Parameterized Verification with

Byzantine Model Checker

Igor Konnov

`<igor@informal.systems>`

Tutorial at FORTE, June 15, 2020

streaming from Vienna / Austria to Valletta / Malta

informal

INTERCHAIN
FOUNDATION

Co-authors and contributors:

Helmut Veith

Ilina Stoilkovska

Marijana Lazić

Jure Kukovec

Francesco Spegni

Ulrich Schmid

Roderick Bloem

Florian Zuleger

Nathalie Bertrand

Josef Widder

Annu Gmeiner

Timeline

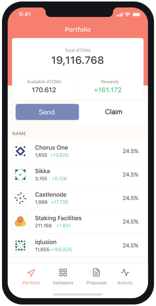
Fault-tolerant distributed algorithms and threshold automata

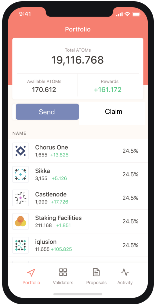
Safety of **asynchronous** threshold-guarded algorithms

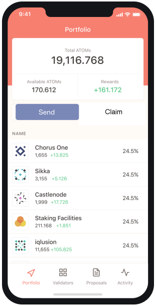
Liveness and **beyond** asynchronous algorithms

our inspiration:

distributed consensus

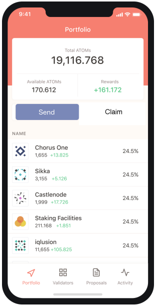

9:41
Portfolio
Total ATOMs
19,116.768
Available ATOMs
170.612
Rewards
+161.172
Send
Claim
NAME
Chorus One
1,655 +13.825
24.5%
Sikka
3,155 +5.126
24.5%
Castlenode
1,999 +17.726
24.5%
Staking Facilities
211.168 +1.851
24.5%
iqlusion
11,655 +105.825
24.5%
Portfolio
Validators
Proposals
Activity

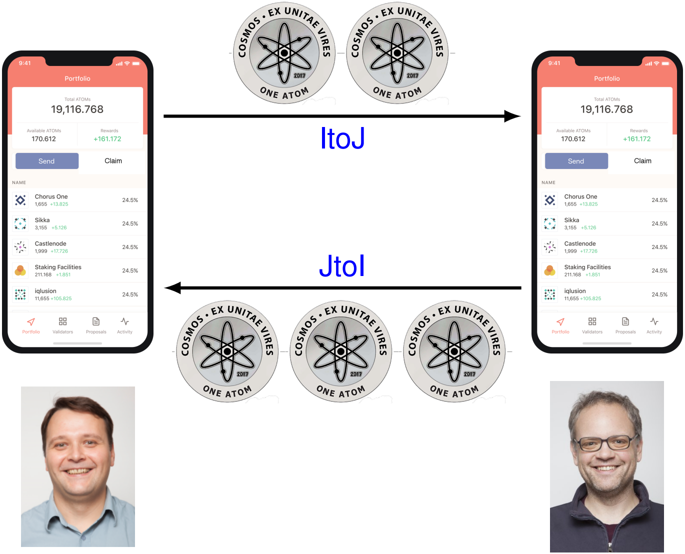

9:41
Portfolio
Total ATOMs
19,116.768
Available ATOMs
170.612
Rewards
+161.172
Send
Claim
NAME
Chorus One
1,655 +13.825
24.5%
Sikka
3,155 +5.126
24.5%
Castlenode
1,999 +17.726
24.5%
Staking Facilities
211.168 +1.851
24.5%
iqlusion
11,655 +105.825
24.5%
Portfolio
Validators
Proposals
Activity


9:41
Portfolio
Total ATOMs
19,116.768
Available ATOMs
170.612
Rewards
+161.172
Send
Claim
NAME
Chorus One
1,655 +13.825
24.5%
Sikka
3,155 +5.126
24.5%
Castlenode
1,999 +17.726
24.5%
Staking Facilities
211.168 +1.851
24.5%
iqlusion
11,655 +105.825
24.5%
Portfolio
Validators
Proposals
Activity

