-6338	10	4772.53	16.8	929%	1791500	16316	7135651	143011	13311
Xeon-Gold-5220R	1	611.6	3.06	98%	1752044	16582	495675	26199	1725
Xeon-Gold-5220R	2	1110.4	8.82	192%	1723772	15915	5735934	26075	2048
Xeon-Gold-5220R	3	1813.24	13.01	286%	1762756	16004	7690833	38986	10797
Xeon-Gold-5220R	4	2018.66	9.91	377%	1752460	16593	5787476	27815	2595
Xeon-Gold-5220R	5	2441.84	9.97	470%	1754868	16029	5721124	28410	5573
Xeon-Gold-5220R	6	3200.45	20.46	565%	1760960	15963	11885752	42328	18081
Xeon-Gold-5220R	7	3336	11.29	654%	1761860	16813	5674249	30869	2480
Xeon-Gold-5220R	8	3866.05	12.42	746%	1734828	16546	6179071	30157	4644
Xeon-Gold-5220R	9	4193.4	12.21	841%	1723360	17132	5985699	30409	3652
Xeon-Gold-5220R	10	4911.09	12.62	932%	1770616	16923	6151319	30740	8386
Xeon-Platinum-8260	1	563.75	3.03	98%	1740336	16988	470416	26512	1515
Xeon-Platinum-8260	2	1068.14	9.19	192%	1750908	16265	5451321	26986	1624
Xeon-Platinum-8260	3	1533.09	9.42	286%	1745340	16720	5343414	29307	1780
Xeon-Platinum-8260	4	1999.47	10.04	379%	1773956	16671	5579938	29877	1897
Xeon-Platinum-8260	5	2445.94	9.64	472%	1766348	16608	5312756	30578	4586
Xeon-Platinum-8260	6	2914.16	10.08	565%	1773884	16631	5337099	30897	2453
Xeon-Platinum-8260	7	3878.2	12.6	663%	1746376	17071	5220353	50625	3309
Xeon-Platinum-8260	8	5507.99	19.28	745%	1734776	17317	6601351	52388	4172
Xeon-Platinum-8260	9	4150.19	11.39	844%	1750668	17371	6149178	35714	4796
Xeon-Platinum-8260	10	4653.69	10.87	940%	1755668	17447	5742076	35266	2734
Xeon-Gold-6150	1	629.64	3.2	98%	1749496	15663	467969	30882	1703
Xeon-Gold-6150	2	983.38	12.13	193%	1760384	16356	5204750	31913	1662
Xeon-Gold-6150	3	1415.61	12.67	287%	1760652	16431	5535897	32225	1756
Xeon-Gold-6150	4	1829.17	12.55	380%	1755072	16375	5281330	33272	2163
Xeon-Gold-6150	5	2260.08	12.49	474%	1758940	16691	5113013	33853	2145
Xeon-Gold-6150	6	2678.5	13.2	570%	1768132	15381	5320971	33843	5951
Xeon-Gold-6150	7	3170.52	13.1	664%	1751688	17095	5406449	32107	4914
Xeon-Gold-6150	8	3560.77	13.98	758%	1734680	16516	5685165	32543	2741
Xeon-Gold-6150	9	4039.14	15.41	852%	1731868	16986	6147687	35411	8930
Xeon-Gold-6150	10	4715.77	18.24	955%	1758956	774	8534793	13437	15962
Xeon-Gold-6338	1	465	2.46	98%	1736992	16906	513459	30586	1275
Xeon-Gold-6338	2	855.66	7.09	192%	1737884	16856	4952098	30835	1327
Xeon-Gold-6338	3	1227.83	7.48	285%	1752092	16983	4888410	30130	2164
Xeon-Gold-6338	4	2884.73	18.46	295%	1779312	16687	10778888	229382	73358

Xeon-Gold-6338	5	1979.75	7.76	471%	1757240	17117	4958075	31418	2685
Xeon-Gold-6338	6	2362.78	8.48	564%	1766344	16133	4917511	42772	4437
