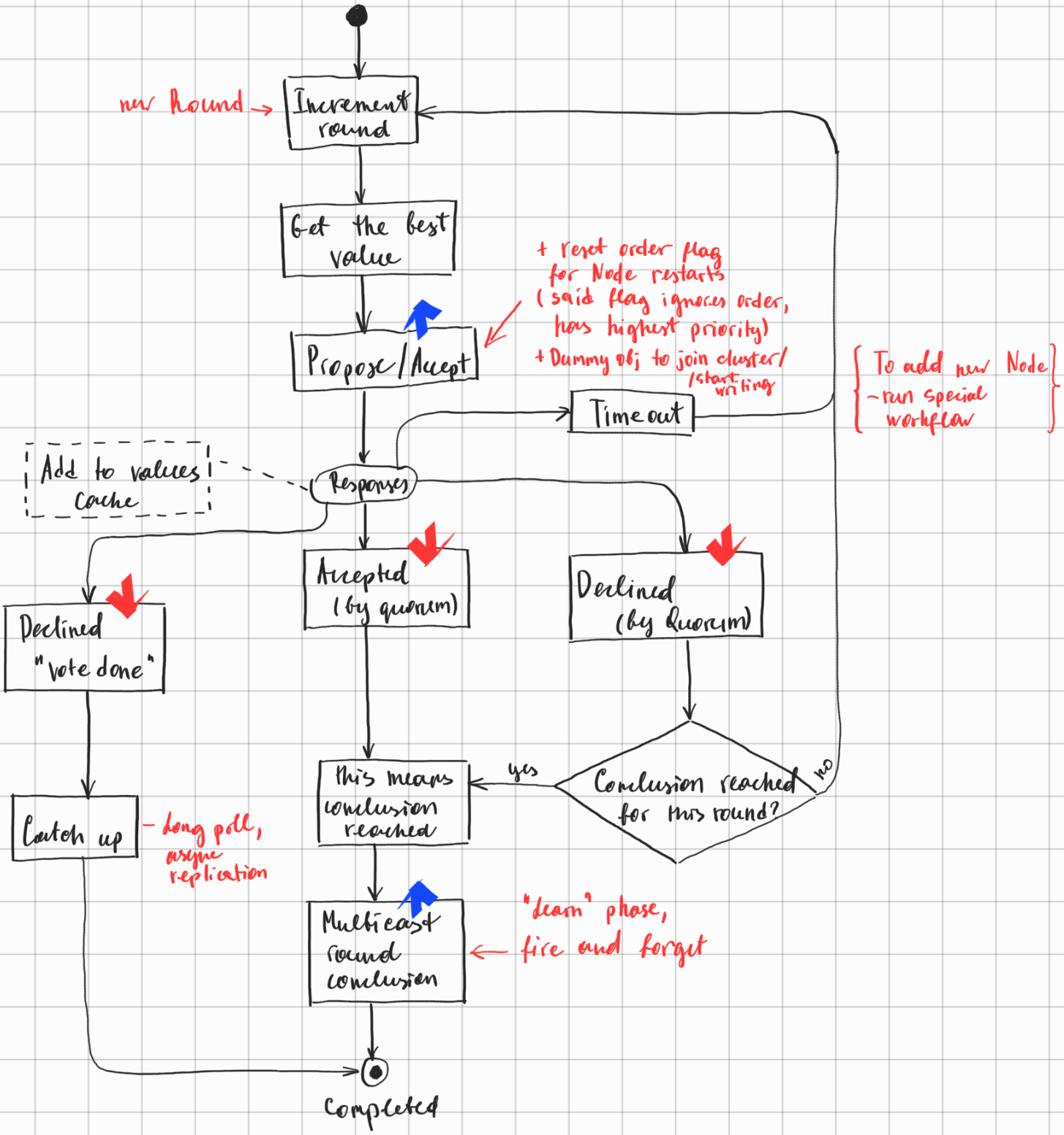
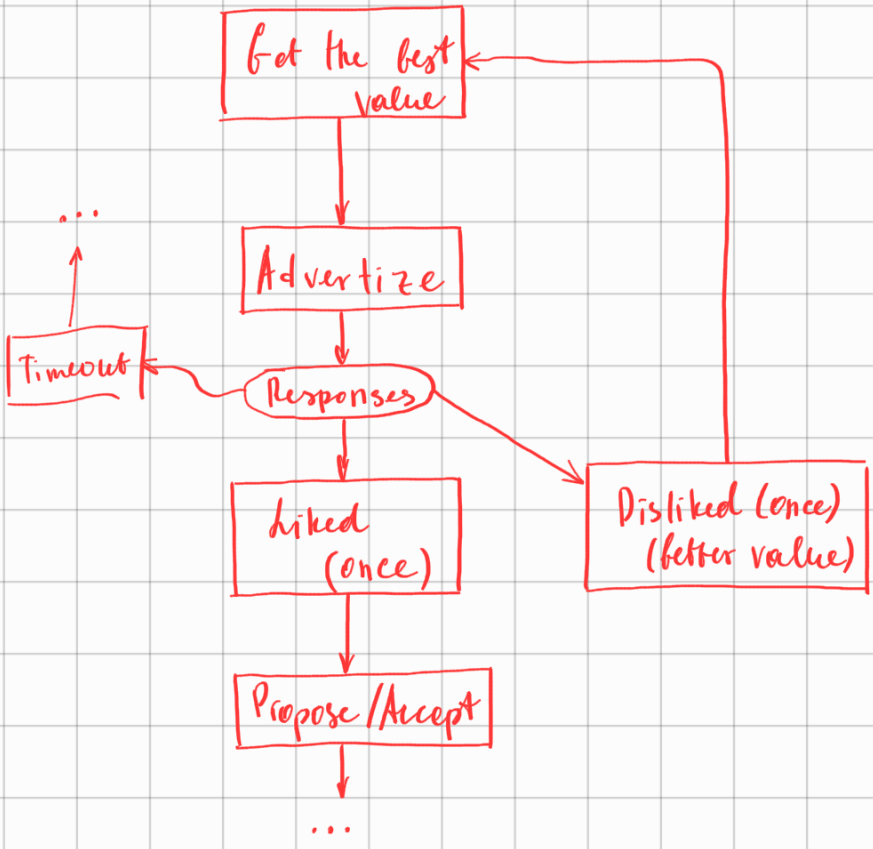

