http://socialarchitecture.science

Gareth Hayes

Director of Technology at Emurgo HK

Ask me after if you want a job.

Github: gazhayes
gareth.hayes@gmail.com

EMURGO

CARDANO

http://socialarchitecture.science

The Trend Towards Decentralisation

轻松筹®

滴滴

滴滴一下 美好出行

KICKSTARTER

agile

Scrum

UBER

git

http://socialarchitecture.science

Where is Decentralisation Taking Us?

- Software will be grown and evolve the same way life evolves
- Developers will still be critical but the way we work will be different
- Roadmaps, planning, meetings, and project managers will all disappear
- Companies that require upfront consensus (planning, meetings, roadmaps, etc) will stop being profitable
- There are underlying principles driving this change. They are forces of nature and can’t be stopped.
- These forces are predictable.

Cost Gravity

- Cool stuff gets cheaper
- Moore’s Law is a subset of cost gravity
- Technology decreases in price at an exponential rate
- Human brains can’t understand exponential changes, we see it as a sudden change

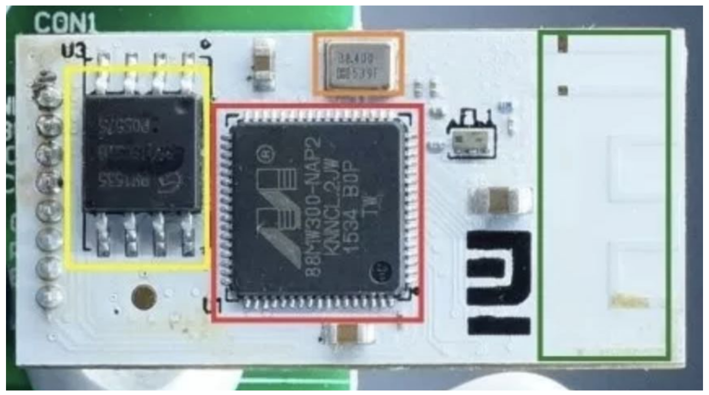
This light bulb has 100x more computing power and 300,000x more bandwidth than the lunar module that put Neil Armstrong on the moon in 1969

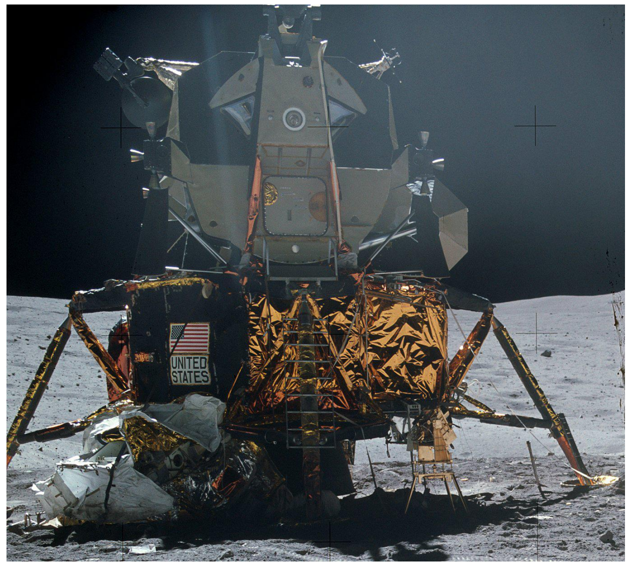
Arm Cortex-M4F 200MHz

16MB SPI Flash

512kB SRAM

450Mbps wifi





2MHz RTL based IC
74kB ROM
4kB RAM
1.6kbps data link

This is what cost gravity does to our stuff.

http://socialarchitecture.science

Cost Gravity

- In 50 years, the average person on the planet will have more computing power and more connectivity than the entire internet today.
- Cost gravity goes back to the stone age.
- The number of silicon chips and the number of connections between them double every ~2 years.
- This is why we have the internet of things (IoT).
- Our other systems are doing the same thing.
- The size of our systems and the number of moving parts in our systems doubles every ~2 years

Cost Gravity

- The number of moving parts doubles every ~2 years, which means the complexity also doubles every ~2 years
- The scale also doubles every ~2 years, so this complexity is flattened out or “distributed”.
- This forces things to become less centralised. The level of decentralisation doubles every ~2 years.
- Cost gravity is creating incredible complexity distributed across a massive scale

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

Wisdom of the Crowd

How Many?