Xeon-Gold-6338	7	2756.39	8.55	657%	1756588	16337	5086581	34395	5671
Xeon-Gold-6338	8	2953.82	8.51	752%	1743796	16657	5589160	35121	2232
Xeon-Gold-6338	9	3642.12	14.44	848%	1749976	16550	7052550	50973	3972
Xeon-Gold-6338	10	3732.31	14.55	939%	1771820	17433	9272061	90438	10431
Xeon-Gold-5220R	1	647.5	3.56	98%	1739424	17040	473900	28415	2165
Xeon-Gold-5220R	2	1170.11	10.75	192%	1745068	16115	5810524	28499	2371
Xeon-Gold-5220R	3	1617.36	10.99	285%	1756124	4892	5731679	14547	3743
Xeon-Gold-5220R	4	2074.08	10.96	376%	1751200	17396	5745702	26576	3136
Xeon-Gold-5220R	5	2467.84	10.66	469%	1750288	17441	5680966	27203	7663
Xeon-Gold-5220R	6	2750.72	9.18	564%	1767488	17560	5510550	32992	2898
Xeon-Gold-5220R	7	3880.65	18.08	663%	1777064	748	8499456	48769	10150
Xeon-Gold-5220R	8	4227.59	20.25	758%	1729956	1428	8224375	121889	12199
Xeon-Gold-5220R	9	5421.31	19.97	854%	1740436	1403	7234481	69196	12336
Xeon-Gold-5220R	10	5539.01	21.58	945%	1779760	2050	9190040	74249	11564
Xeon-Platinum-8260	1	585.35	3.1	98%	1739724	17536	526180	30226	1604
Xeon-Platinum-8260	2	1061.13	8.8	192%	1758864	17306	5644132	29913	1621
Xeon-Platinum-8260	3	1594.14	9.58	285%	1757464	16721	5749863	31832	1838
Xeon-Platinum-8260	4	2026.94	9.43	378%	1778856	17345	5448641	31262	2308
Xeon-Platinum-8260	5	1896.85	7.82	464%	1765392	17359	4725399	31859	2012
Xeon-Platinum-8260	6	2895.6	9.72	565%	1773244	17384	5663860	32590	2773
Xeon-Platinum-8260	7	4415.17	16.72	648%	1763360	17384	6603953	34539	3053
Xeon-Platinum-8260	8	3725.82	11.06	751%	1745444	17409	6264946	33745	4554
Xeon-Platinum-8260	9	4258.59	11.51	846%	1748504	17455	6022219	34401	5716
Xeon-Platinum-8260	10	4707.38	10.62	947%	1766692	799	6206212	13343	3031
Xeon-Gold-6150	1	521.42	2.72	96%	1738212	16840	466398	31160	1752
Xeon-Gold-6150	2	999.66	11.8	188%	1746304	16800	5510734	31472	1785
Xeon-Gold-6150	3	1400.36	11.92	279%	1757180	16509	5449342	32175	1971
Xeon-Gold-6150	4	1854.49	12.4	371%	1771564	16542	4981550	31366	4336
Xeon-Gold-6150	5	2297.4	13	462%	1734904	16919	5326997	32340	2129
Xeon-Gold-6150	6	2681.77	13.09	552%	1797648	16921	5224812	33291	2203
Xeon-Gold-6150	7	3179.45	13.27	645%	1760328	16600	5159219	32408	5669
Xeon-Gold-6150	8	3565.76	13.95	737%	1742256	16090	5509993	32798	3088
Xeon-Gold-6150	9	3922.28	14.17	836%	1735912	16710	6076163	30120	4492
Xeon-Gold-6150	10	4452.08	14.05	914%	1710780	17691	5953913	31454	5271
Xeon-Gold-6338	1	470.24	2.01	95%	1729340	17179	471221	29664	1450
Xeon-Gold-6338	2	851.8	6.83	186%	1716628	16771	4937965	29856	1574
