Higher Order Company

Welcome to the massively parallel future of computing!

by Victor Taelin & friends

Problem

Software isn't ready for parallel hardware

- Since 2006, **processors** are shipping with **increasingly more cores**
- All programming languages in use today are **single threaded** by default
- **Threaded programming** is **very expensive**, because:
- 1. concurrency errors are complex (race conditions, deadlocks, etc.)
- 2. non-deterministic behavior is very hard to debug
- 3. parallelism overhead can actually reduce performance
- The pressure to parallelize software will be huge when 100+ cores are norm

Solution

A massively parallel runtime

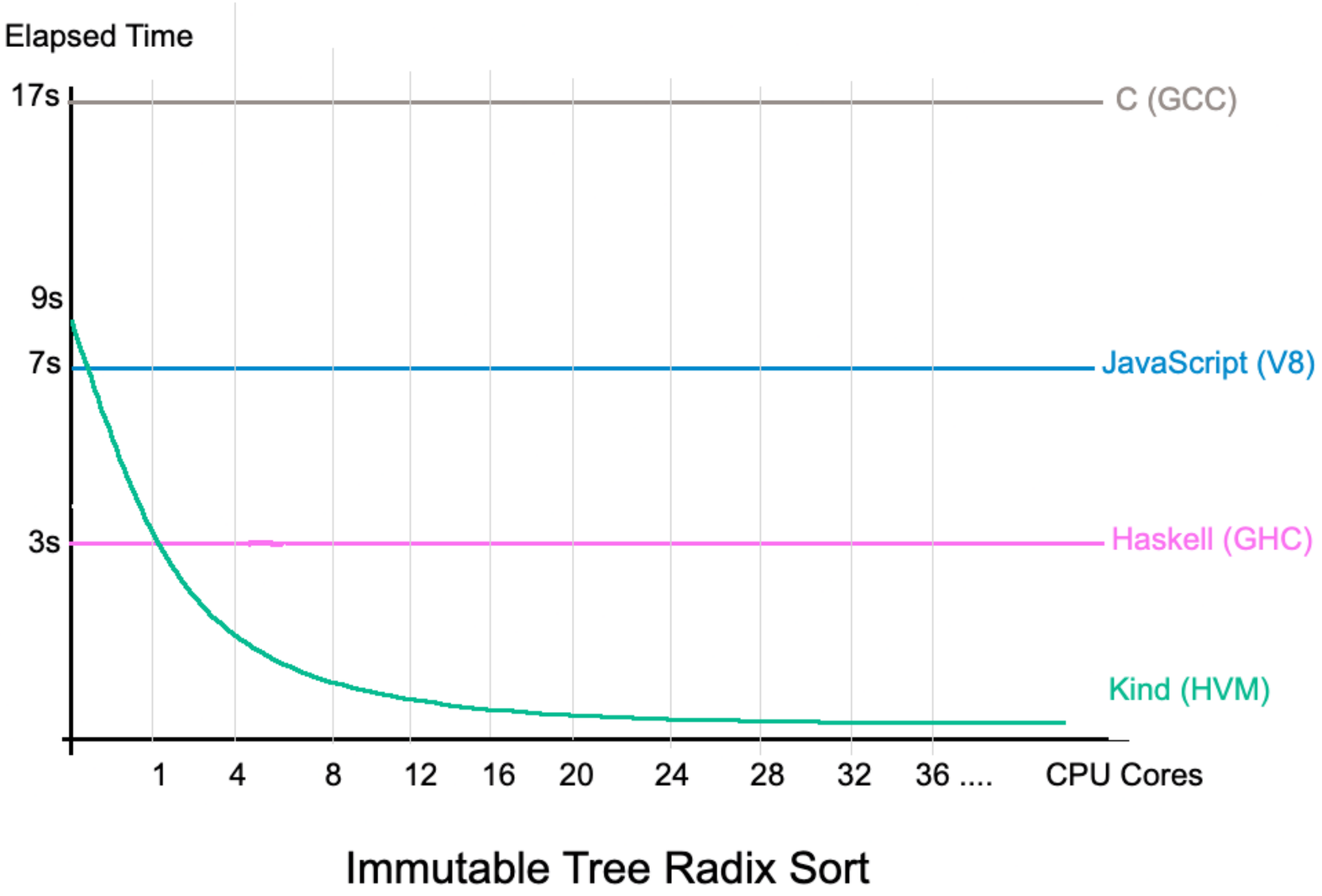
Automatic parallelism is solved by the **Interaction Calculus**, which *completes* the **Lambda Calculus** with **Interaction Net** semantics. Looks complex, but the key insights are simple!