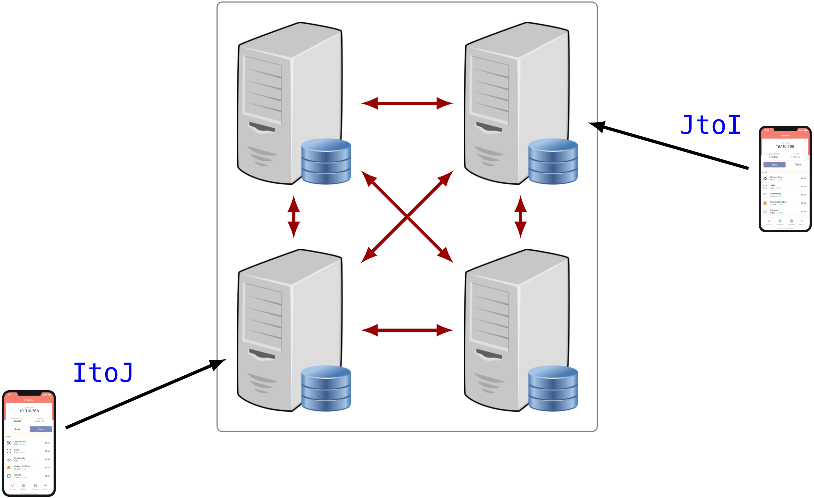

COSMOS • EX UNITAE VIRES
2017
ONE ATOM
COSMOS • EX UNITAE VIRES
2017
ONE ATOM

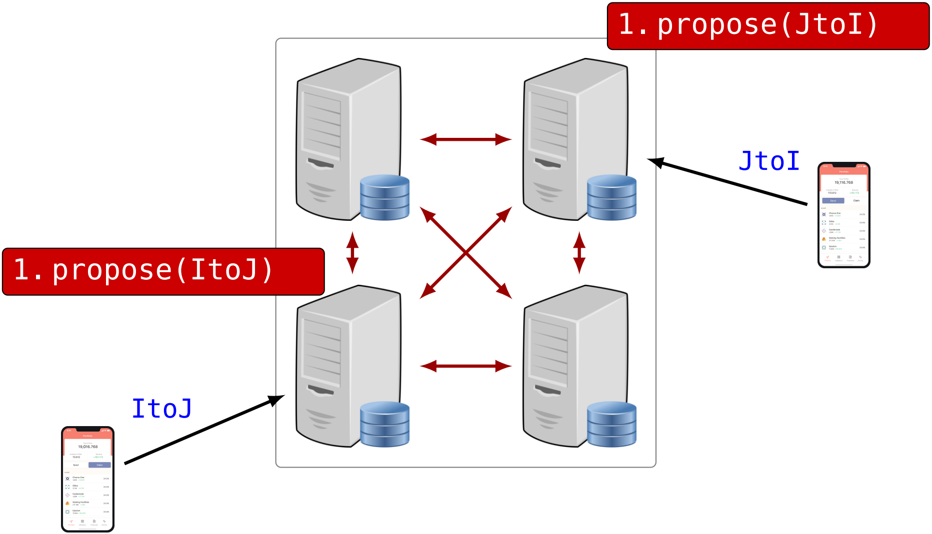
ItoJ

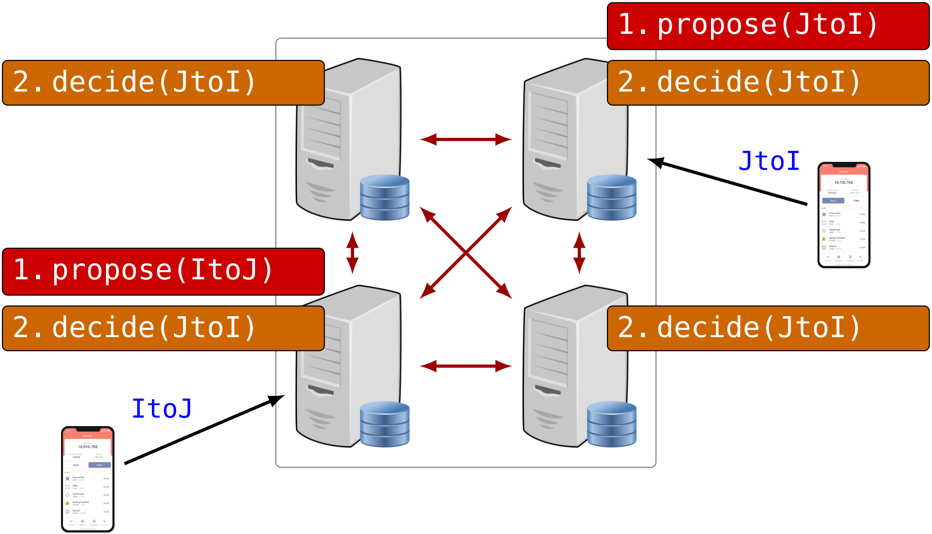

9:41
Portfolio
Total ATOMs
19,116.768
Available ATOMs
170.612
Rewards
+161.172
Send
Claim
NAME
Chorus One
1,655 +13.825
24.5%
Sikka
3,155 +5.126
24.5%
Castlenode
1,999 +17.726
24.5%
Staking Facilities
211.168 +1.851
24.5%
iqlusion
11,655 +105.825
24.5%
Portfolio
Validators
Proposals
Activity



COSMOS • EX UNITAE VIRES
2017
ONE ATOM
ItoJ
JtoI
9:41
Portfolio
Total ATOMs
19,116.768
Available ATOMs
170.612
Rewards
+161.172
Send
Claim
NAME
Chorus One
1,655 +13.825
24.5%
Sikka
3,155 +5.126
24.5%
Castlenode
1,999 +17.726
24.5%
Staking Facilities
211.168 +1.851
24.5%
iqlusion
11,655 +105.825
24.5%
Portfolio
Validators
Proposals
Activity


JtoI
ItoJ


1. propose(JtoI)
JtoI
1. propose(ItoJ)
ItoJ


1. propose(JtoI)
2. decide(JtoI)
2. decide(JtoI)
JtoI
1. propose(ItoJ)
2. decide(JtoI)
