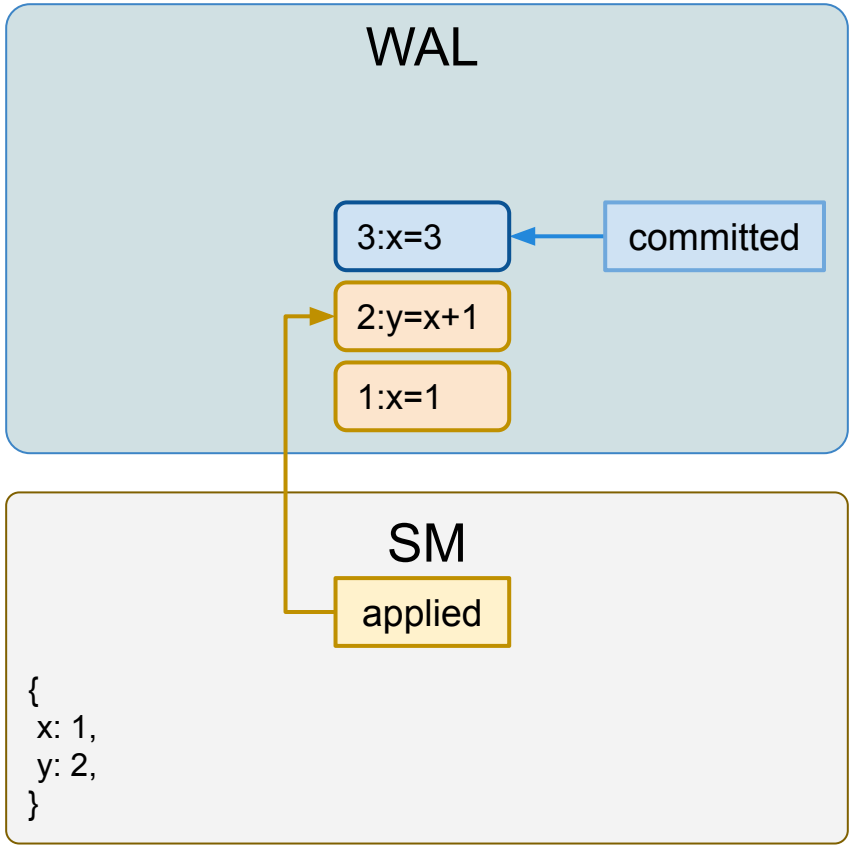
Raft

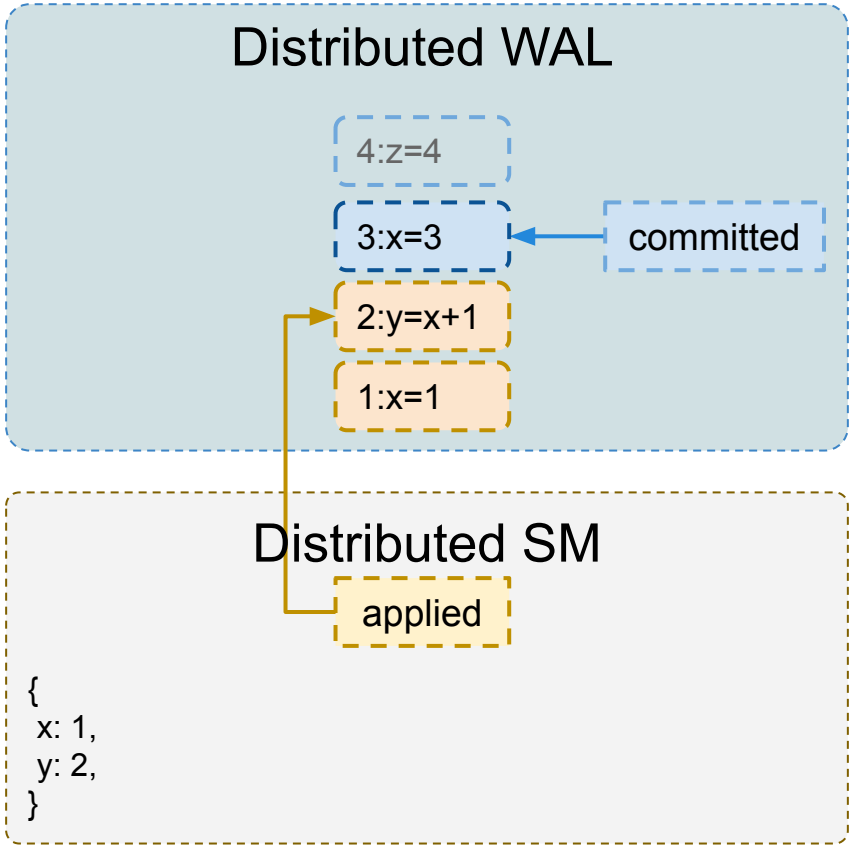
...