Xeon-Gold-6338	3	1210.11	9.27	275%	1735596	17092	5866117	79851	8146
Xeon-Gold-6338	4	1558.38	7.26	367%	1762668	16562	5062407	30979	3877

Xeon-Gold-6338	5	2895.95	15.82	463%	1766376	17294	7113369	106526	7944
Xeon-Gold-6338	6	2350.35	7.69	548%	1785896	17335	4983733	32351	1928
Xeon-Gold-6338	7	2748.03	8.42	637%	1745388	17309	4961953	32948	4146
Xeon-Gold-6338	8	2948.74	8.37	724%	1746332	17332	5579931	33462	3224
Xeon-Gold-6338	9	3412.79	16.49	813%	1737748	16237	8711831	136468	5346
Xeon-Gold-6338	10	5033.26	22.09	909%	1769080	17694	12816399	162820	11126
Xeon-Gold-5220R	1	634.2	3.66	96%	1757584	16530	471177	27876	2060
Xeon-Gold-5220R	2	1168.55	10.75	188%	1751100	16453	5696514	28530	3170
Xeon-Gold-5220R	3	1545.6	9.06	278%	1742312	17259	5329240	30423	2513
Xeon-Gold-5220R	4	1972.43	8.85	368%	1771856	17255	5592907	31059	5115
Xeon-Gold-5220R	5	2436.65	9.04	468%	1762660	17486	5533882	27344	5754
Xeon-Gold-5220R	6	2846.69	9.17	526%	1760820	17438	5161681	29790	12492
Xeon-Gold-5220R	7	3390.94	10.53	588%	1778904	16785	5613986	33829	11283
Xeon-Gold-5220R	8	4678.51	21.01	683%	1748268	17227	9138794	62938	11735
Xeon-Gold-5220R	9	4239.54	12.08	762%	1750300	17685	6168340	32072	6161
Xeon-Gold-5220R	10	4642.47	11.23	839%	1737456	17672	5823201	32751	19567
Xeon-Platinum-8260	1	585.07	3.78	96%	1748912	16743	470597	28834	1817
Xeon-Platinum-8260	2	1059.91	8.61	188%	1735636	17219	5548637	29751	1882
Xeon-Platinum-8260	3	1549.54	10.07	279%	1741176	16211	5630958	29400	1954
Xeon-Platinum-8260	4	1977.86	9.31	369%	1758048	16823	5709294	31356	3547
Xeon-Platinum-8260	5	2388.25	9.59	460%	1755944	16956	5862244	32696	2533
Xeon-Platinum-8260	6	2830.34	9.64	550%	1749284	16612	5467866	34311	2225
Xeon-Platinum-8260	7	3258.01	9.48	642%	1768160	16609	4896793	33615	4866
Xeon-Platinum-8260	8	2995.1	9.99	724%	1760244	17297	5417690	33583	2502
Xeon-Platinum-8260	9	4943.24	29.16	831%	1739348	16854	10668291	138769	12350
Xeon-Platinum-8260	10	8360.59	35.38	926%	1747092	17351	10104919	91113	6458
Xeon-Gold-6150	1	596.31	2.98	92%	1544004	11974	532187	22475	1796
Xeon-Gold-6150	2	987.35	12.08	178%	1541788	11195	4639823	24046	1845
Xeon-Gold-6150	3	1378.29	14.03	264%	1570972	11972	4587140	25273	1854
Xeon-Gold-6150	4	1810.86	14.68	351%	1560148	11494	4829275	25836	97689
Xeon-Gold-6150	5	2224.38	14.78	475%	1558600	1315	4735365	10766	2296
Xeon-Gold-6150	6	2823.65	15.13	565%	1590496	1725	5014059	11753	42586
Xeon-Gold-6150	7	3091.33	16.13	601%	1561504	12325	4916444	22728	46262
Xeon-Gold-6150	8	3462.15	16.44	684%	1539264	12271	5068558	23301	2420
