DAT470/DIT065 Assignment 4

Mirco Ghadri
mircog@chalmers.se

May 5, 2024

This assignment will focus on using Spark in Python (PySpark) to solve problems related to processing large volumes of data. Part 1 will use PySpark on the same twitter dataset from assignment 3[1] to solve the same questions related to followers/followings. Part 2 will focus on using PySpark to analyze large volumes of OpenSSH logs[2] to extract meaningful aggregate statistics.

Problem 1: Twitter followers (12 pts)

(a) Implement the missing bits of `pyspark_twitter_follows.py` *to determine the maximum number of people followed, the Twitter id of the account with the maximum number of people followed, the average number of people followed, and the number of accounts that follow no-one.*(4 points)
Answer: See `assignment4_problem1a.py`

(b) Measure the scalability of your algorithm on 1, 2, 4, ..., 32 cores. Plot the empirical speedup as the function of cores. In addition to the plot, report the single-core runtime on the dataset. (2 points)
Answer: We got the following table of running times

Table 1: Number of Workers vs. Total Running Time (New Data)

Number of Workers	**Total Running Time (seconds)**
1	196.91
2	108.12
4	67.34
8	51.49
16	41.40
32	29.14

To get the empirical speedup as a function of the number of CPU cores, we used the equation

$$S(n) = \frac{t_1}{t_n} \tag{1}$$

where t_1 is the total running time with 1 CPU core/worker and t_n is the total running time with n CPU cores/workers. We got the following table for the empirical speedup as a function of the number of CPU cores.

Table 2: Speedup Calculation

n (Number of Workers)	**Speedup (S)**
1	1
2	1.821
4	2.924
8	3.822
16	4.754
32	6.760

From this table, we could then plot the empirical speedup

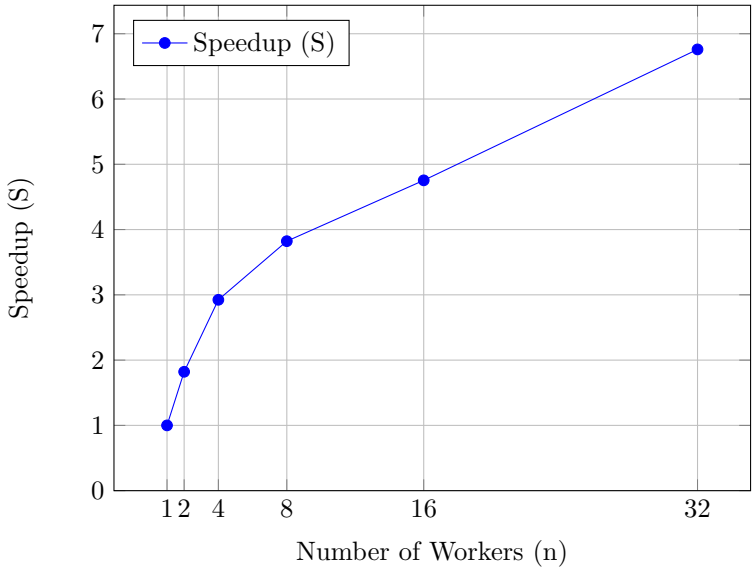

Figure 1: Speedup vs. Number of Workers

(c) Implement the missing bits of `pyspark_twitter_followers.py` *to determine the maximum number of followers, the Twitter id of the account with the maximum number of followers, the average number of followers, and the number of accounts that have no followers.* (4 points)
Answer: See `assignment4_problem1c.py`

(d) Measure the scalability of your algorithm on 1, 2, 4, ..., 32 cores. Plot the empirical speedup as the function of cores. In addition to the plot, report the single-core runtime on the dataset. (2 points)
Answer: We got the following table of total running times

Table 3: Number of Workers vs. Total Running Time

Number of Workers	**Total Running Time (seconds)**
1	475.80
2	328.12
4	164.72
8	140.40
16	90.18
32	56.45

From the table of total running times, we could derive the table of Speedup for different number of CPU cores

Table 4: Speedup Calculation

Number of Workers	**Speedup (S)**
1	1
2	1.450
4	2.890
8	3.391
16	5.276
32	8.429

From the table of speedup, we could construct a plot that shows Speedup as a function of the number of CPU cores/workers.