http://socialarchitecture.science

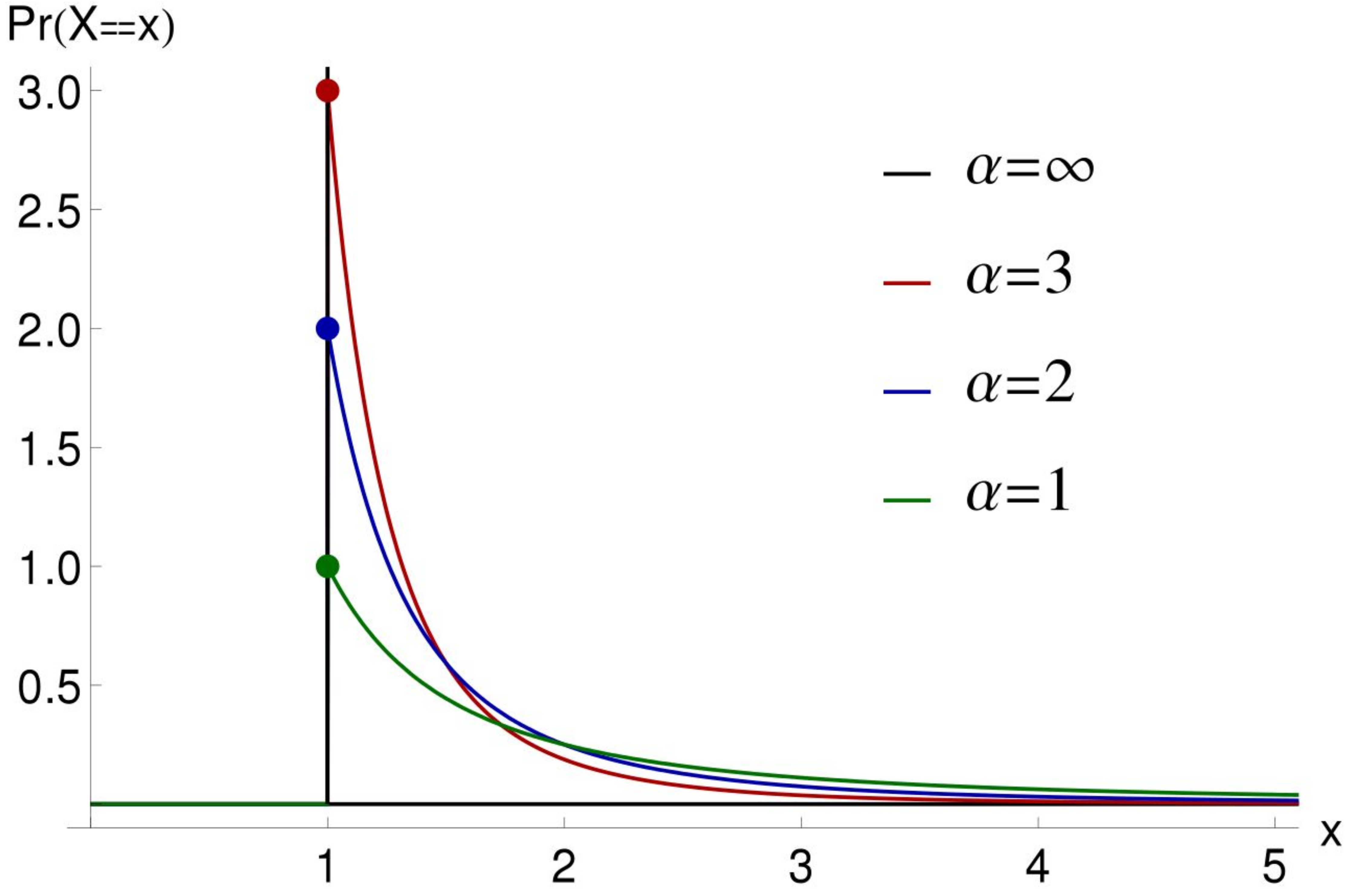
Wisdom of the Crowd

- The more guesses there are, the more accurate the average guess is
- The larger and more diverse or random a group of people is the more accurate they become - diversity of **thought** is a requirement for collective intelligence
- Adding experts to the group makes the group less accurate
- Wikipedia uses this phenomena, the more people involved in editing an article, the more accurate the article becomes.

