

Data (set) verificationism and Byzantine failures

Verificationism:

A form of logical empiricism (distributed observations)

Logical Positivism

verifiability criterion

Abstract:

I propose a set of questions about the IBM project as well as a set of general questions (fig 1) and observations, overall theoretical - described in numerous distinguished recent ~<20 years papers^1, as well as, excitingly, generally, known as positivism in philosophy.

DISCUSSION:

A technique in the philosophy of science dear to the heart of every *epistemologist* - and even *ontologist* and *deontologists*: [knowing, categorizing, and purposeful use (duty)], **Verification** is a cornerstone of knowledge. Verification is like justification; we use both to understand that some beliefs are not facts.

I have many questions about the scope, mission and progress of the enterprise. I find it exciting for many reasons, and, this: In philosophy -we use 'the philosophy of science' to provide a history of liturgical and didactic ([praxis] reading/thinking, teaching/learning), progress. Our new technology, is a manifestation of well established paradigms - and yet is, interestingly, excitedly, new. I see /hear great and profound applications as this, a highly verificationist technique, could propagate throughout nearly any digital system being it's better equivalent... e.g.,...

[And as example I hear a week ago a candidate for NJ governorship on NPR : *paraphrase*: "a blockchain ledger would save municipalities like NJ millions or hundreds of millions of dollars in 'misappropriated' funds: stolen and corrupt monies...]

TECHNICAL QUESTIONS AND OBSERVATIONS:**0.0**

Is IBM working on **centralized** (1) or **decentralized** (2) networks?

- 1) Privately secure (tautology) ^2
- 2) Public and durable

Data (set) verificationism and Byzantine failures

1.0

How is IBM preparing to deal with increasing database size?

How cumbersome are these (record) databases? (Bitcoin is ~ 100gb) ^4

2.0

Series of applications for IBM (**yes**) or (**no**)

.e.g.,

Financial Records (**yes**)

Questions: **(FIGURE ONE)**

Technical applications-

- *Financial Records*
- Large databases (think torrents)
- Encrypted records (,.i.e, medical records)
- Sophisticated records (dynamic computation) ^4
- Diamonds [the example IBM site [[link video](#)]]
- Archival record: museums like the MET (with million+ artifacts)
- Public records such as voter forms, DMV databases, DNC records?
- How readable are these records? ^3
- * Scalable processes? ^4

-----NOTES-----

^1

1999 [Miguel Castro and Barbra Liskov's](#)) PBFT [practical byzantine fault tolerance algorithm](#) (PBFT)

^2

Resemble corporate databases, are centralized, easy to manipulate. And tautological in that are extensions of an authority

^3 How ?

Query these records:

What substrate are they in - I would almost love to see XML or YAML objects (coupling readability, visuality of data, and result(ing) in human usable and highly machine readable records ^1 ^4

Data (set) verificationism and Byzantine failures

^4

This brings me to a GPU or graphical processing unit style mechanism: *Is this an application of the technology?* For instance, GPU implies 'micro records', and simultaneous record keeping and integration across vast - brute force - capabilities - both of data (input) as well as hardware.

I imagine (the 'micro record keeping';) can provide important capabilities for many highly automated and other functions not in the normal realm of the published element of the technology. A record keeping process analogous to this, having decentralized resources matrices may be faster: and a highly more capable mechanism of quality assurance in and out of a system, enabling (a) brute force process to become more streamlined, (reducing a backwards need of data integration) -- ^ see note #1-- ([since] each *note* contains both a stamp and a record- thus, for discrepancies, to a system of high 'fault coverage' - [have] low effectiveness)

Glossary & terms:

Safety Critical systems, error detecting codes, 'CRC'

PBFT : (Practical Byzantine Fault Tolerance)

FC : fault coverage

PFT : proof though testing

Low latency (microsecond)

Highly non corruptible records

High byzantine fault tolerance

Hyper ledger fabric

Byzantine fault tolerance:

Source congruency

The 'generals problem' : (involving: forged messages, loyal commanders, counterfeits, confederates, traitors.)

Phenomenological byzantine faults

DEFINITION/ Proof:

Adjusted signature for incoming message, (to other recipients) act of repeating (propagating) (blocks byzantine symptoms) resulting in measure of FC (fault coverage) in PFT (proof though testing)

Data (set) verificationism and Byzantine failures

Other notes

CONTACTS:

Maria Gaugler-Penn - Recruiting Partner IBM - Global Business Services

Maria Von Mindan - NA Talent Search Team Recruiter Supporting GBS/IX&MIBM CHQ - Human Resources

Nagarajan Seshadri - Capital Markets Technologist (Sr. Management Consultant) (2017-4-19)

Bala Vellanki - Senior Managing Consultant (2017-4-28)

Grace Pierce - GBS Recruiting Coordinator

Rangarajan Ramanujan - Managing Consultant - Blockchain & FS

Shelly Bury - Recruiting Coordinator IBM GBS Cognitive Process & Transformation

(possibly)

Venkat Rammohan - Business Transformation

OTHER REFERENCES:

npr.org/2017/using-the-blockchain-to-change-prisons

npr.org/tags/blockchain

VERIFICATION.unc.edu/~ujanel/Verif.htm

stanford.edu/logical-empiricism