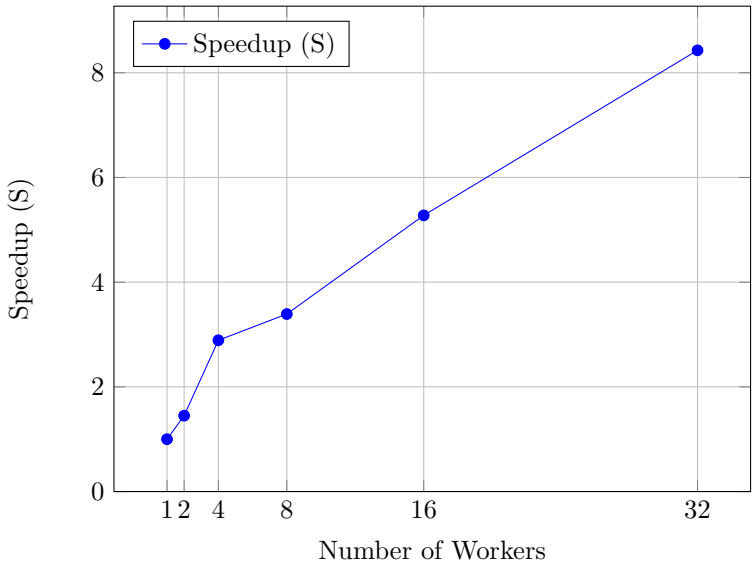

Figure 2: Speedup vs. Number of Workers

Problem 2: OpenSSH logs (12 pts)

For the python file that answers all of these questions, see `assignment4_problem2.py`.

(a) Determine which user account had the largest number of login attempts (successful or unsuccessful). Which account was it and how many attempts were there? (2 pt)
Answer: The account was **root**. It had a total of 323827 login attempts. All of the login attempts however were unsuccessful. There was not a single successful login attempt that contributed to the total count.

(b) Determine which user account had the largest number of successful login attempts. Which account was it and how many attempts were there? (1 pt)
Answer: The account **curi** had 63 successful login attempts which was the most of any user.

(c) Determine which user account had the largest number of unsuccessful login attempts. Which account was it and how many attempts were there? (1 pt)
Answer: The user root had the most unsuccessful login attempts. This was 323827 unsuccessful login attempts.

(d) Which user account had the highest success rate of logins (successful login attempts / all login attempts). What was the rate? If there were multiple accounts with the same rate, report them all. (2 pt)
Answer: The highest login success rate for an account was 100%. Multiple accounts had this rate. The accounts were : **xxchen, hxu, zachary, suyuxin, jmzhu, tzhao, fztu, curi, yuewang**. They all had a login success rate of 100%.

(e) Determine the top 3 IP addresses with the largest number of failed login attempts and report the IP addresses together with the number of login failures. (2 pt)
Answer:

Table 5: Top 3 Failed Logins by IP Address

IP Address	Failed Logins
59.63.188.30	86294
58.242.83.25	43147
218.65.30.30	25102

(f) Determine the top 10 invalid user accounts with the largest number of failed login attempts and report the user accounts together with the number of login failures. (2 pt)
Answer:

Table 6: Top 10 Invalid User Accounts and Failed Login Attempts

User Account	Failed Login Attempts
admin	8073
test	543
oracle	489
support	486
user	369
pi	352
nagios	296
guest	259
postgres	216
ubnt	214

(g) Which day had the most login activity? Which day had the least? (2 pts)
Answer: The login activity was measured in number of failed/successful login attempts for all accounts(valid/invalid).

Table 7: Maximum and Minimum Login Activity

Day	Login Activity
Jan 4	77409
Dec 23	627

References

[1] Haewoon Kwak and et al. "What is Twitter, a Social Network or a News Media?" In: *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 2010. DOI: 10.1145/1772690.1772751.

[2] Jieming Zhu and et al. "Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics". In: *Proceedings of the 34th IEEE International Symposium on Software Reliability Engineering (ISSRE 2023)*. 2023. DOI: 10.1109/ISSRE59848.2023.00071.