2021-06-09 @drdrxp

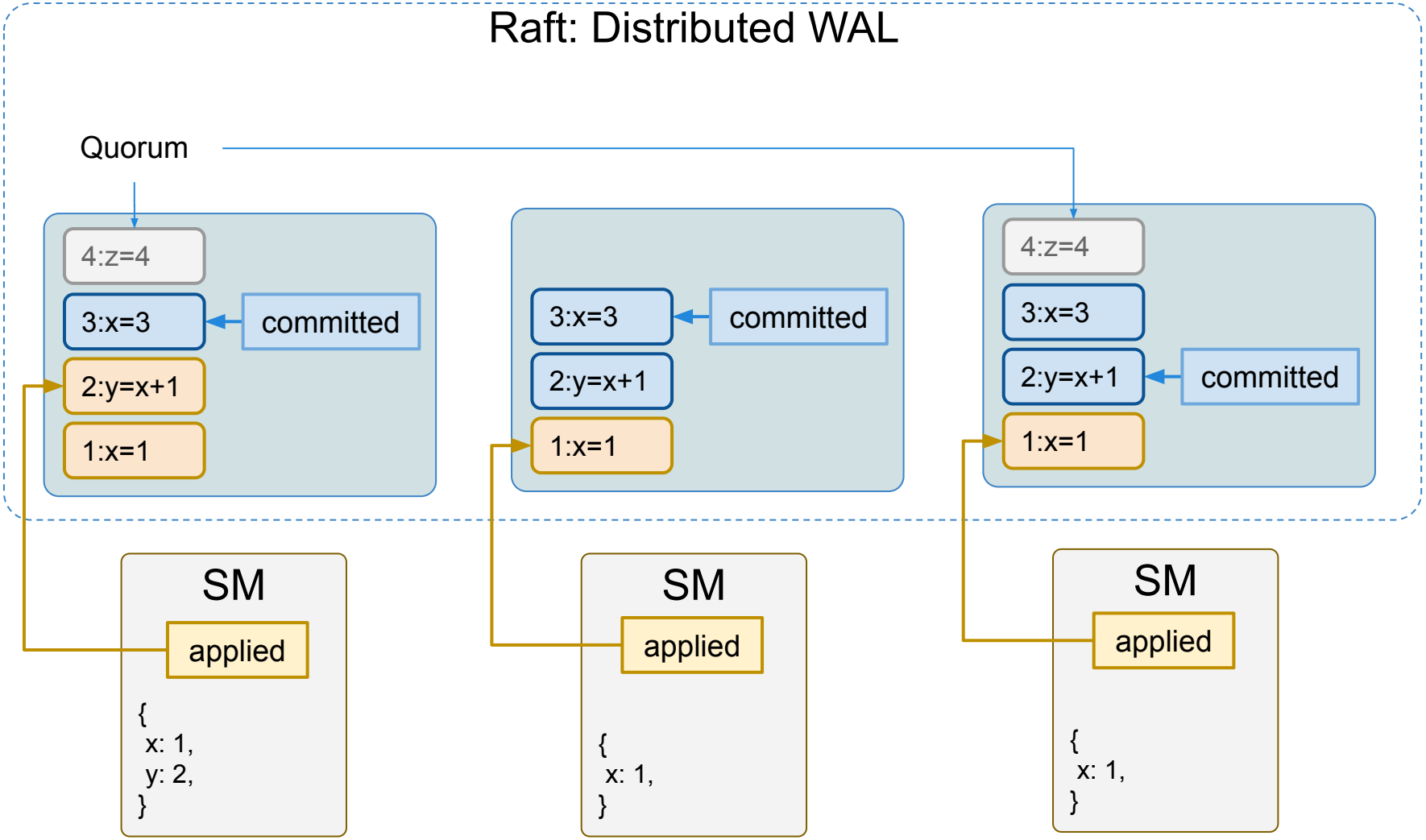
WAL



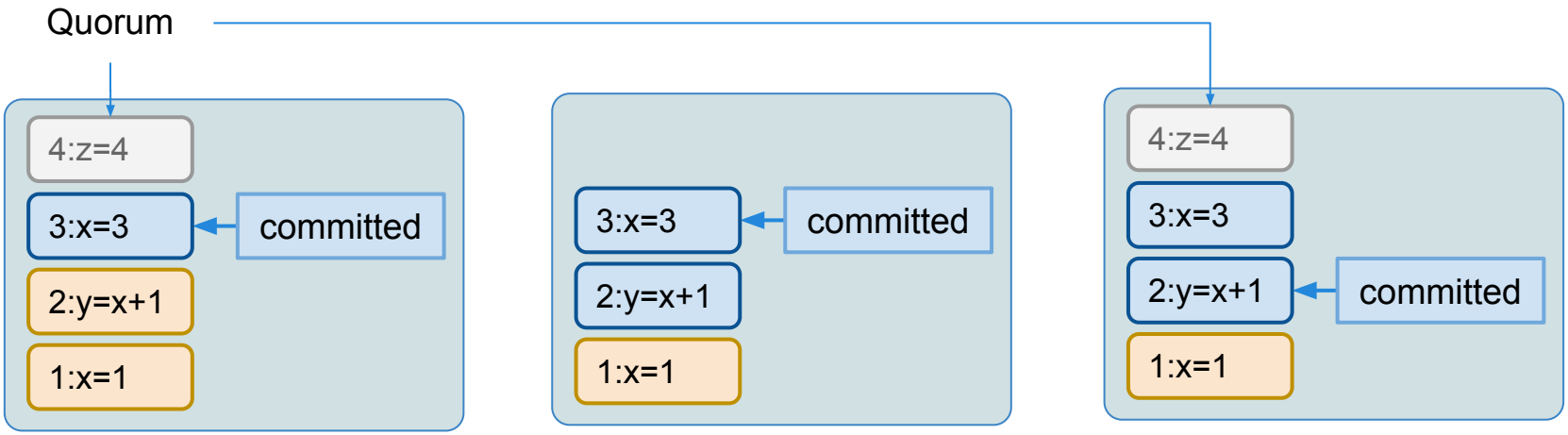
Distributed WAL



Raft


Raft: Distributed WAL
Quorum
4:z=4
3:x=3
committed
2:y=x+1
1:x=1
3:x=3
committed
2:y=x+1
1:x=1
4:z=4
3:x=3
2:y=x+1
committed
1:x=1
SM
applied
{
x: 1,
y: 2,
}
SM
applied
{
x: 1,
}
SM
applied
{
x: 1,
}

