VLSMs

analyzing faulty distributed systems
without resorting to Byzantine faults

Vlad Zamfir
Founder, Smart Transactions

Section 1

Validation Theory

Validating Labelled State Transition and Message Production Systems

A Theory for Modelling Faulty Distributed Systems

Vlad Zamfir
Ethereum Foundation
Runtime Verification, Inc.

Mihai Calancea
Runtime Verification, Inc.

Denisa Diaconescu
University of Bucharest
Runtime Verification, Inc.

Wojciech Kołowski
Runtime Verification, Inc.

Brandon Moore
Runtime Verification, Inc.

Karl Palmskog
KTH Royal Institute of Technology
Runtime Verification, Inc.

Traian Florin Șerbănuță
University of Bucharest
Runtime Verification, Inc.

Michael Stay
Pyrofex Corp

Dafina Trufas
University of Bucharest
Runtime Verification, Inc.

Jan Tušil
Masaryk University
Runtime Verification, Inc.

Validation Theory

- VLSM definition
- VLSM composition
- Validator definition

Validation Theory – VLSMs

Definition 2.1 (VLSM,). *A **Validating Labeled State transition and Message production system** (**VLSM**, for short) is a structure of the form*

$$\mathcal{V} = (L^{\mathcal{V}}, S^{\mathcal{V}}, S_0^{\mathcal{V}}, M^{\mathcal{V}}, M_0^{\mathcal{V}}, \tau^{\mathcal{V}}, \beta^{\mathcal{V}}),$$

where $L^{\mathcal{V}}$ is a set of labels, $(S_0^{\mathcal{V}} \subseteq)$ $S^{\mathcal{V}}$ is a non-empty set of (initial) states, $(M_0^{\mathcal{V}} \subseteq)$ $M^{\mathcal{V}}$ is a set of (initial) messages, $\tau^{\mathcal{V}} : L^{\mathcal{V}} \times S^{\mathcal{V}} \times (M^{\mathcal{V}}?) \to S^{\mathcal{V}} \times (M^{\mathcal{V}}?)$ is a transition function which takes as arguments a label, a state, and possibly a message, and outputs a state and possibly a message, while $\beta^{\mathcal{V}}$ is a validity constraint predicate on the inputs of the transition function, i.e., $\beta^{\mathcal{V}} \subseteq L^{\mathcal{V}} \times S^{\mathcal{V}} \times (M^{\mathcal{V}}?)$.[1]

State and Message Validity

$$
\begin{aligned}
S_0^v &= S_0 \text{ and } M_0^v = M_0 \cup \{\text{✖}\},\\
S_{n+1}^v &= S_n^v \cup \{\tau^s(l,s,m) \mid l \in L,\ s \in S_n^v,\ m \in M_n^v,\ \beta(l,s,m)\},\\
M_{n+1}^v &= M_n^v \cup \{\tau^m(l,s,m) \mid l \in L,\ s \in S_n^v,\ m \in M_n^v,\ \beta(l,s,m)\}.
\end{aligned}
$$

Validation Theory – VLSMs

If the validity predicate is satisfied, a transition or trace is called “constrained”, then

A valid trace is a constrained trace in which any input message is a valid message

Validation Theory – VLSM Composition

Definition 3.1 (Free composition,). *The **free VLSM composition** of* $\{\mathcal{V}_i\}_{i=1}^{n}$ *is the VLSM*

$$\sum_{i=1}^{n} \mathcal{V}_i = (L, S, S_0, M, M_0, \tau, \beta),$$

where $L = \bigcup_{i=1}^{n} \{i\} \times L_i$ *is the disjoint union of labels,* $S = \prod_{i=1}^{n} S_i$ *is the product of states,* $S_0 = \prod_{i=1}^{n} S_{i,0}$ *is the product of initial states,* M *is the same set of messages as for each* $\mathcal{V}_i$*,* $M_0 = \bigcup_{i=1}^{n} M_{i,0}$ *is the union of all initial messages,* $\tau : L \times S \times M? \to S \times M?$ *and* $\beta \subseteq L \times S \times M?$ *are defined component-wise, guided by labels, i.e.,*