- 1. Make **everything pure** (like Haskell) - *no side effects*
- 2. Make **everything linear** (like Rust) - *no shared references*
- 3. Add a **pervasive lazy cloner** ("*fan nodes*") - makes it *Turing complete*
- 4. Keep a **thread pool** with a **work stealing queue** of **interaction rules**

With these insights, we built the **Higher-order Virtual Machine (HVM)**.
An efficient, general-purpose, parallel runtime with near linear speedup!

HVM: https://github.com/Kindelia/HVM Interaction Calculus: https://github.com/Kindelia/Wikind/blob/master/IC/_.kind2

HVM's performance scales with cores!



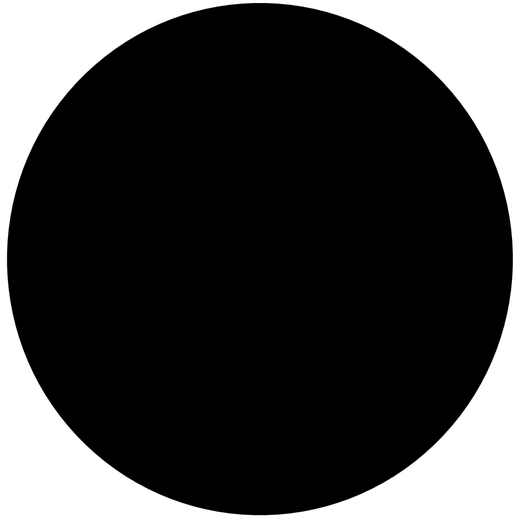
Immutable Tree Radix Sort

Benchmarks: https://github.com/victortaelin/hoc

Market #1

Software Development Industry

Size: it is big.

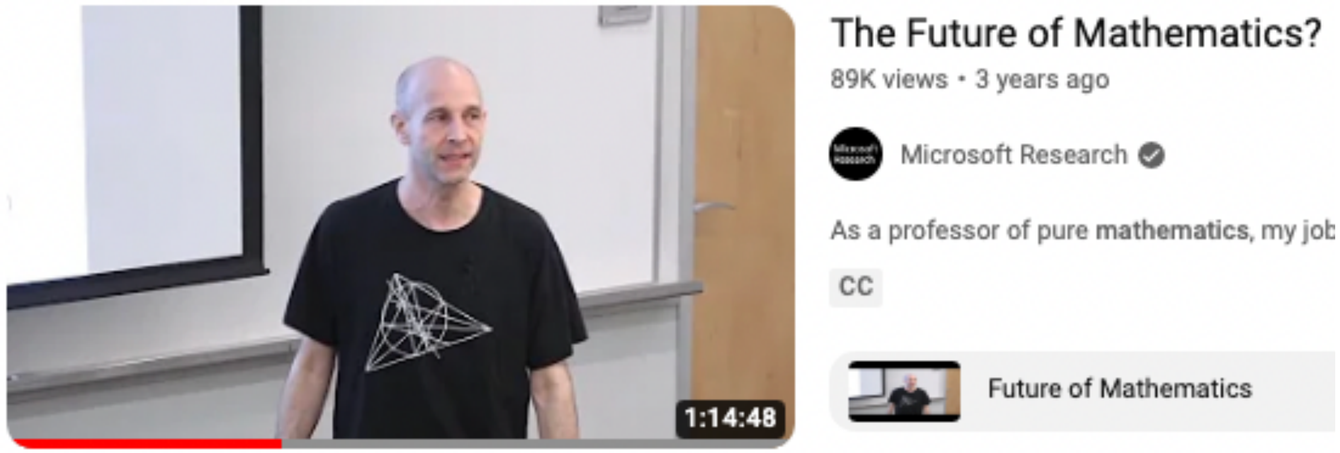


- We'll eat this market by developing **HVMCC** to compile **popular languages.**
- JavaScript, Python, Go, PHP and more. ***Soon, the entire world will run on the HVM.***

Market #2

Formalization of Mathematics

Size: it **will** be big.