Quorum: what and why

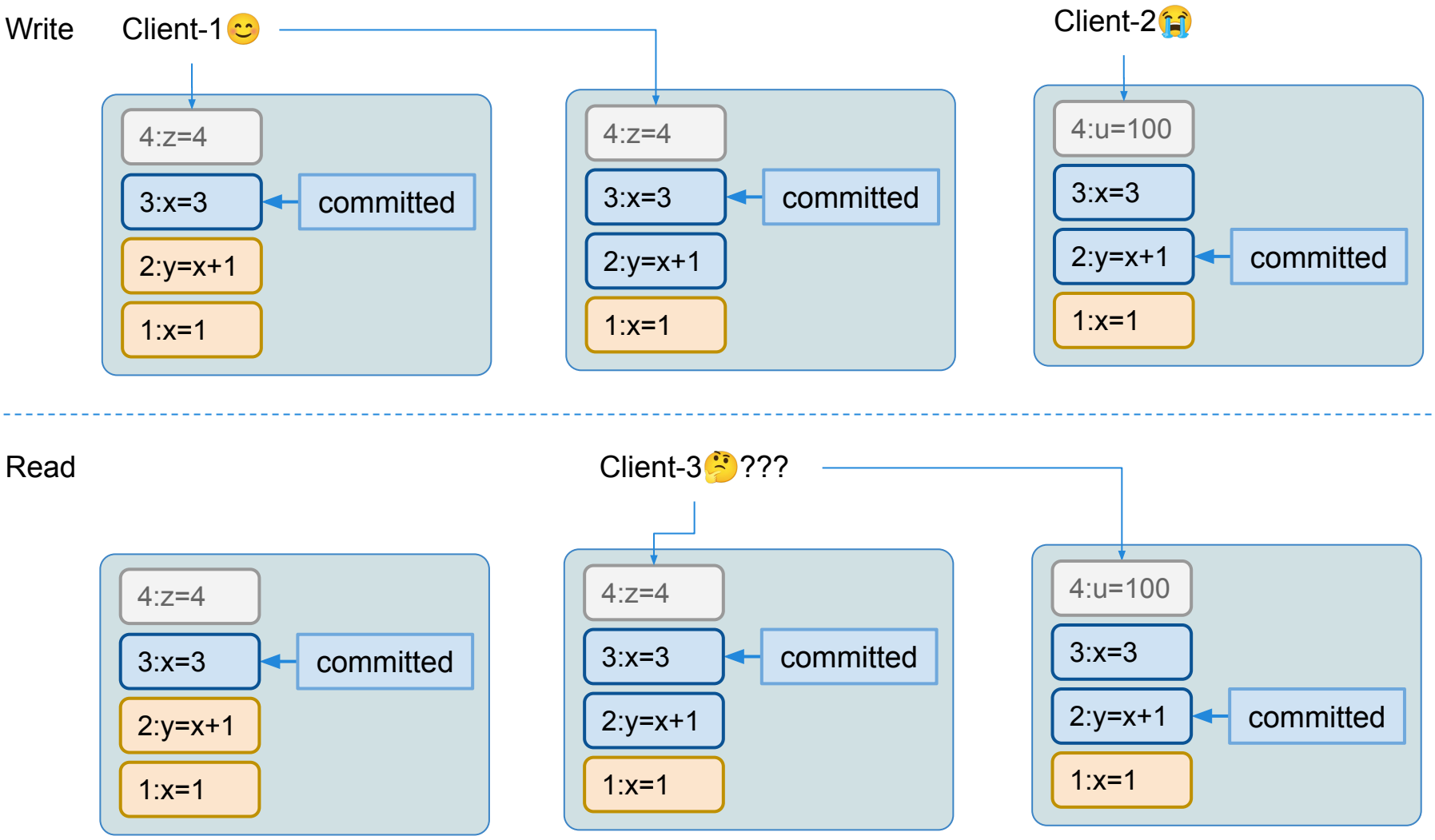


$Q_w \cap Q_r \neq \emptyset$

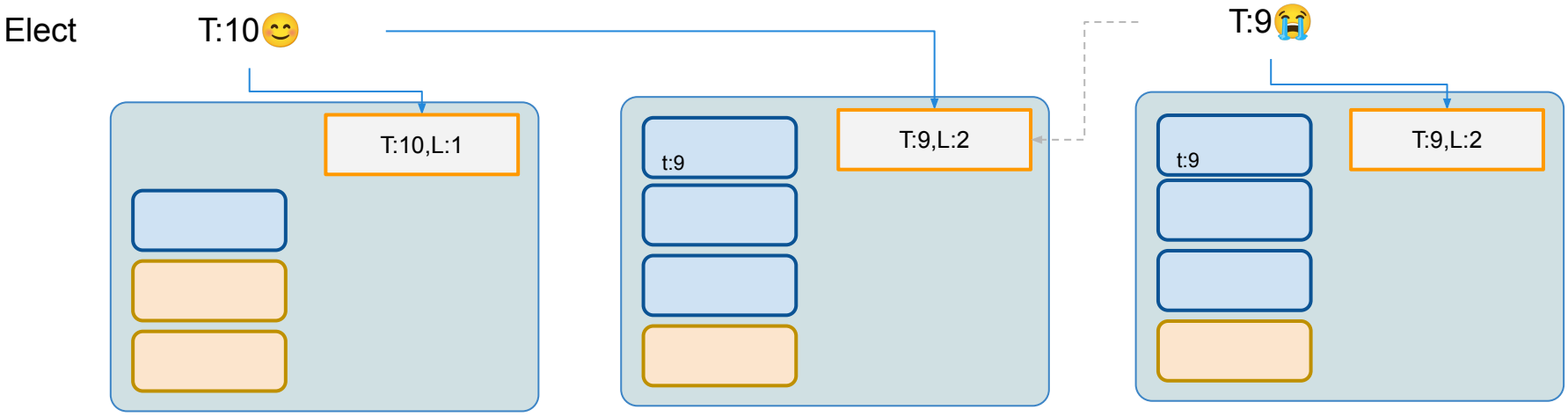
Quorum $\supseteq$ Majority

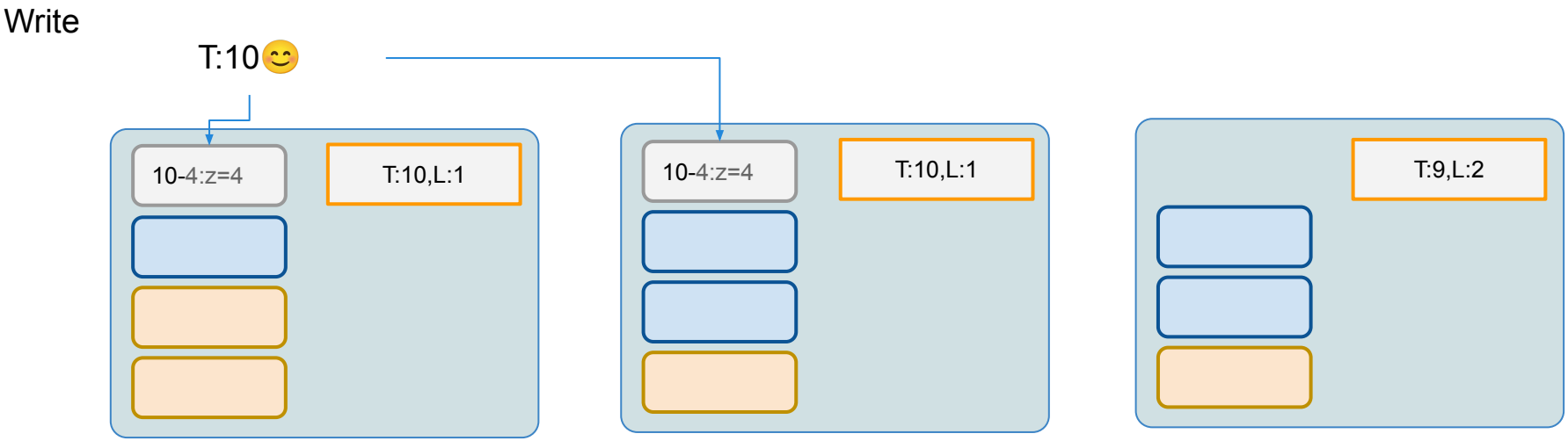
$\mathbb{M}_{abc}$ = {ab, ac, bc}
$\mathbb{M}_{abd}$ = {ab, ad, bd}
$\mathbb{M}_{abcd}$ = {abc, abd, acd, bcd}
$\mathbb{Q}_{abcd}$ = {abc, abd, acd, bcd, ab} = $\mathbb{M}_{abc} \times \mathbb{M}_{abd}$