2. decide(JtoI)
ItoJ

Problem of Distributed Consensus

A distributed algorithm for n replicas

every replica proposes a value $w \in V$

Termination

every correct replica eventually decides on a value $v \in V$

Agreement

if a replica decides on v, no replica decides on $V \setminus \{v\}$

Validity

if a replica decides on v, the value v was proposed earlier

crucial to verify safety and liveness

Problem of Distributed Consensus

A distributed algorithm for n replicas

every replica proposes a value $w \in V$

Termination

every correct replica eventually decides on a value $v \in V$

Agreement

if a replica decides on v, no replica decides on $V \setminus \{v\}$

Validity

if a replica decides on v, the value v was proposed earlier

crucial to verify safety and liveness

~~Termination~~

every replica eventually decides on a value $v \in V$

Agreement

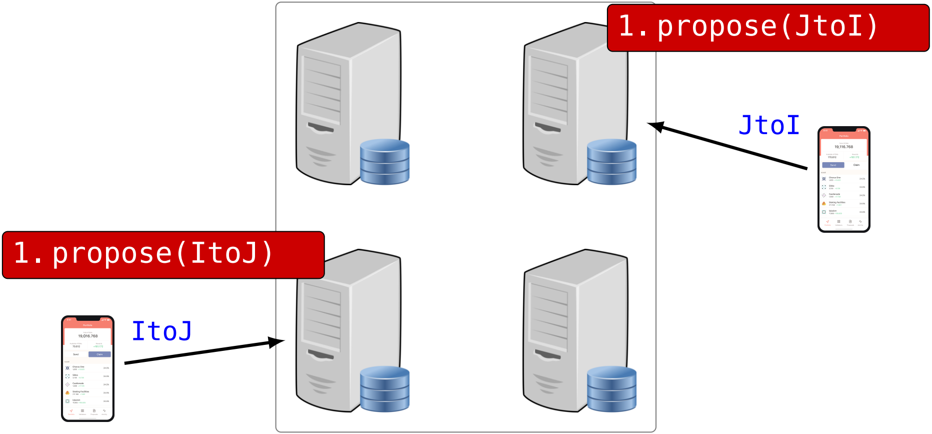
if a replica decides on v, no replica decides on $V \setminus \{v\}$