Xeon-Gold-6150	9	3868.49	16.07	770%	1554596	12303	4628916	24109	2549
Xeon-Gold-6150	10	4946.96	17.26	872%	1566236	12318	5518057	24617	3220
Xeon-Gold-6338	1	473.02	1.95	90%	1567860	11939	563350	22455	1489
Xeon-Gold-6338	2	862.9	8.77	175%	1550572	12048	4324824	27271	1707
Xeon-Gold-6338	3	1222.37	12.03	259%	1580128	12082	4643879	27610	2371
Xeon-Gold-6338	4	1584.58	12.14	344%	1573440	11511	4548655	28427	144312

Xeon-Gold-6338	5	1907.19	12.9	427%	1569004	11695	4431074	30049	2752
Xeon-Gold-6338	6	2330.04	19	506%	1565164	12172	9687015	58502	4566
Xeon-Gold-6338	7	2913.69	16.26	603%	1560788	12107	4719795	37602	6599
Xeon-Gold-6338	8	3018.26	15.59	680%	1562044	12246	5232507	28104	5364
Xeon-Gold-6338	9	4547.57	23.66	788%	1537600	12165	8812648	56527	360285
Xeon-Gold-6338	10	3742.97	15.21	845%	1556888	12172	4876146	29907	2501
Xeon-Gold-5220R	1	676.43	3.27	92%	1555408	11862	471736	22510	3743
Xeon-Gold-5220R	2	1233.69	13.28	181%	1565228	11413	5274264	22824	5442
Xeon-Gold-5220R	3	1723.22	15.64	266%	1576984	11432	5462244	23291	36816
Xeon-Gold-5220R	4	2190.77	15.68	353%	1569084	11138	5254542	24065	137007
Xeon-Gold-5220R	5	2302	14.26	435%	1573996	12132	4816252	25702	72448
Xeon-Gold-5220R	6	2742.58	15.21	520%	1578496	12158	4821417	26362	2358
Xeon-Gold-5220R	7	3181.19	16.22	605%	1591888	11823	5036724	27059	2538
Xeon-Gold-5220R	8	3675.27	17.13	691%	1568332	12381	5370130	28047	15314
Xeon-Gold-5220R	9	4057.17	16.67	838%	1538424	1748	5039160	15217	2817
Xeon-Gold-5220R	10	4685.09	18.99	847%	1583676	12287	5358444	29572	6640
Xeon-Platinum-8260	1	613.29	3.6	92%	1545280	11932	650196	22487	1942
Xeon-Platinum-8260	2	1170.35	10.68	180%	1563936	12112	5173150	25764	2255
Xeon-Platinum-8260	3	1532.89	13.58	265%	1549952	12109	4896877	24171	86115
Xeon-Platinum-8260	4	2010.36	14.38	351%	1562172	11748	4969788	24817	85497
Xeon-Platinum-8260	5	2379.58	14.43	436%	1556948	11943	4953260	27253	2565
Xeon-Platinum-8260	6	3425.36	28.06	529%	1581928	12232	11421764	72829	314339
Xeon-Platinum-8260	7	3281.19	15.31	610%	1564248	11682	4345052	26747	2417
Xeon-Platinum-8260	8	3795.63	17.04	695%	1564808	12299	5310165	28585	2466
Xeon-Platinum-8260	9	4204.45	17.01	780%	1562100	11895	5448508	29692	2907
Xeon-Platinum-8260	10	4687.74	17.68	867%	1564092	12329	5616871	30526	3825
Xeon-Gold-6150	1	596.34	2.88	93%	1565492	11975	483252	21646	1795
Xeon-Gold-6150	2	1154.31	13.13	183%	1590396	11500	5189320	23872	2322
Xeon-Gold-6150	3	1382.65	14.27	269%	1578668	11845	4996535	23849	1841
Xeon-Gold-6150	4	1925.44	14.83	358%	1570484	12097	4900707	24678	45521
Xeon-Gold-6150	5	2209.19	15.38	475%	1552964	1468	4926970	12473	59520
