Designs of Algorithms and Programming for Massive Data

DS8001

Andriy Miranskyy

Nov. 7, 2016

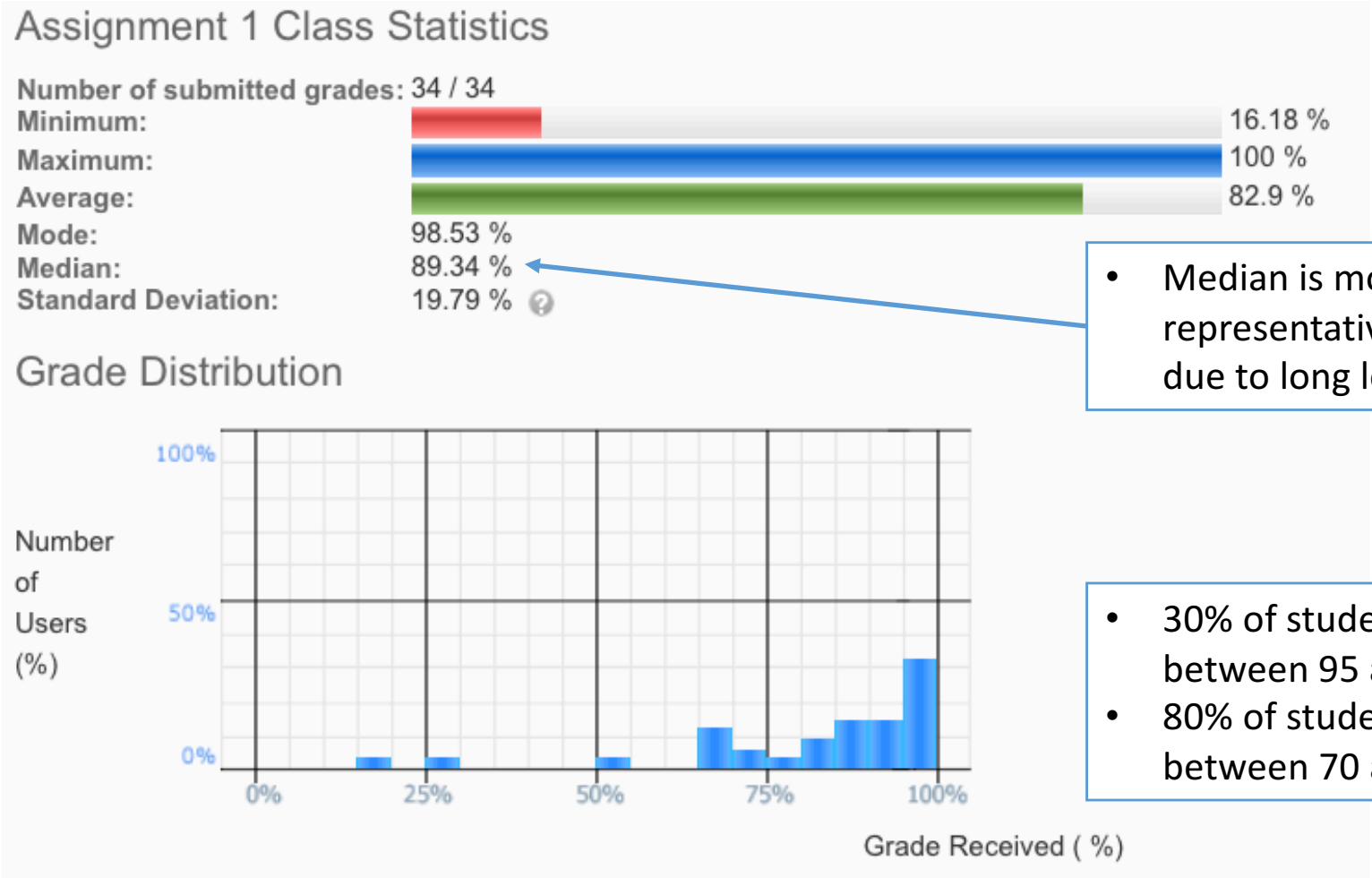
Outline

- Misc.
- Parallelization

Course

Miscellanea

Assignment 1 Stats



- Median is more representative than the mean due to long left tail

- 30% of students got a mark between 95 and 100%
- 80% of students got a mark between 70 and 100%

Examples of plagiarism

- *Copying and pasting material from a website*
- *Making minor changes to an author's words or style and then presenting the material as your own*
- *Using a direct quotation but leaving out the quotation marks*

- See [1, Section 2.1] for other cases
- Check [2] to understand how to avoid

1 Taken from http://www.ryerson.ca/senate/policies/pol60.pdf

2. http://www.ryerson.ca/academicintegrity/students/what-is-integrity-and-misconduct/#quick-hints

Consequences*

- The **minimum** penalty you will receive is a mark of zero on the test, exam, paper, project or assignment in question
- The “Disciplinary Notice (DN)” will be placed on your academic record where it will remain until you graduate.
- The professor might also decide to fail you in the course.
- If you already have a DN on your record you will be placed on “Disciplinary Withdrawal (DW)”.
- The University also has the right to expel you from the University.