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

Pareto Distribution



http://socialarchitecture.science

Pareto Distribution

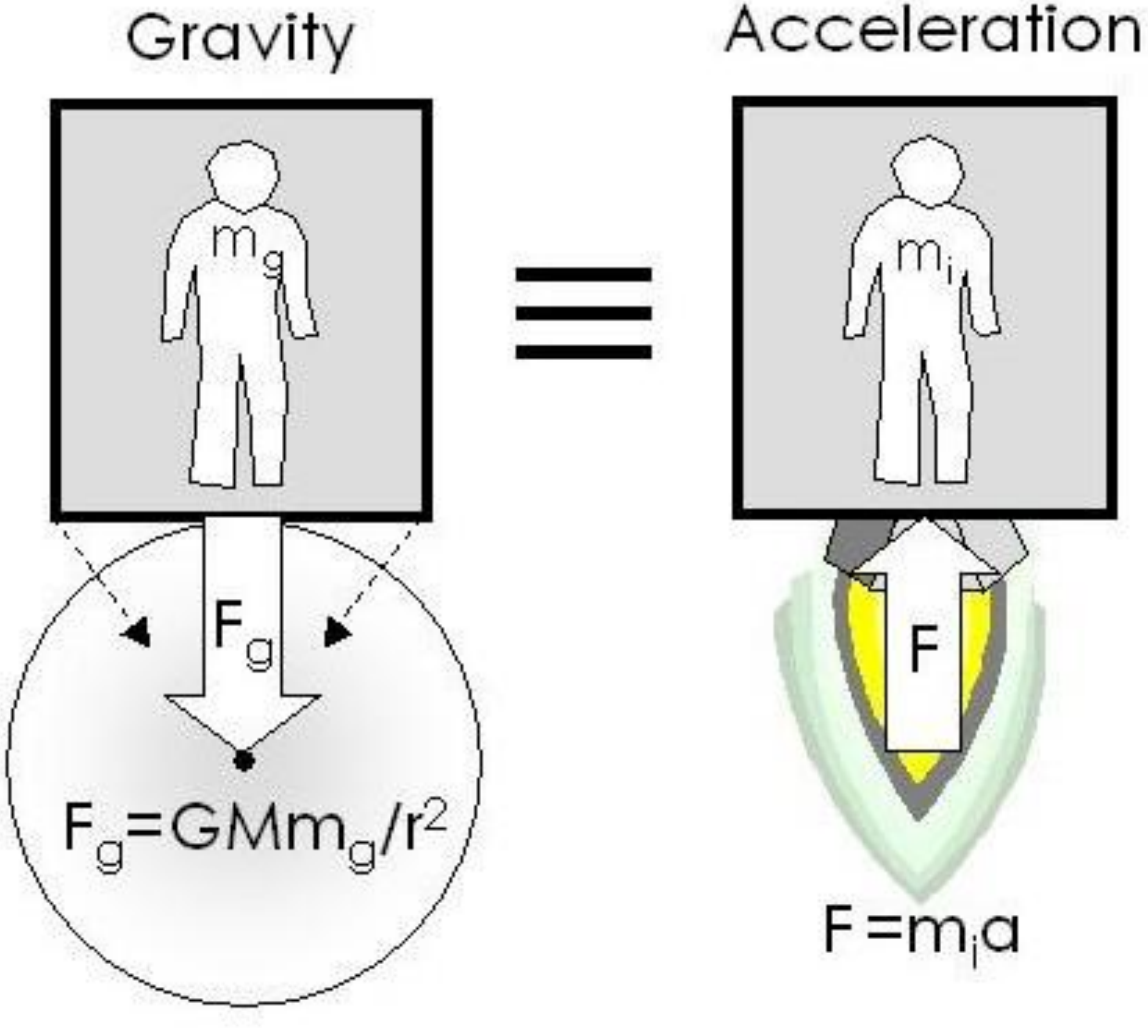
- Similar to 80/20 rule
- 80% of developer time is spent on 20% of the code
- There’s nothing you can do to avoid it
- Before you write your code, you don’t know which code is the 20% that will eat 80% of your time
- You should never work on code that isn’t critical
- Shitty code that solves problems is not shitty code
- Great code that doesn’t solve a problem is a waste of time

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto principle tells us code that doesn’t solve a problem is a waste of time

Equivalency principle



Equivalency principle

- Accelerating forward feels the same as accelerating towards the ground
- Producing beautiful code != moving in correct direction
- What is the correct direction? How can you know?
- Code that doesn’t solve a problem is worthless, so that’s the wrong direction
- Code that solves a problem is the right direction
- Code that doesn’t solve a problem is technical debt

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto and equivalency principle tells us code that doesn't solve a problem is a waste of time **and** it adds technical debt