Xeon-Gold-6150	6	2652.47	15.37	558%	1551048	4041	4915757	15611	2217
Xeon-Gold-6150	7	3680.47	16.76	617%	1582740	12491	5533094	24509	37871
Xeon-Gold-6150	8	3455.6	16.4	684%	1557012	12446	5060210	24987	2477
Xeon-Gold-6150	9	3879.88	17.02	769%	1579980	12449	5328781	25643	2888
Xeon-Gold-6150	10	4386.49	16.55	850%	1568476	12458	5120739	26298	2991
Xeon-Gold-6338	1	478.78	1.8	91%	1564552	11763	436301	20868	1450
Xeon-Gold-6338	2	877.47	9.12	179%	1564888	11881	4463734	21848	1590
Xeon-Gold-6338	3	1227.15	11.52	283%	1574068	2482	4408353	12622	1887
Xeon-Gold-6338	4	1716.78	13.15	374%	1567628	11915	4628753	24784	93117

Xeon-Gold-6338	5	2096.38	13.95	467%	1577368	11549	4665179	25677	2049
Xeon-Gold-6338	6	2361.35	13.64	549%	1577132	12706	4383022	26198	2435
Xeon-Gold-6338	7	2733.7	19.91	637%	1592956	12323	11394192	51966	343109
Xeon-Gold-6338	8	2975.95	14.48	738%	1541764	11797	4946240	28349	2561
Xeon-Gold-6338	9	3580.71	22.15	828%	1563640	12326	8903218	137520	529121
Xeon-Gold-6338	10	3826.94	15.06	925%	1567792	11839	4975812	28713	60365
Xeon-Gold-5220R	1	670.63	4.19	97%	1567104	11880	607232	21015	4787
Xeon-Gold-5220R	2	1229.89	14.77	190%	1565940	11517	5307306	21379	5915
Xeon-Gold-5220R	3	1468.59	13.25	283%	1582388	12140	4972647	23704	2195
Xeon-Gold-5220R	4	1886.11	13.97	375%	1576724	11785	4894380	24440	2146
Xeon-Gold-5220R	5	2308.7	14.35	467%	1586660	12023	4879360	25697	2603
Xeon-Gold-5220R	6	2730.13	14.99	559%	1565568	11668	4916412	25940	2297
Xeon-Gold-5220R	7	3605.78	18.1	646%	1594076	12190	5135027	26177	3725
Xeon-Gold-5220R	8	3572.82	16.92	703%	1575380	12261	4872022	23946	2891
Xeon-Gold-5220R	9	4194.46	18.09	778%	1562456	12307	5252495	28649	6435
Xeon-Gold-5220R	10	4507.95	17.25	866%	1541112	12300	5047053	29874	168334
Xeon-Platinum-8260	1	582.43	2.25	97%	1546608	11937	425173	22598	12211
Xeon-Platinum-8260	2	1177.99	12.42	190%	1573924	12010	5188465	22422	2486
Xeon-Platinum-8260	3	1537.91	13.78	282%	1571496	12243	5145545	23030	2348
Xeon-Platinum-8260	4	2015.75	14.37	375%	1583596	11845	5216986	24210	304715
Xeon-Platinum-8260	5	2474.31	14.62	468%	1592604	11518	5207630	24451	2381
Xeon-Platinum-8260	6	3522.06	27.84	560%	1575904	10969	10960882	76258	429929
Xeon-Platinum-8260	7	3249.51	15.84	653%	1565732	12225	5268840	27015	4525
Xeon-Platinum-8260	8	3713.87	16.46	745%	1572248	12381	5457407	26998	2508
Xeon-Platinum-8260	9	4145.76	17.06	836%	1575388	12153	5267256	28342	4875
Xeon-Platinum-8260	10	4713.1	17.26	931%	1540892	12197	5478202	29408	2795

Appendix 2: Collected results (n=200)