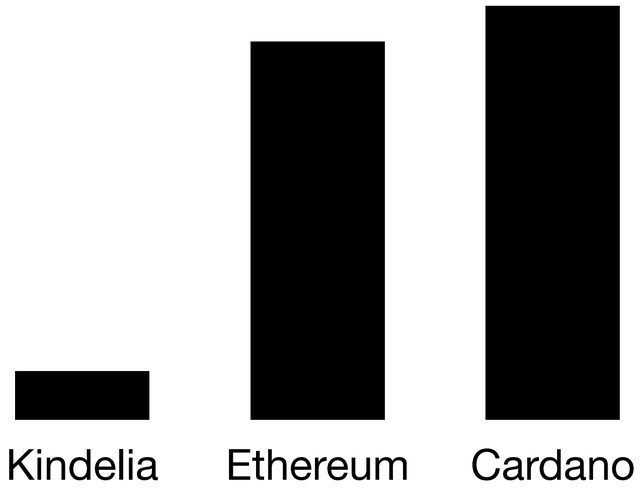


- We'll eat this market by making **Kind** the best proof assistant in the world!
- It is already the fastest. By far! ***Soon, all of mathematics will be digitalized on Kind-Lang.***

Benchmarks: https://github.com/kindelia/functional-benchmarks

Market #3

Decentralized Computers



Time to run Fib(32) on-chain

"*Wait, it outperforms Ethereum too? I'm shocked!*" - Nobody

```
// Supply Conservation Theorem

MyToken.send_conserves_supply
  (b : Map U60) // for any a balance map `b`
  (s : Address) // for any source address `s`
  (d : Address) // for any destination address `d`
  (a : U60)     // for any amount `a`

: // Sending `a` from `s` to `d` conserves `b` supply
  let old_supply = Map.sum b
  let new_supply = Map.sum (MyToken.send b d s a)
  Equal U60 old_sum new_sum

// Proof:
  let aux_0 = MyToken.send_conserves_aupply.aux_0 (MyT
  let aux_1 = Equal.rewrite aux_0 (Equal.mirror (MyTok
  let aux_2 = match aux_1 {
```

And it supports **native formal verification**!
Can't get more secure than a mathematical proof.

