

Peter keres

Dec 7, 2018

Data Structures 2108

Final Project

“Technical Report”

This technical report is for Columbus Software Solutions (CSS) on which company's sorting library better suite our needs. The two companies in review are ACME Corporation and AJAX Inc. They both offer Software as a Service (SaaS) that can be used to sort our daily records. The report will detail how both companies' algorithms work for sorting records and also look at the price structure of each SaaS. By the end of the report, there will be a clear decision on which SaaS to purchase.

What is SaaS

Software as a Service generally means it is software that allows data to be accessed from a device over the internet. There are many examples and different types of services that fall under SaaS. G-mail is an example of such service. Because we access the data that google stores, your e-mails, on a device, your computer or smart phone, over an internet connection. Not all SaaS's are free, for example Netflix is one SaaS service that must be paid for in order to access the data that it provides. In the case of CSS, we are going to use either one of the SaaS provided by ACME or AJAX for the purpose of sorting our daily record counts.

Why SaaS

Generally speaking, SaaS are used by smaller companies and individual. Because it is more cost effective to pay a company to use their SaaS than implement one of your own. For example if we look at g-mail, If a small company wanted to start their own g-mail like service. They would have to, install e-mail servers, keep the servers up and running, and also pay an IT to keep the servers in check. This gets very expensive specially if the amount of data you have to process is low and in that case it is better off to pay another company to do this service who has already sunk the cost into the upkeep of the service. In the case of CSS, it is much cheaper for us to pay another company to run the algorithm that sorts our records instead of us paying the salary of software engineers and housing the data ourselves.

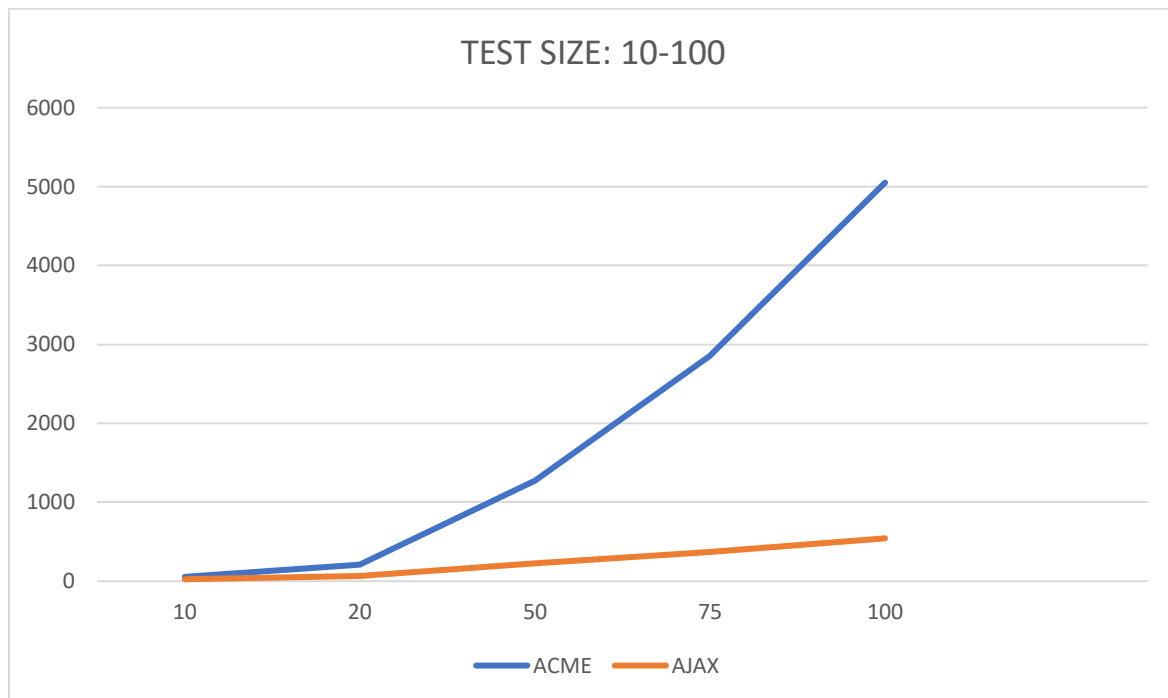
Problems with SaaS

One of the biggest problems with SaaS is its cost at a grand scale. So, as it was mentioned above, once the amount of data you need to access gets above a certain amount, it is generally more cost effective for the company to implement their own service internally. Another problem with using SaaS is that if that company was to shut down, the service they provided to you is now longer

accessible. Now you must wait until the service is back up and running or transfer your data to a new SaaS provider. Finally, it is possible that you will not own the data that is send off to the SaaS. This depends on the terms you and the SaaS agree on when starting the service.

