Scaleable Computing

Status and practical#5

CS7NS1/CS4400
Stephen Farrell
stephen.farrell@cs.tcd.ie

https://github.com/sftcd/cs7ns1/
Note: PRs for repo are welcome!

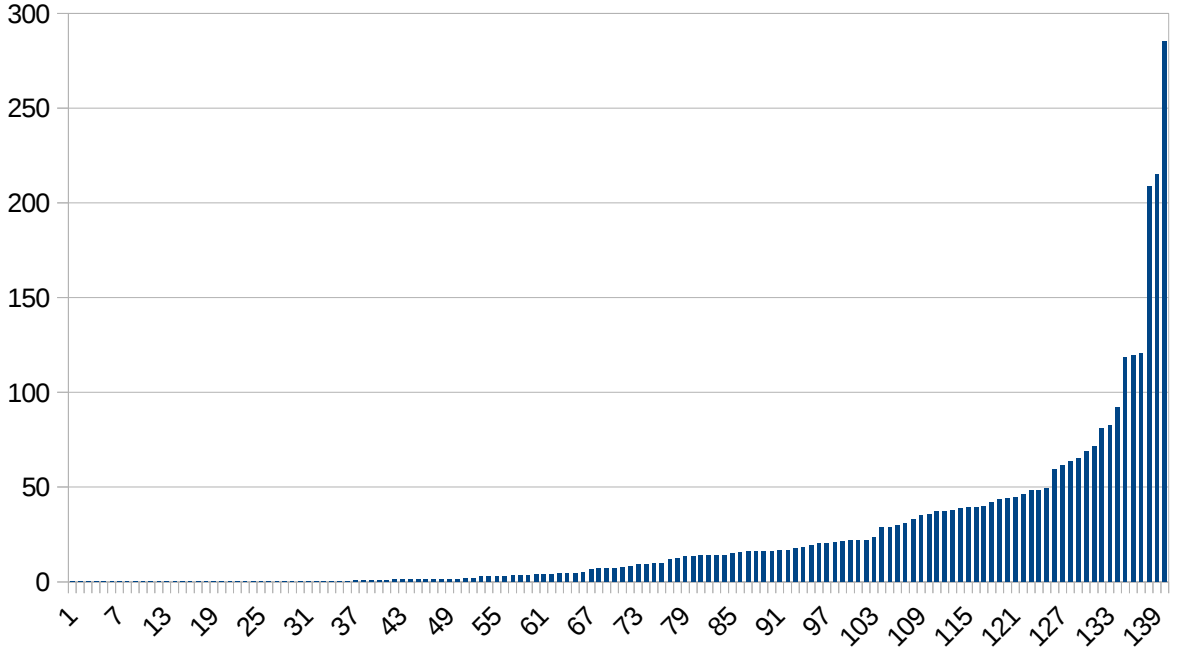
Contents

- Practical#4 status
- Practical#5 intro - “InfernoBalls”

Course Numbers (20181014)

- TCD student DB (SITS): 150 students
  - CS7NS1: 95 Students
  - CS4400: 55 Students
- No longer doing module, replied to my 20180925 mail: 12
- Rosetta Hub:
  - Registrations uploaded: 175 bulk + 1 ad-hoc
  - Accounts created: 140
  - Registrations never touched: 36
  - AWS Budget >= $100: 127
  - AWS Budget issues all resolved?

AWS Spend (20181014)



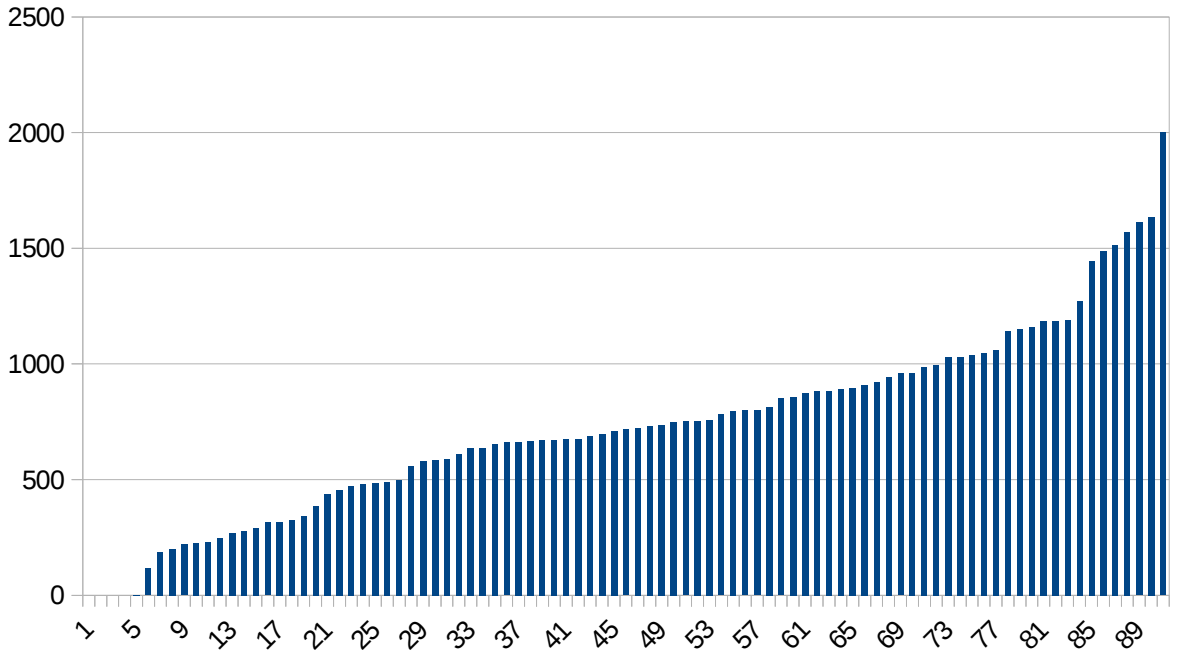
- Students at different spend levels:
  >$200: 3; $100-$200: 3; $60-$100: 8; $40-$60: 9; $20-$40: 22; $0-$20: 95