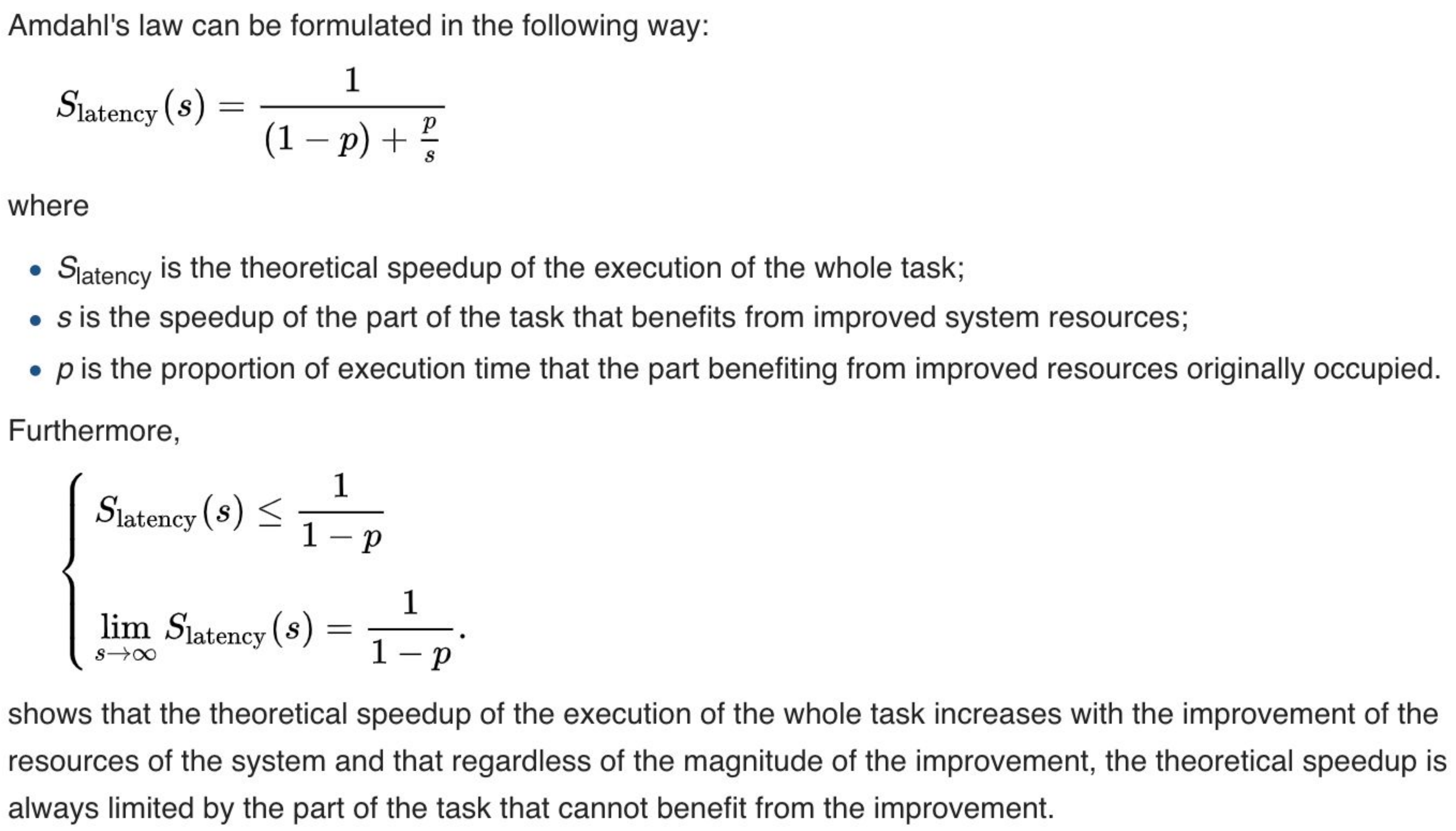
1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto and equivalency principle tells us code that doesn't solve a problem is a waste of time **and** it adds technical debt

Amdahl’s Law

- Used to calculate speed increase by adding more parallel cores/workers/threads to a system
- If your code is running on multiple cores but it needs to spend 25% of the time in some kind of mutex situation, where everything has to stop so that it can synchronise data, then your maximum number of cores is 4.
- Adding more cores after this does **not** increase speed
- Mutex situations or synchronisation points are points where ***upfront consensus*** (upfront agreement) is needed before a process can continue working.



