Parameterized and Runtime-tunable Snapshot Isolation in Distributed Transactional Key-value Stores

Hengfeng Wei, Yu Huang, Jian Lu

Nanjing University, China

September 19, 2017

Parameterized and Runtime-tunable Snapshot Isolation

RVSI: Relaxed Version Snapshot Isolation

1. Experimental Evaluation

Impacts of RVSI specification

on the *transaction abort rates* in various scenarios

Impacts of RVSI specification

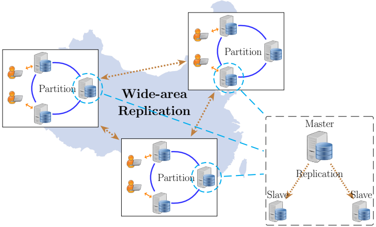
on the *transaction abort rates* in various scenarios

What about performance?

- Not done yet in this work
- RVSI-MS and RVSI-MP protocols in CHAMELEON are simple

CHAMELEON prototype on Aliyun:

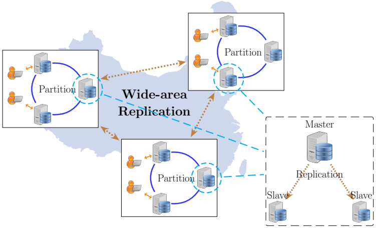
- 3 datacenters [1]
- 3 nodes in each datacenter
- Partition & Replication
- Clients in our lab [2]



[1] Located in East China, North China, and South China, respectively.
[2] Located in East China.
[3] https://github.com/hengxin/aliyun-ping-traces

CHAMELEON prototype on Aliyun:

- 3 datacenters [1]
- 3 nodes in each datacenter
- Partition & Replication
- Clients in our lab [2]



(One-way) delays among nodes [3]:

Within datacenter: $1 \sim 2$ms

Across datacenters: $15 \sim 25$ms

Clients to nodes: $15 \sim 20$ms

[1] Located in East China, North China, and South China, respectively.
[2] Located in East China.
[3] `https://github.com/hengxin/aliyun-ping-traces`

Three categories of workload parameters for experiments on Aliyun.

Parameter	Value	Explanation
#keys	$25 = 5$ (rows) $\times$ 5 (columns)	
#clients	5, 10, 15, 20, 25, 30	
#txs/client	1000	
#ops/tx	$\sim$ Binomial(20, 0.5)	
rwRatio	1:2, 1:1, 4:1	#reads/#writes
zipfExponent	1	parameter for Zipfian distribution
minInterval	0ms	min/max/mean inter-transaction time
maxInterval	10ms	
meanInterval	5ms	
(k_1, k_2, k_3)	(1,0,0) (1,1,0) (1,1,1) (2,0,0) (2,0,1) (2,1,1)	

Three categories of workload parameters for experiments on Aliyun.

Parameter	Value	Explanation
#keys	$25 = 5$ (rows) $\times$ 5 (columns)	
#clients	5, 10, 15, 20, 25, 30	
#txs/client	1000	
#ops/tx	$\sim$ Binomial(20, 0.5)	
rwRatio	1:2, 1:1, 4:1	#reads/#writes
zipfExponent	1	parameter for Zipfian distribution
minInterval	0ms	min/max/mean inter-transaction time
maxInterval	10ms	
meanInterval	5ms	
(k_1, k_2, k_3)	(1,0,0) (1,1,0) (1,1,1) (2,0,0) (2,0,1) (2,1,1)	

Transactions abort for two reasons:

- "vc-aborted": RVSI version constraints violated
- "wcf-aborted": the WCF property violated

[1] https://github.com/hengxin/chameleon-transactional-kvstore

Transactions abort for two reasons:

- "vc-aborted": RVSI version constraints violated
- "wcf-aborted": the WCF property violated

Transaction abort rates due to "vc-aborted" are *sensitive* to different values of k_1, k_2, or k_3,

[1] https://github.com/hengxin/chameleon-transactional-kvstore

Transactions abort for two reasons:

- "vc-aborted": RVSI version constraints violated
- "wcf-aborted": the WCF property violated

Transaction abort rates due to "vc-aborted" are *sensitive* to different values of k_1, k_2, or k_3, but those due to "wcf-aborted" are not.