$$\tau(\langle j, l_j \rangle, \langle s_1, \_ s_n \rangle, m) = \left(\langle s_1, \_, s_{j-1}, \tau_j^s(l_j, s_j, m), s_{j+1}, \_ s_n \rangle, \tau_j^m(l_j, s_j, m) \right),$$

$$\beta(\langle j, l_j \rangle, \langle s_1, \_, s_n \rangle, m) = \beta_j(l_j, s_j, m).$$

Validation Theory – VLSM Composition

Definition 3.2 (Constrained composition,). *A **composition constraint** φ is a predicate additionally filtering the inputs for the composed transition function, $\varphi \subseteq L \times S \times M?$. The **constrained VLSM composition under** φ of $\{\mathcal{V}_i\}_{i=1}^{n}$ is the VLSM which has the same components as the free composition, except for the validity predicate which is further constrained by φ, namely*[4]

$$\left(\sum_{i=1}^{n} \mathcal{V}_i\right)\Big|_{\varphi} = (L, S, S_0, M, M_0, \tau, \beta \wedge \varphi).$$

Validation Theory – Validators

4.1 Definition of validator

Let $\mathcal{V} = \left(\sum_{i=1}^{n} \mathcal{V}_i\right)\Big|_{\varphi}$ be the composition under φ of $\{\mathcal{V}_i\}_{i=1}^{n}$. Let $j \in \{1, .., n\}$ be the index of a component.

Definition 4.1 (Validator,). *The component $\mathcal{V}_j$ is a **validator** for $\mathcal{V}$ if any constrained transition from a constrained state in $\mathcal{V}_j$, $s_j \xrightarrow[m \to m']{l} s'_j$, can be "lifted" to a valid transition in $\mathcal{V}$, $\sigma \xRightarrow[m \to m']{\langle j,l \rangle} \sigma'$, such that the j^{th} components of σ and σ' are s_j and s'_j, respectively.*

Section 2

Equivocation Theory

Equivocation Theory

- Evidence of equivocation
  - Local
  - Global
- Limiting equivocation
  - By fixed set
  - By weight
- Models of equivocation
  - State
  - Message

Local Evidence of Equivocation

Definition 5.1 (Local evidence of equivocation,). *A pair of messages is a* ***local evidence of equivocation*** *for their sender in a component state of a VLSM if*

(1) *the messages have the same sender,*

(2) *the messages have been (indirectly) observed in the component state, and*

(3) *the messages could not have been produced by their sender in a single run of the protocol.*

Global Evidence of Equivocation

Definition 5.2 (Global evidence of equivocation,). *A message is a* ***global evidence of equivocation*** *for its sender in a composite state of a composite VLSM if*

(1) *the message has been (indirectly) observed in the composite state, and*

(2) *the message was not observed as a sent message in the composite state.*

Theorem 5.1 (). *Let* $\mathcal{V} = \left(\sum_{i=1}^{n} \mathcal{V}_i\right)\Big|_{\varphi}$ *be the constrained composition under* φ *of* $\{\mathcal{V}_i\}_{i=1}^{n}$. *For any constrained component state* s *and any constrained composite state* σ *such that one of its components is* s, *we have*

$$localEqv(s) \subseteq globalEqv(\sigma).$$

Non-Equivocation and Limited Equivocation

We can use a VLSM (global) composition constraint to disallow (global) evidence of equivocation

$$\sum_{j \in globalEqv_{full}(\sigma')} weight(j) < t$$

We can use a full node assumption

We can limit equivocation to a subset of components

$$\sum_{j \in localEqv_{full}(s)} weight(j) < t.$$

And we can assign weights and threshold t

State Equivocations and Message Equivocations

A state equivocator is allowed to run parallel copies of itself by forking existing states or spawning new machines