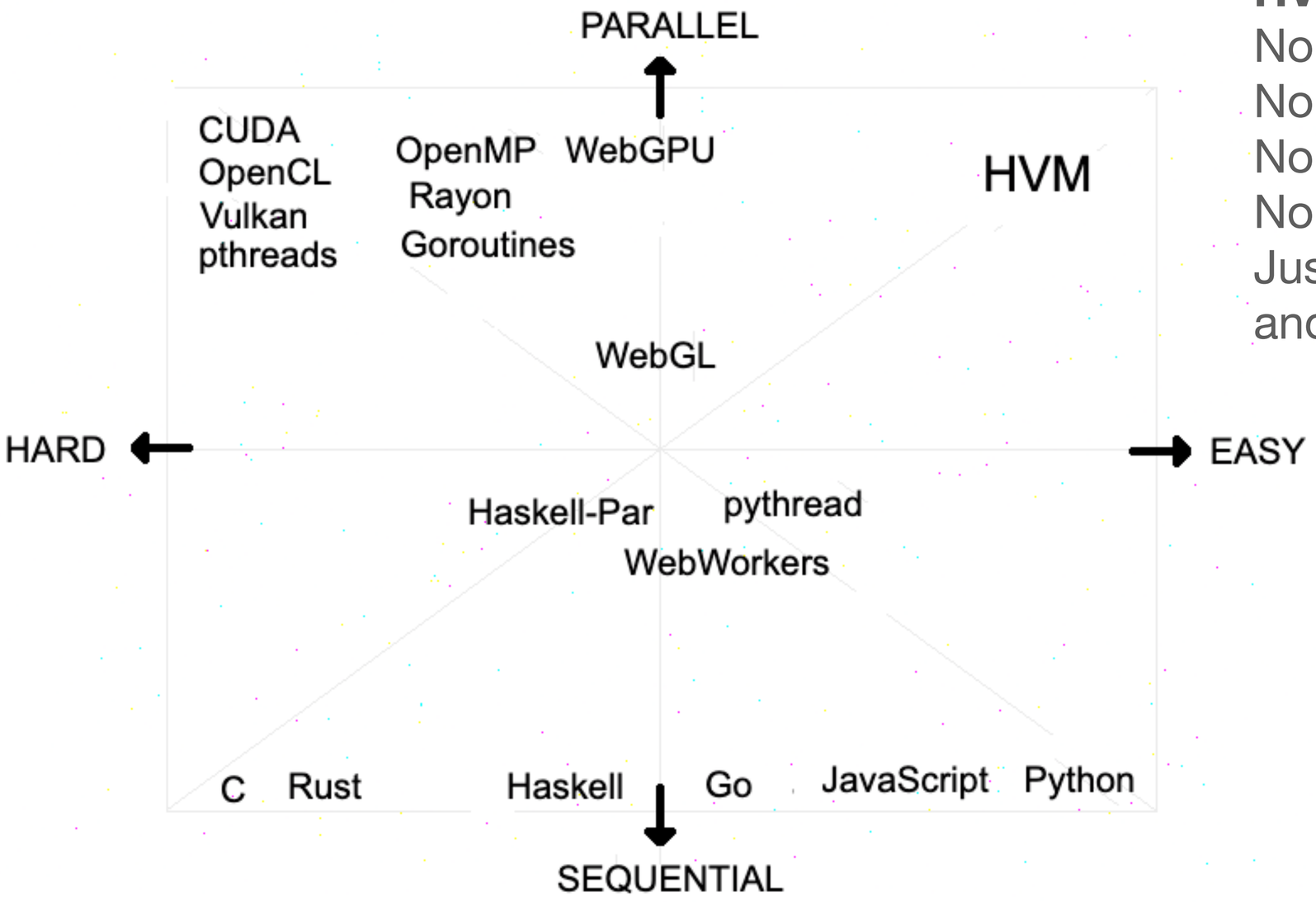
- We'll eat this market by making **Kindelia** the *fastest*, *most secure* layer 1 in the world.
- Kindelia has 867x more throughput than Ethereum. ***Soon, Kindelia will host massive metaverses.***

Business Model

We'll charge a license for large companies using HVM

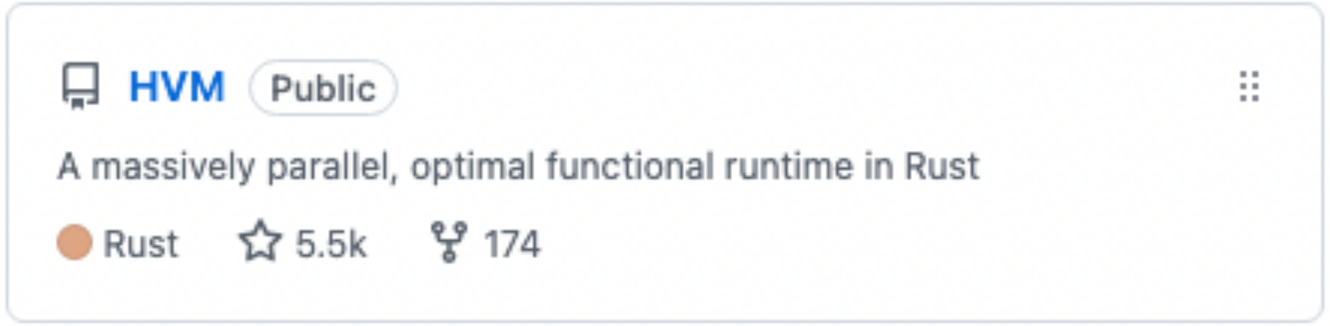
- Companies **will use HVM indirectly** as soon as they import any lib that uses it
  Mass adoption will be driven by HVMCC. Kindelia and Kind will push it further.
- Once a large company depends on HVM, we'll **sell a commercial license**
- In other words, we will do to **runtimes** what MongoDB did to **databases**
- This model is used by **Game Engines** (like Unreal 3D and Unity 3D)
- We'll popularize the concept of **Parallel Engines** for software development