Term: what and why



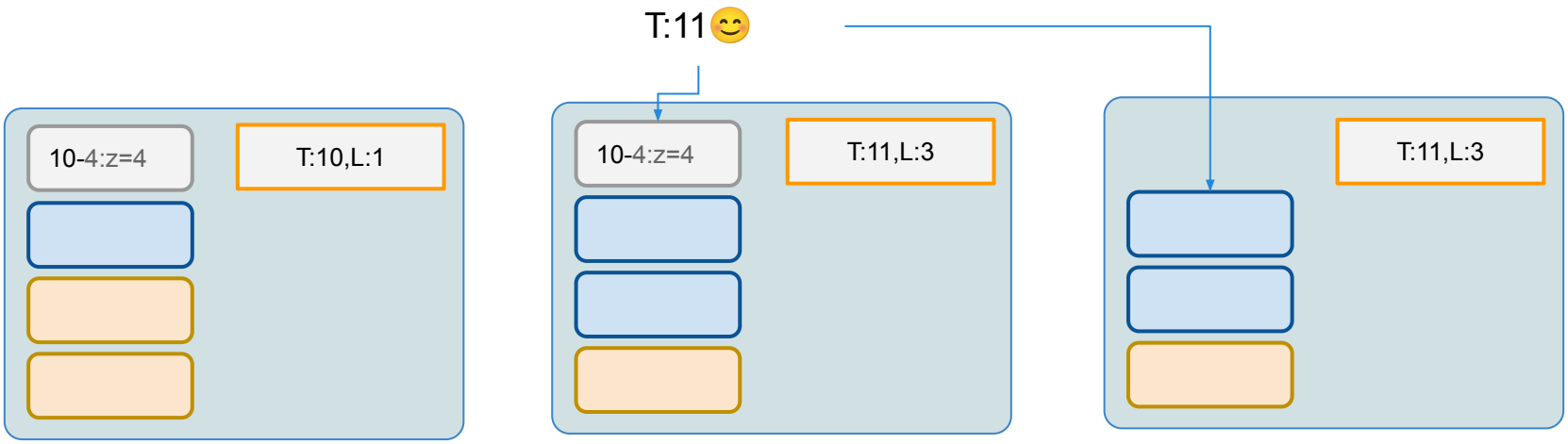
Elect Leader





Elect Leader - 2

Elect

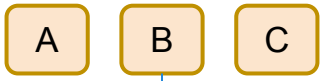


Membership change

要选择一个正确且高效的成员变更算法, 需要满足几个条件. 假设系统要从 Q_1 变更到 Q_2:

- 提交的变更必须可见, 换句话说, 如果系统中有一个已提交的变更, 未提交的变更必须能被识别出来.
- 并发的变更只有一个能成功, 因此多个变更进程必须选择一个相同的 Q 作为提交变更的 quorum 集合. 多个进程共识的数据只有 Q_1, 因此变更必须提交到 Q_1 或 Q_1 的一个确定的扩张.
- 变更必须提交到 Q_2 中的一个 quorum 中.

Membership change - 2



$\mathbb{Q}_0$ = {ab, bc, ac}

$\mathbb{Q}_{01} = \mathbb{Q}_0 \times \mathbb{Q}_1$
{
ab ∪ xy,
ab ∪ yz,
ab ∪ xz,

bc ∪ xy,
bc ∪ yz,
bc ∪ xz,

ac ∪ xy,
ac ∪ yz,
ac ∪ xz,
}

$ab \cap xy = \emptyset$

Joint $\mathbb{Q}_{01}$:

for any $q_i \in \mathbb{Q}_0$, $q_j \in \mathbb{Q}_1$:
$q_i \cap q_j \neq \emptyset$

$\mathbb{Q}_1$ = {xy, yz, xz}
Q1 = {

Q&A

Thanks

drdr.xp@gmail.com
https://blog.openacid.com/

weibo.com: @drdrxp