Submitty Numbers (20181014)

- Submitty:
  - “Live” students: 161
- Assignment#2: 131 all 15/15
- Practical#3: 131 submitted
  - Average: 736.25/1000, Std dev: 427.72 !!!
  - Lots of 1000/1000; some 99x, likely due to i18n; some outliers who will score less well

Submitty Numbers (20181015)

- Practical#4: 90 submitted @ 20181015T090900Z
- Average: 377.82/2000
- Std dev: 400.59
- Best: 1633/2000
- 4 with 0 score!
  - Fix formatting
  - Or file PR/bug



Practical#4? What'd you do so far?

- Done
  - Hascat & jtr work; GPUs@ home work
  - Masking works
  - Rockyou, 5 lower case random, 4+4 english words
  - DEScrypt collisions
  - Rules: dive
- Not done, or not much done:
  - Combine recovered password lists (why not for the entire class?)
  - Orchestration: script things up!

Practical#4: Next Steps...

- Deadline extension to Friday Oct 19th
- Because:
  - I want you to collaborate in the meantime
  - Pay attention to the actual data with which you are (collectively, scalably:-) computing
  - You'll be happier with practical#5 if you do that!

Practical#5 Logistics

- Open for submissions from Monday Oct 29th
- Due-date: Monday November 19th
- Teams of 4
- Initial teams randomly selected from those who scored >0 on some submitty practical already
  - If you haven't scored >0 on a submitty practical, please see me @ end of class or by email, (and have your excuses primed;-)
- Given the extension to practical#4 I may (or may not) choose to make the team selection randomly verifiable [RFC 3797]
  - You'll hear more via email if I do
- Any team member can submit to submitty
- Last submitted version scores for the assignment

Practical#5: InfernoBalls

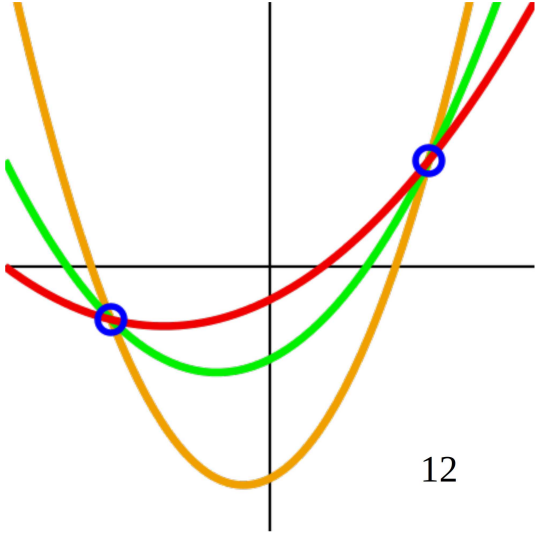
- Dante’s Inferno
  - https://en.wikipedia.org/wiki/Dantes_Inferno
  - 9 (+1) circles of hell from 800 years ago
- Dante (the pilgrim) and Virgil descended from the top to the bottom
  - Then went on up to Purgatory/Paradise
    - That last isn’t part of this module:-)
- We’re gonna do a digital analog

InfernoBall structure

- An InfernoBall is a JSON structure with these fields:
  - Hashes: a set of password hashes
  - Shares: a variant on a set of Shamir-share values
  - Ciphertext: an encrypted version of the next level of hell
- Once you recover enough passwords from the hashes, then using those, and the shares values, you can recover the secret to decrypt the ciphertext and enter the next level of hell
  - Rinse and repeat, ‘till done :-)