Competition



HVM makes parallelism trivial.
No manual resource allocation
No explicit thread spawning
No non-deterministic debugging
No concurrency errors
Just write a code...
and run it in 100's cores!

Adoption



HVM's *prototype* has reached 5.5k stars on GitHub!

HVMCC will bring it to *popular languages*, creating another surge in usage.

Kind will drive the *formalization of mathematics*, further increasing adoption.

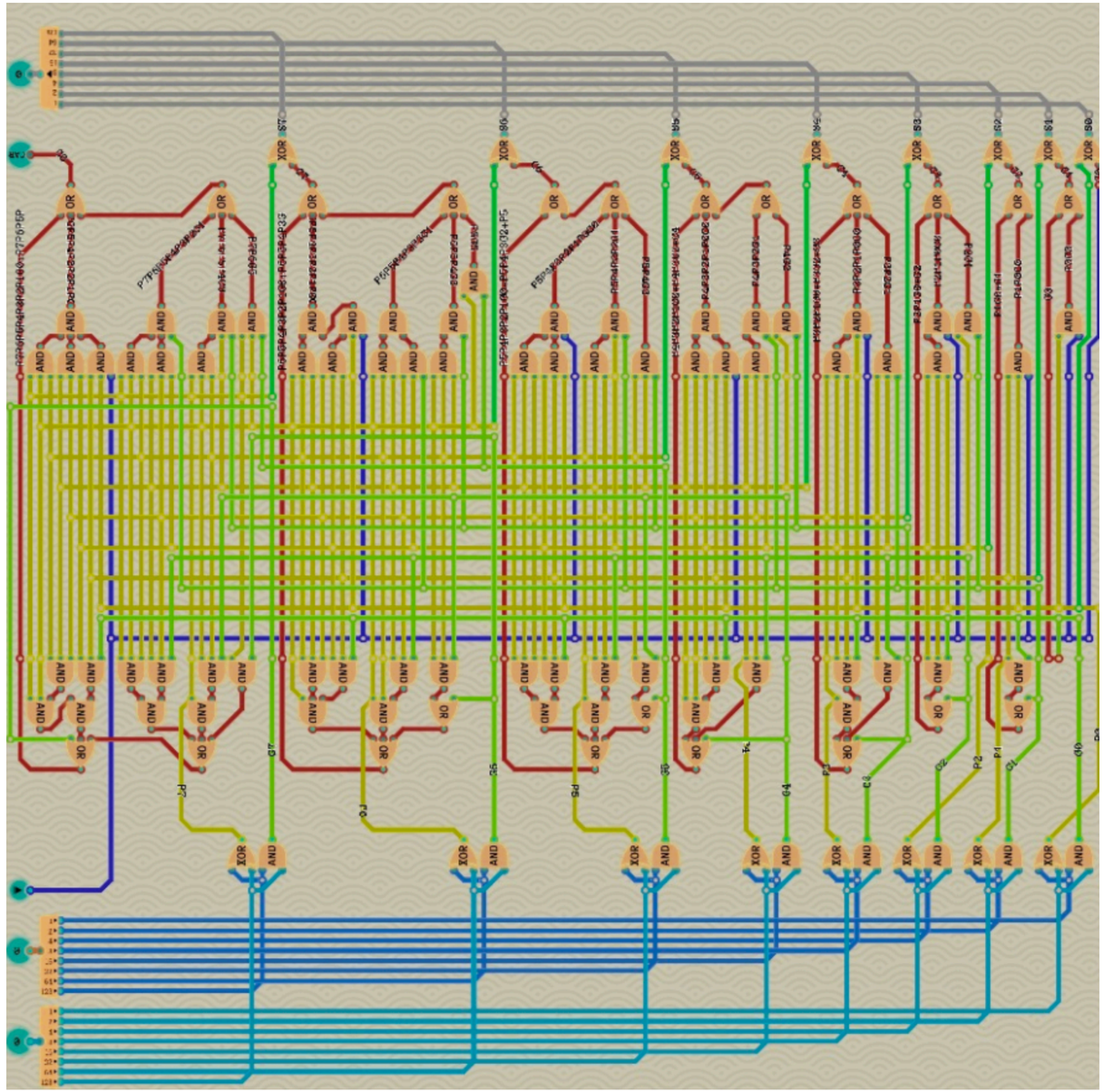
Kindelia will be the *fastest layer 1* in the world, showing the power of the HVM.