[1] `https://github.com/hengxin/chameleon-transactional-kvstore`

Transactions abort for two reasons:

- "vc-aborted": RVSI version constraints violated
- "wcf-aborted": the WCF property violated

Transaction abort rates due to "vc-aborted" are *sensitive* to different values of k_1, k_2, or k_3, but those due to "wcf-aborted" are not.

$$h \in \text{RVSI} \iff h \in k_1\text{-BV} \cap k_2\text{-FV} \cap k_3\text{-SV} \cap \text{WCF}.$$

[1] https://github.com/hengxin/chameleon-transactional-kvstore

Transactions abort for two reasons:

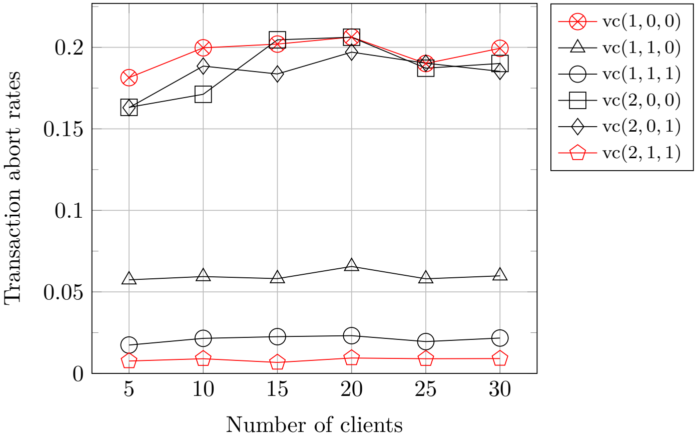
- "vc-aborted": RVSI version constraints violated
- "wcf-aborted": the WCF property violated

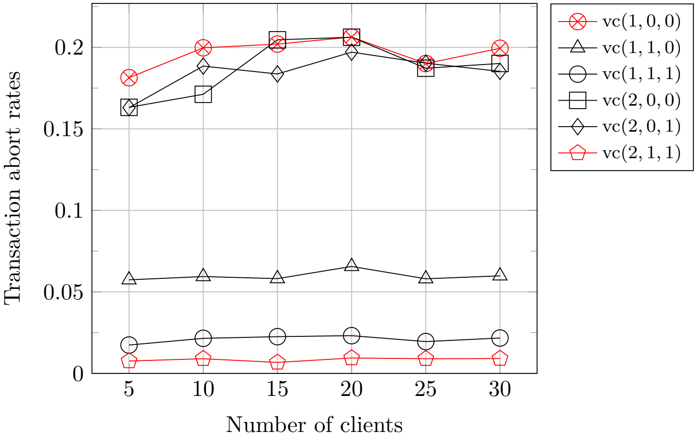
Transaction abort rates due to "vc-aborted" are *sensitive* to different values of k_1, k_2, or k_3, but those due to "wcf-aborted" are not.

$$h \in \text{RVSI} \iff h \in k_1\text{-BV} \cap k_2\text{-FV} \cap k_3\text{-SV} \cap \text{WCF}.$$

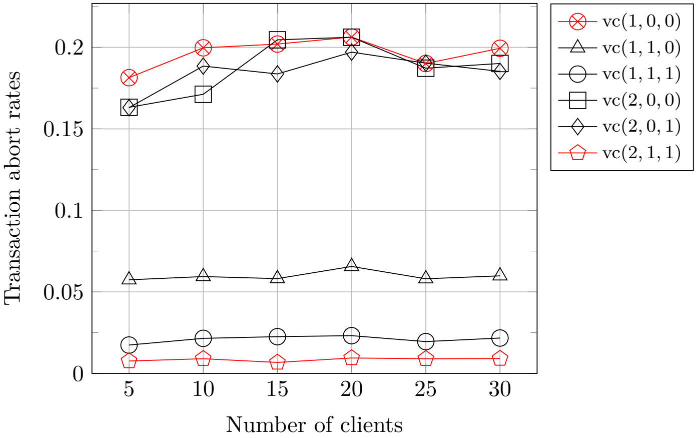
We report the results under the read-frequent workloads [1].