Amdahl's law can be formulated in the following way:

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

where

- S_{latency} is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

Furthermore,

$$\begin{cases} S_{\text{latency}}(s) \leq \dfrac{1}{1-p} \\[2ex] \lim\limits_{s\to\infty} S_{\text{latency}}(s) = \dfrac{1}{1-p}. \end{cases}$$

shows that the theoretical speedup of the execution of the whole task increases with the improvement of the resources of the system and that regardless of the magnitude of the improvement, the theoretical speedup is always limited by the part of the task that cannot benefit from the improvement.

https://en.wikipedia.org/wiki/Amdahl%27s_law

Amdahl's Law

- The more you need consensus (agreement between people), the less work you can do
- Meetings are a type of mutex
- 1 hour of meetings = max team size of 8
- Any blocking process or synchronisation point reduce max team size
- Do you need approval to start working? Do you need approval to merge a patch into production?
- Max team size is usually 4-6 in reality (anecdotally)

Amazon 2 Pizza Teams

- Teams working at Amazon are limited to how many people can be fed with 2 pizzas: usually 4 to 6 people.
- People think this is because smaller teams are more productive, but this is very wrong.
- Teams **do not** become less productive when they get bigger. What **actually** gets less productive when you add more people is central planning.
- The real reason for the 2 pizza rule is that Amazon hasn’t worked out how to decentralise enough, their management style has too many mutex situations.

Amdahl’s Law

- Upfront consensus (agreement) and mutex situations **do not scale**
- Scalability of teams is proportional to the amount of upfront consensus
- A massively scalable team can have 0 upfront consensus, 0 mutex situations, 0 central planning
- If you can’t have upfront consensus, you can’t have any agreement on what to work on. You can’t have roadmaps. You can’t have goals. You can’t know what you’re building before you start working. All of these things require upfront consensus/agreement.
- The less upfront consensus a structure requires, the more efficient it will be

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto and equivalency principle tells us code that doesn’t solve a problem is a waste of time **and** it adds technical debt

4. Amdahl’s law means that the less upfront consensus a structure requires, the more efficient it will be. Large scale cannot have **any** upfront consensus

Conway’s Law

- The structure of the organisation is the same as the structure of the software it makes.
- If your organisation has a monolithic structure, the software you make will have a monolithic structure.
- If you have a broken up and semi-decentralised organisation, you’ll build broken up and semi-decentralised software, like microservices.
- You can work against Conway’s Law but it’s more expensive
- Massively scalable and distributed systems require massively scalable and distributed organisational structures

Conway's Law

- Cost gravity means hardware and software are doubling in scale and doubling in the number of moving parts every 2 years. Conway's law means organisational structures have to do the same in order to stay profitable.
- Amazon example: organisational structure has become less centralised at the same time as their software (microservice architecture etc)
- Does decentralisation of organisations come first, or decentralisation of software? Neither. They happen at the same time because the structure of the organisation **is** the structure of the software.

Conway's Law

- To build software that handles massive scale and massively distributed complexity, you need an organisational structure that handles massive scale and is massively distributed.
- Cost gravity is pushing us towards decentralisation at an exponential rate. Very soon, we'll suddenly realise that everything is decentralised, the same way we suddenly realised that everyone has a smartphone.
- When this happens, the most dominant organisational structure on the planet will be whatever structure is most decentralised, because that will be the most efficient structure.

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto and equivalency principle tells us code that doesn't solve a problem is a waste of time **and** it adds technical debt

4. Amdahl's law means that the less upfront consensus a structure requires, the more efficient it will be. Large scale cannot have **any** upfront consensus

5. Conway's law tells us that the organisational structure that dominates our future world will be whatever structure is most decentralised and most scalable.

http://socialarchitecture.science

1. Cost Gravity: Our systems double in scale and number of moving parts every 2 years. Our systems are becoming more distributed as a result. This forces everything to become less centralised.

2. Wisdom of the Crowd means that the larger and more random the community is that creates a codebase, the more accurately that codebase will solve a problem.

3. The Pareto and equivalency principle tells us code that doesn’t solve a problem is a waste of time **and** it adds technical debt

4. Amdahl’s law means that the less upfront consensus a structure requires, the more efficient it will be. Large scale cannot have **any** upfront consensus

5. Conway’s law tells us that the organisational structure that dominates our future world will be whatever structure is most decentralised and most scalable.

http://socialarchitecture.science

The full protocol, how to implement it, and the latest research in improving it can be found at:

http://socialarchitecture.science

If you want to be paid to work with this protocol, send me your CV and GitHub profile.

@gazhayes

gareth.hayes@gmail.com