Validity

if a replica decides on v, the value v was proposed earlier

consensus without termination:

do nothing!



Naïve majority voting

n replicas follow the code:

input $u_i \in \{0, 1\}$
send u_i **to** all
wait until some value $v_i \in \{0, 1\}$ is **received** $\lceil \frac{n+1}{2} \rceil$ times
decide **on** v_j

Does it satisfy Validity, Agreement, and Termination?

What is the computation model?

n replicas follow the code:

input $u_i \in \{0, 1\}$
send u_i **to** all
wait until some value $v_i \in \{0, 1\}$ is **received** $\lceil \frac{n+1}{2} \rceil$ times
decide **on** v_j

Does it satisfy Validity, Agreement, and Termination?

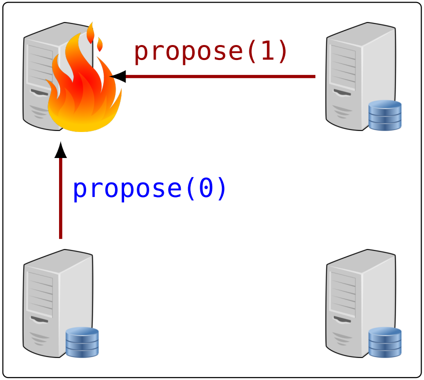
What is the computation model?

Asynchronous systems with faults

Various processor speeds

Various message delays, unbounded but finite

crashes



later today,

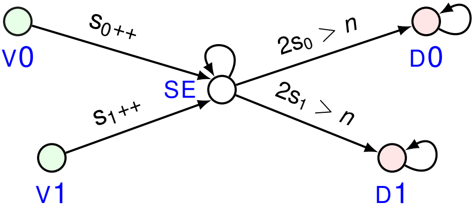
Byzantine

...

Formalizing pseudo-code…

```
input u_i ∈ {0,1}
send u_i to all
wait until some value v_i ∈ {0,1} is received ⌈(n+1)/2⌉ times
decide on v_j
```

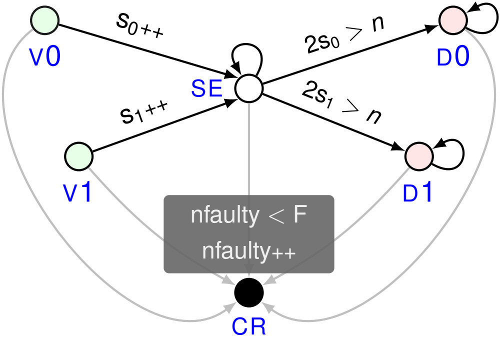
as a **threshold automaton:**



Formalizing pseudo-code…

input $u_i \in \{0, 1\}$
send u_i **to** all
wait until some value $v_i \in \{0, 1\}$ is **received** $\lceil \frac{n+1}{2} \rceil$ times
decide **on** v_j

as a **threshold automaton:**



Formalizing the distributed system...