new Round
Increment round
Get the best value
Propose/Accept
+ reset order flag for Node restarts (said flag ignores order, has highest priority)
+ Dummy obj to join cluster/ /start writing
Timeout
{ To add new Node - run special workflow }
Add to values cache
Responses
Accepted (by quorum)
Declined (by Quorum)
Declined "vote done"
Catch up
- long poll, async replication
This means conclusion reached
yes
Conclusion reached for this round?
no
Multicast round conclusion
"learn" phase, fire and forget
Completed

Problem: if we lose $1_{of}^{out}3$ the quorum will timeout whenever conflicts occur.

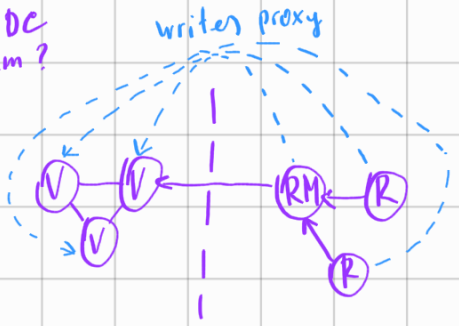
Idea: Advertize shape, as an optimization for small clusters?



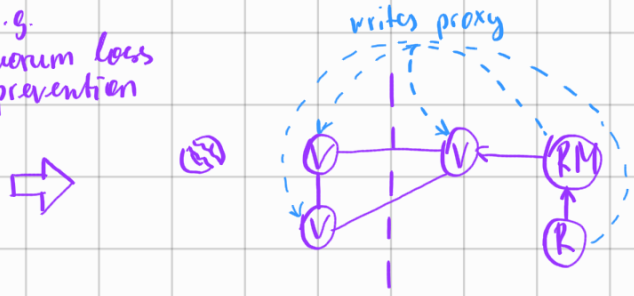
Note: if other Vs have nothing, they'll propose right there, without "Advertize" in their own flow, to minimize overhead on low concurrency traffic.

Idea: can we make it so that only "Advertize" carries the payload, while "Propose" is low-traffic "key" only packet?

Single DC quorum?



E.g. quorum loss prevention



Ideally though, you'd want a cross-DC quorum.
But that's poor write latency, due to cross-DC sync.

It's possible to represent a single range with dual quorum, as below.
But this will effectively double the read load, unless we use a common index.
In this case [N·R·V] tuple should reside on the same machine, sharing the same index.