[1] https://github.com/hengxin/chameleon-transactional-kvstore


Transaction abort rates
0.2
0.15
0.1
0.05
0
5
10
15
20
25
30
Number of clients
vc(1, 0, 0)
vc(1, 1, 0)
vc(1, 1, 1)
vc(2, 0, 0)
vc(2, 0, 1)
vc(2, 1, 1)

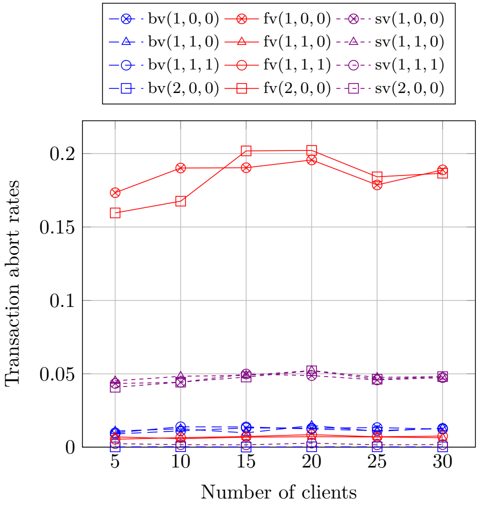


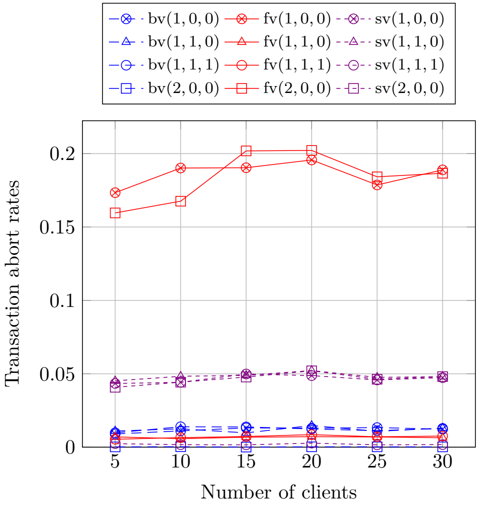
The transaction abort rates due to “vc-aborted”



The transaction abort rates due to “vc-aborted” can be greatly reduced by slightly increasing the values of k_1, k_2, or k_3:

$$vc(1,0,0) = 0.1994 \implies vc(2,1,1) = 0.0091 \quad (\#\text{clients} = 30)$$


bv(1, 0, 0)
fv(1, 0, 0)
sv(1, 0, 0)
bv(1, 1, 0)
fv(1, 1, 0)
sv(1, 1, 0)
bv(1, 1, 1)
fv(1, 1, 1)
sv(1, 1, 1)
bv(2, 0, 0)
fv(2, 0, 0)
sv(2, 0, 0)
Transaction abort rates
0.2
0.15
0.1
0.05
0
5
10
15
20
25
30
Number of clients



Most "vc-aborted" transactions abort because of violating k_2-FV.

$$fv(1,0,0) = 0.1889 \implies fv(2,0,0) = 0.1866 \implies fv(1,1,0) = 0.0064$$

Question: when does k_1 for k_1-BV take effect?

It seems that k_1-BV has *little* impact on the transaction abort rates.

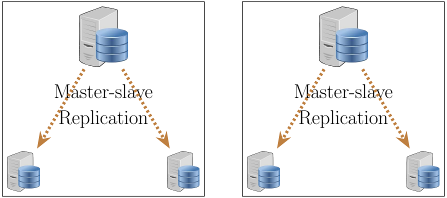
Question: when does k_1 for k_1-BV take effect?

It seems that k_1-BV has *little* impact on the transaction abort rates.

It may be the case in the Aliyun scenarios.

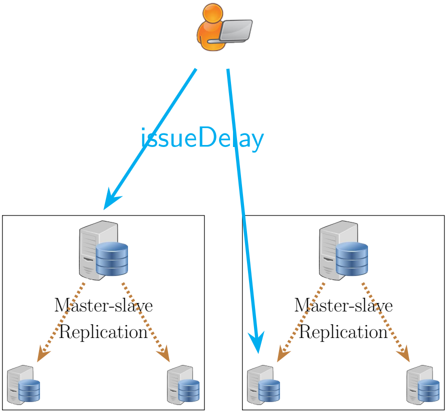
What about other scenarios?