replica 1:	$u_1 = 0$	$\mathsf{V0} \xrightarrow{\mathsf{s}_0\text{++}} \mathsf{SE}$		
replica 2:	$u_2 = 1$		$\mathsf{V1} \xrightarrow{\mathsf{s}_1\text{++}} \mathsf{SE}$	
replica 3:	$u_3 = 0$			
replica 4:	$u_4 = 0$			$\mathsf{V0} \xrightarrow{\mathsf{s}_0\text{++}} \mathsf{SE}$

as a **counter system:**

$\boldsymbol{\kappa}_{\mathsf{V0}} = 3$	$\boldsymbol{\kappa}_{\mathsf{V0}} = 2$		$\boldsymbol{\kappa}_{\mathsf{V0}} = 1$
$\boldsymbol{\kappa}_{\mathsf{V1}} = 1$		$\boldsymbol{\kappa}_{\mathsf{V1}} = 0$	
$\boldsymbol{\kappa}_{\mathsf{SE}} = 0$	$\boldsymbol{\kappa}_{\mathsf{SE}} = 1$	$\boldsymbol{\kappa}_{\mathsf{SE}} = 2$	$\boldsymbol{\kappa}_{\mathsf{SE}} = 3$
$\mathsf{s}_0 = 0$	$\mathsf{s}_0 = 1$		$\mathsf{s}_0 = 2$
$\mathsf{s}_1 = 0$		$\mathsf{s}_1 = 1$	
...			

Formalizing properties...

Termination: every replica eventually decides on a value $v \in V$

Agreement: if a replica decides on v, no replica decides on $V \setminus \{v\}$

Validity: if a replica decides on v, the value v was proposed earlier

as **temporal formulas**:

Termination: *fairness* $\rightarrow \Diamond\,(\boldsymbol{\kappa}_{\mathsf{V0}} = 0 \wedge \boldsymbol{\kappa}_{\mathsf{V1}} = 0 \wedge \boldsymbol{\kappa}_{\mathsf{SE}} = 0)$

Agreement: $\Box\,(\boldsymbol{\kappa}_{\mathsf{D0}} = 0 \vee \boldsymbol{\kappa}_{\mathsf{D1}} = 0)$

0-Validity: $\boldsymbol{\kappa}_{\mathsf{V1}} = 0 \rightarrow \Box\,(\boldsymbol{\kappa}_{\mathsf{D1}} = 0)$

1-Validity: $\boldsymbol{\kappa}_{\mathsf{V0}} = 0 \rightarrow \Box\,(\boldsymbol{\kappa}_{\mathsf{D0}} = 0)$

Let’s ask ByMC. . .

```
user@bymc: ~/fault-tolerant-benchmarks/forte20
user@bymc: ~/fault-tolerant-benchmarks/forte20 80x29
  --limit-time: limit (in seconds) cpu time of subprocesses (ulimit -t)
  --limit-mem: limit (in MB) virtual memory of subprocesses (ulimit -v)
  -h|--help: show this help message

 bymc_options are as follows:
  -O schema.tech=ltl          (default, safety + liveness as in POPL'17)
  -O schema.tech=ltl-mpi      (parallel safety + liveness as in ISOLA'18)
  -O schema.tech=cav15        (reachability as in CAV'15)
  --smt 'lib2|z3|-smt2|-in'   (default, use z3 as the backend solver)
  --smt 'lib2|mysolver|arg1|arg2|arg3'  (use an SMT2 solver)
  --smt 'yices'               (use yices 1.x as the backend solver, DEPRECATED)
  -v               (verbose output, all debug messages get printed)

 Fine tuning of schema.tech=ltl:
  -O schema.incremental=1 (enable the incremental solver, default: 0)

  -O schema.noflowopt=1  (disable the control flow optimizations, default: 0
                          may lead to a combinatorial explosion of guards)
  -O schema.noreachopt=1 (disable the reachability optimization, default: 0
                          i.e., reachability is not checked on-the-fly)
  -O schema.noadaptive=1 (disable the adaptive reachability optimization, defaul
t: 0
                          i.e., the tool will not try to choose between
                          enabling/disabling the reachability optimization)
  -O schema.noguardpreds=1 (do not introduce predicates for
                            the threshold guards, default: 0)
  -O schema.compute-nschemas=1 (always compute the total number of
                                schemas, even if takes long, default: 0)
user@bymc:~/fault-tolerant-benchmarks/forte20$
```

Time for questions!

[bit.ly/2z8mE51]

(the examples and links for this talk)