* Taken from http://www.ryerson.ca/academicintegrity/students/penalties-and-consequences/#graduate-student-policy

Midterm marking

- In progress, should be done by next week

Project

- Have you converged on a topic for your project?
- Share with me please

IBM: Data Science Hub

IBM Data Science Hub

- http://datascience.ibm.com
  - When you login for the 1st time – agree to create Spark instance right away (you don't want to do it manually ☺)
- To get 6 month trial: https://ryerson.onthehub.com/WebStore/OfferingDetails.aspx?o=809b931d-4060-e611-9420-b8ca3a5db7a1

- Developers would love to hear your feedback!

Parallelization

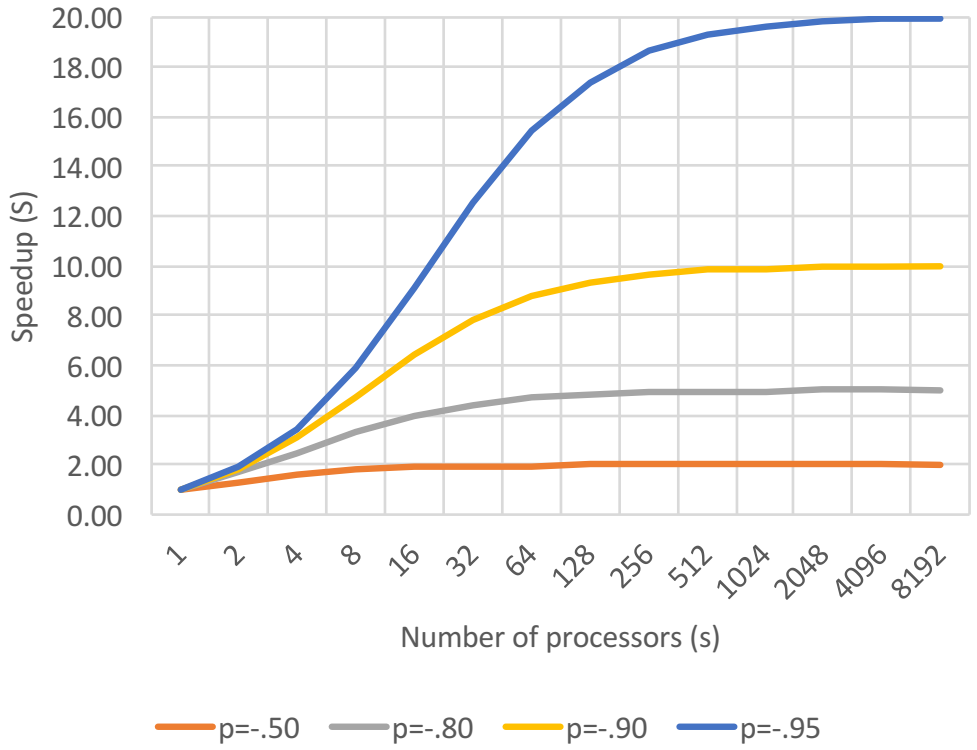
Amdahl's Law

- Variables:
  - Overall execution time: T
  - Fraction of the time (before improvements) that can be improved (parallelizable): p
  - Speedup in latency of execution: s
- Before improvement / parallelization:
  - $T = (1 - p)T + pT$
    - pT – the part than can be improved / parallelized
- After improvement / parallelization:
  - $T(s) = (1 - p)T + pT/s$
  - pT/s – time after improvement / parallelization is applied

Amdahl's Law

- Speedup, given workload W:
  - $S_{latency}(s) = [\ T\ W\ ] / [\ T(s)\ W\ ] = [(1 - p)T + pT] / [(1-p)T + pT/s] = 1 / [\ 1 - p + p/s]$

Why the speedup is not linear?



Amdahl's Law

- Think of a car travelling between two cities 100 km apart
- If we traveled for 1 hour at 50 km/h, we covered 50 km
- Given the current point in space and time, if instantaneously teleport to our destination, our overall speed cannot exceed 100 km/h

- Informal corollary:
  - If the non-parallelizable code consumes 50% of the time, at best I can achieve a speedup of 2x.

Amdahl's Law

- Note that that the law ignores
  - Load balancing
    - What if 9 jobs finish in 1 hour, but the 10^{th} job take 1 day?
  - Overhead
    - Starting parallel jobs takes time…
    - Passing data between jobs takes time...
    - etc.

Karp-Flatt metric

- Measure the actual speedup ψ and the number of processors *n*
- Then the measure of parallelization *e* is defined as

$$e = \frac{\frac{1}{\psi} - \frac{1}{n}}{1 - \frac{1}{n}}$$

- *e* ranges between 0 and 1
- The closer *e* to 0 – the more efficient is the implementation
  - The best case is if $\psi = n$

Parallel algorithms

- Embarrassingly parallel problems
  - Easy to divide
  - Example: extraction of keywords from web-pages
- Inherently serial problems
  - Cannot divide
    - Typically because the next step depends on the previous step
  - Example: finding roots of a real-valued function using Newton's method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Parallel Algorithms Design

Suggested reading

- A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, 2nd ed., Addison Wesley, 2003
  - Chapters 3, 5