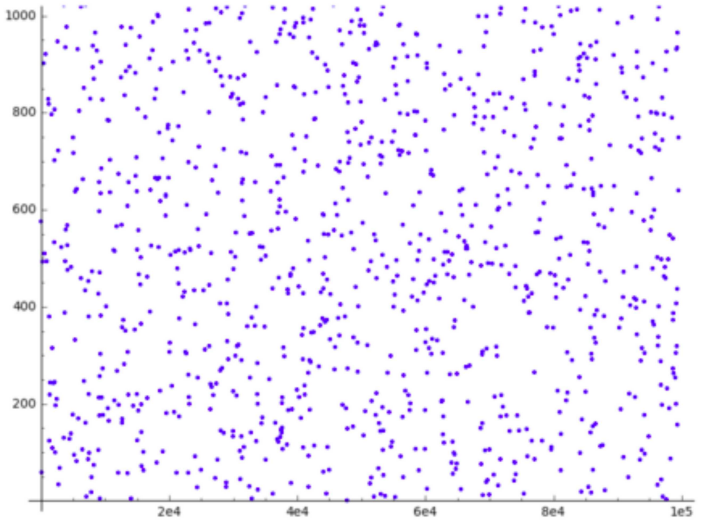
Shamir Secret Sharing

- Given a secret, how can I give n people a value such that once at least k (<=n) people get together they can reconstruct the secret?
- Shamir's scheme: If I have a polynomial of degree k-1, and I give each person one point (e.g. "i,f(i)" for i=1..n) then if any k of them get together they can find the formula for the polynomial and hence the secret
  - Usually the secret is "f(0)"
- Intuition: I need two points to know the formula for a line. I need three points to find the formula for a quadratic, etc.
  - https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing
  - Figures on this and next slide from that Wikipedia page



More on Shamir

- For security reasons we choose polynomials over a finite field e.g. F(p) with p a prime that's big enough for the given k-of-n scenario
  - With a polynomial over the reals, k-1 points can allow guessing of the k-th point a bit too easily, e.g. if secret value were "small" guessing may be cryptographically efficient
- Intuition: polynomial curves over finite fields are more "random" looking
- We end up dealing with polynomical coefficients that are maybe 64 or 128 bit values
  - Hey, it's crypto:-)
- I'll give you reference code for this (Python 2.7)
  - That code is part of the spec for this practical
    - Not all Shamir packages interop!



Shamir isn’t quite what we want

- Shamir scheme starts with a secret. We want to start from a set of recovered passwords (i.e. things like shares) - a bit of thought will show that we can’t immediately do that.
- But...
  - Generate a secret ‘s’
  - Generate a set of n shares s_i for s using a standard Shamir k-of-n setup
  - Select n passwords, p_i
  - Distribute E(s,plaintext), { SHA256(p_i) XOR s_i } for i=1..n
- Recovery of s: given k passwords we can find k Shamir shares recover s and then decrypt E(s,plaintext) to get plaintext
- I’ll give you Python 2.7 code to construct (not decrypt!) such an InfernoBall
- Note: I nearly buggered up the implementation of this due to lengths!!
- **This is not claimed to be “secure” (but will be fine for the practial)**

So InfernoBalls then...

- Shares are Shamir-variants as described above
- Each level of hell has a different, randomly selected secret and k-of-n setup
  - You can see n by observation
  - You are not given k but have to find that
- There are 10 levels of hell (9+1, same as for Dante:-)
- Hashes use algorithms previously seen
- Password kinds include those previously used
  - There are some new ones... go figure!
- Some passwords are part of >1 InfernoBall
- No hash value is included in >1 InfernoBall

More InfernoBall detail

- You will be given:
  - A sample InfernoBall and the secrets involved
  - Python 2.7 code to create an InfernoBall
  - Your own team InfernoBall
- Your team tasks include:
  - Orchestrating CPU/GPU and AWS budget issues
  - Wordlist handling
  - Coding to unwrap InfernoBall
  - (Possible: crypto-break against the InfernoBall implementation?)
- Submission format is a file with one secret per InfernoBall level (so 10 lines total)
- Bonus marks for being first to reach a level

InfernoBall Collaboration

- It is entirely ok to compare data found and techniques tried
- It is not ok to share coding/debugging, nor to run cracking jobs for other teams
- The bonus marks for first to solve a level are an attempt to encourage that

InfernoBall Logistics

- Open for submissions from Monday Oct 29th
- Due-date: Monday November 19th
- Teams of 4
- Initial teams randomly selected from those who scored >0 on some submitty practical already
  - If you haven't scored >0 on a submitty practical, please see me @ end of class or by email, (and have your excuses primed;-)
- Given the extension to practical#4 I may (or may not) choose to make the team selection randomly verifiable [RFC 3797]
  - You'll hear more via email if I do
- Any team member can submit to submitty
- Last submitted version scores for the assignment

Questions? Things to add?

- I’ll come back on Thursday and go over the InfernoBall stuff again
  - Unless you’re all just fine with it as is?
- <your text here>