Three types of delays for controlled experiments on local hosts.



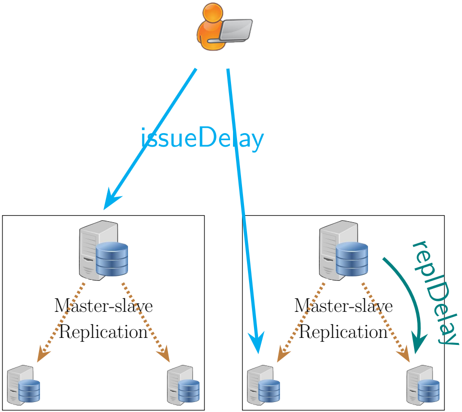
Types	**Values (ms)**	**Explanation**

Three types of delays for controlled experiments on local hosts.



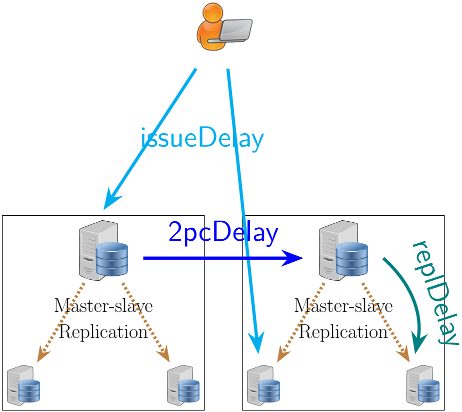
Types	Values (ms)	Explanation
issueDelay	5, 8, 10, 12, 15, 20	delays between clients and replicas

Three types of delays for controlled experiments on local hosts.

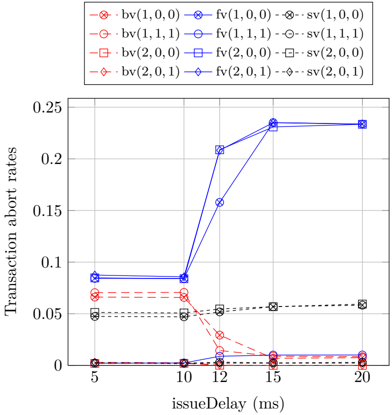


Types	Values (ms)	Explanation
replDelay	5, 10, 15, 20, 30	delays between masters and slaves

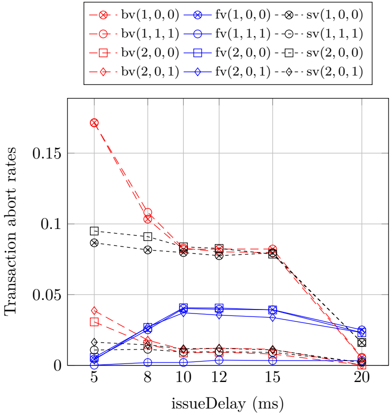
Three types of delays for controlled experiments on local hosts.



Types	Values (ms)	Explanation
2pcDelay	10, 20, 30, 40, 50	delays among masters



(Under read-frequent workloads.)



(Under write-frequent workloads.)

When the "issueDelay" gets shorter,
the impacts of k_1-BV have begun to emerge.

Generally, RVSI *helps to reduce the transaction abort rates* when applications are willing to tolerate certain anomalies.

Generally, RVSI *helps to reduce the transaction abort rates* when applications are willing to tolerate certain anomalies.

k_2-FV: In the Aliyun scenarios, most transactions have been aborted because of violating k_2-FV.

Generally, RVSI *helps to reduce the transaction abort rates* when applications are willing to tolerate certain anomalies.

- k_2-FV: In the Aliyun scenarios, most transactions have been aborted because of violating k_2-FV.
- k_1-BV: In controlled experiments, the impacts of k_1-BV emerge when the "issueDelay" gets shorter.

Generally, RVSI *helps to reduce the transaction abort rates* when applications are willing to tolerate certain anomalies.

- k_2-FV: In the Aliyun scenarios, most transactions have been aborted because of violating k_2-FV.
- k_1-BV: In controlled experiments, the impacts of k_1-BV emerge when the "issueDelay" gets shorter.
- k_3-SV: Complex and challenging (involving multiple data items)