Algorithms efficiency

In this section, we will be going over how efficient each company's algorithm are in terms on number of instructions the CPU has to perform. First we look at how each hand a small amount of records. The sample size will be 10 – 100 records. The number on the left is the amount of instructions, while the number at the bottom is the number of records for each test.



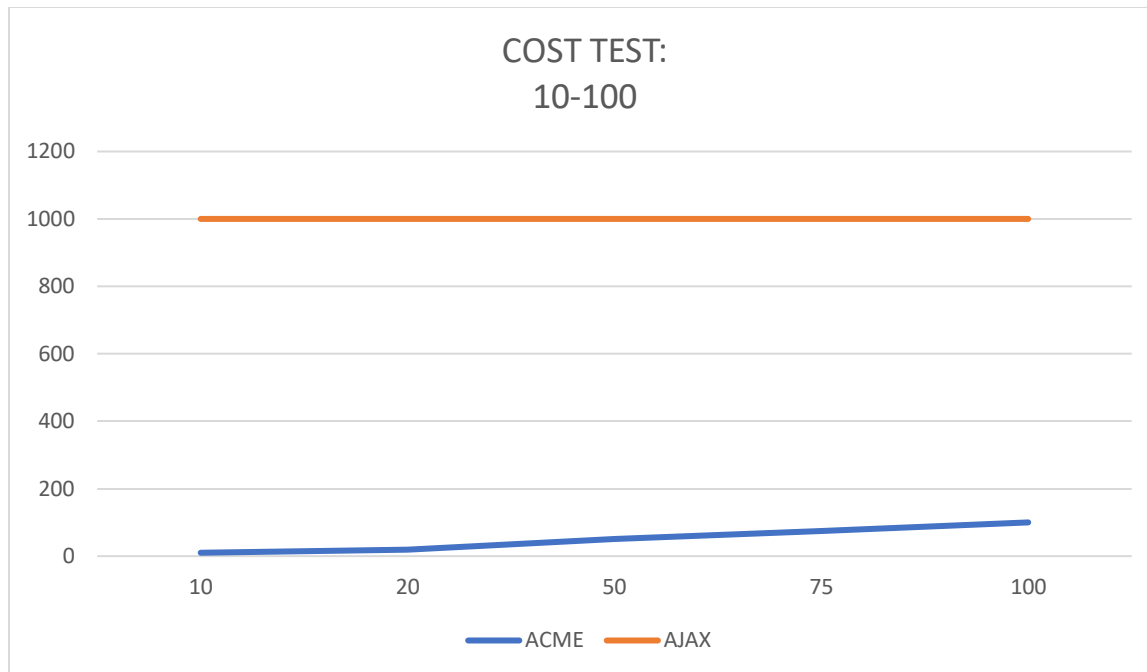
The graph shows that if the amount of records is under 100, then they both act about the same in terms of efficiency. Once the amount of recorders goes over 100, the gap between the two algorithm starts to grow immensely. The second graph will show a sample size with much larger records to be processed.



As you can see, once the data gets very large it simply just cannot handle that amount of data in an efficient manner. Also noticed during the testing process, is that the time in Nano seconds also mimics this trend between the two algorithms. So not only does the AJAX algorithm perform the sorting of records faster in terms on CPU instructions, but also in terms on real world time.

SaaS cost

From the prospective of CSS, How each company's algorithm handle is not our concern. But how much it will cost us for them to process our data. To figure out which one was better for CSS, I implemented a test that would see how much it would cost based on number of records. The first graph will show a test size of 10-100 records. The left side will show the amount of money it will cost and the bottom shows the amount of records used.



As you can see, with a small amount of records ACME is much better in terms of cost. If you notice in the graph that the cost of ACME is starting to rise, that's because its getting out of the lower tier cost category. The second graph will show how much it will cost once the amount of records starts to get larger.



Once the data gets very large, AJAX is much more cost efficient. Because CSS is in the million of records, it would be much cheaper to go with AJAX as the SaaS. The following graph will show the amount of savings by using AJAX as a SaaS based on the daily averages

	MONDAY	TUESDAY	WENSDAY	THURSDAY	FIRDAY	SATURDAY	SUNDAY
ACME	115,098,112\$	1,300,000,000\$	345,294,336\$	575,490,560\$	650,000,000\$	1,150,981,120\$	130,000,000\$
AJAX	10,000,000\$	1,000,000\$	30,000,000\$	50,000,000\$	500,000\$	100,000,000\$	100,000\$
SAVINGS	105,098,112\$	1,299,000,000\$	315,294,336\$	525,490,560\$	649,500,000\$	50,981,120\$	129,900,000\$

Conclusion

If the amount of records that CSS was under 100 daily, I would recommend to go with ACME. Because from a business prospective, we don't care about how efficient the algorithm runs but how much it would cost us. Considering the fact that CSS normal daily records is over 1 million, ultimately I suggest AJAX. Not only is their algorithm more efficient but it also costs much less.