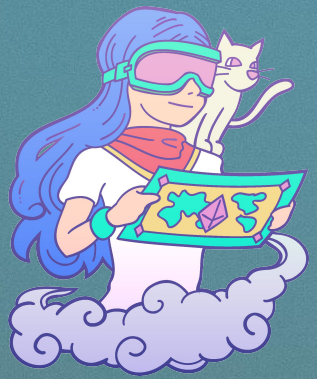
In the message equivocation model, an equivocation is instead understood as the receipt of a message that wasn't sent in this trace

Theorem 6.2. *Under full node assumption for all components,*

1. *The trace reduct of a valid trace of the state equivocation model is a valid trace for the message equivocation model (↗).*
2. *Each valid trace for the message equivocation model can be "lifted" to a valid trace for the state equivocation model such that its trace reduct is the original trace (↗).*

Section 3

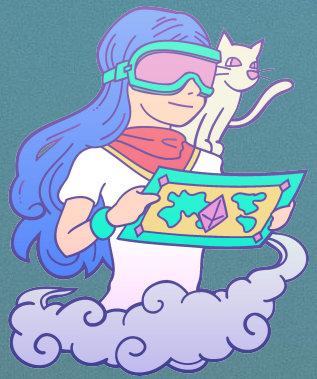
Reduction of Byzantine faults to Equivocation faults



Modeling Byzantine faults

We model Byzantine faults by replacing VLSM components with nodes that can send any message at any time.

(But constrain the faulty components so that they can’t forge messages on behalf, and have a full node assumption)

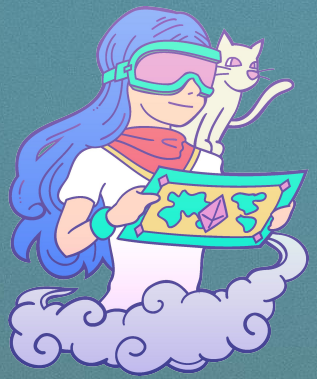


Modeling Byzantine faults

Replacing some equivocation-limited validators with Byzantine components,

we find that the remaining validators have **exactly** the traces they would have when composed with equivocating components

$$Tr(\mathcal{V}^{B}_{Byz})|_{NonByz} = Tr(\mathcal{V}^{B}_{m\_eqv})|_{NonByz}.$$



Modeling Byzantine faults

Because if the validators can transition on a message from a Byzantine sender,

then there is a composite state which also has that same validator transition,

but where the Byzantine nodes are replaced with equivocating validators.

Theorem 7.2 (). *If all components are validators for the t-limited message equivocation model $\mathcal{V}^{<1}_{m\_eqv}$, then the possible behaviors of the non-equivocating components are the same under t-limited Byzantine behaviour as under t-limited equivocation behavior.*

Section 4

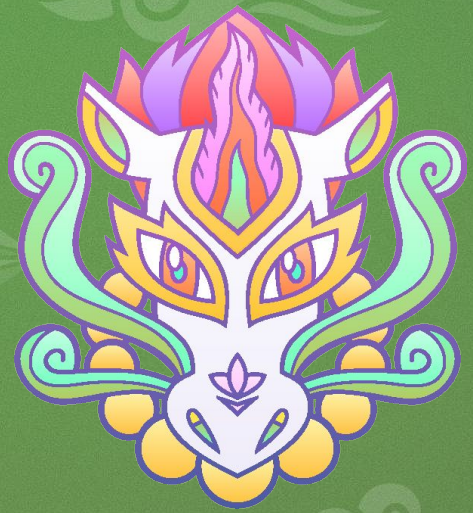
Conclusion

Conclusion

We showed that equivocation faults are exactly as expressive as Byzantine faults when it comes to their influence on equivocation-limited validators

This forms an alternative basis for analyzing faulty distributed systems.

We leave it to later work to relax the full node assumption and to treat synchronization faults.



Thank you!

Vlad Zamfir
Founder, Smart Transactions
v@stxn.io
@VladZamfir