LongShot plan: HPU

One day, everything will run on interactional processors

- After **HVM** disrupts the software industry, we'll develop **HPU processors**
- These **Higher-order Processing Units** will perform *on-chip interactions*
- **Memory** and **Computation** will be **unified** in a single *interactional mesh*
- This parallel architecture will greatly outperform existing processors
- Interaction Nets will disrupt computing, science, math and AI

Hypothesis: an **interaction rule** would use **less circuit space** than an **8-bit adder**!



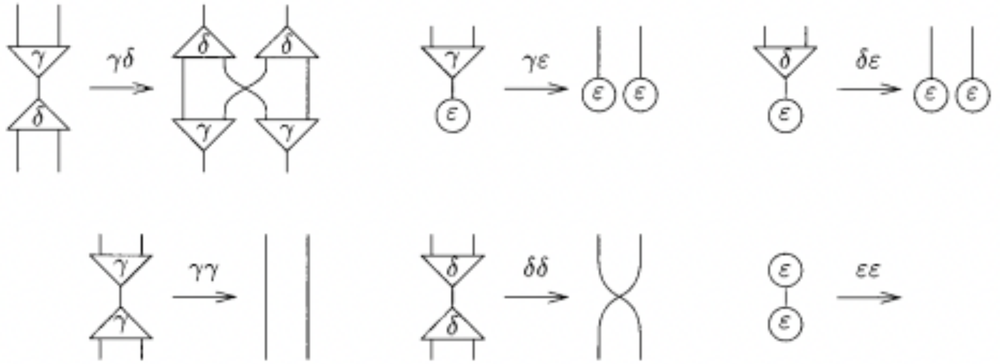


FIG. 2. Interaction rules for the combinators.

((λt.tt)) .(λx.x)

Seed Round

We're raising 10 million to make a trillion

- In our seed round, we'll offer 20% of HOC for a $10m ask
- These funds will be used to:
- 1. Hire a dev team to improve, polish and maintain the HVM
- 2. Develop and ship our adoption drivers: HVMCC, Kindelia and Kind-Lang
- 3. Structure the company and any necessary business expenses
- We've accomplished a lot. But there is still a lot to do.

It is not everyday that...

- Someone builds a new compiler with a 100k budget (!)
- ... that outperforms GCC, GHC and V8 by 10x (!!!)
- Once we raise $10m, *there is no going back*
- - we're not making a PDF converter
- - we're not competing with local restaurants
- + we're set to disrupt the entire tech industry
- + we're aiming for Google, nVidia, Intel, Apple
- **This is your best chance to own a big part of it!**

Founders

Victor Taelin

- codes daily since 2002
- functional programmer
- helped build Ethereum
- hacked HVM in Rust
- likes animes and cats
- hardstuck on LoL

Vitor Chiareli

- works out daily since 2002
- entrepreneur, actor and speaker
- reached #1 Trindamere in SA
- structured the company
- likes animes and cats
- not hardstuck on LoL