Groundhog : Additional Analysis

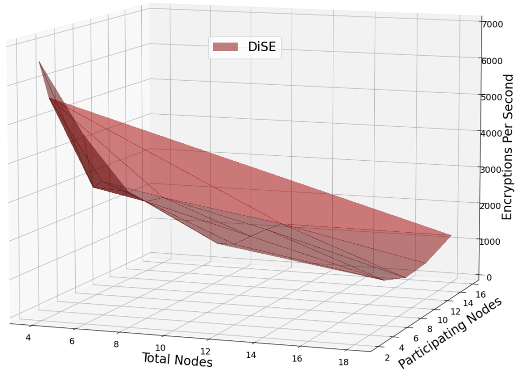

Figure 1: Baseline Performance of DiSE shows that for a given threshold (t), the performance for DiSE drops for larger n, as the initiator waits for a larger number of partial encryption responses.

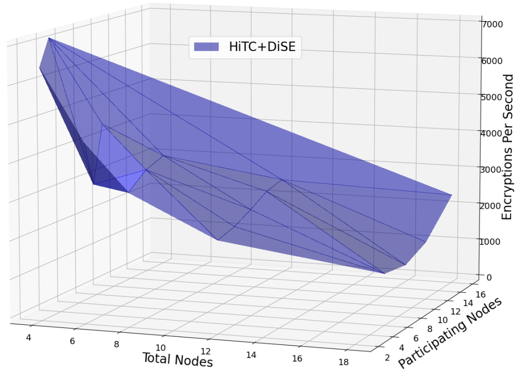

Figure 2: Performance of DiSE using Groundhog showing comparable performance at lower thresholds and better performance at higher thresholds in relation to baseline DiSE.

1. Groundhog on AWS EC2

We also ran Groundhog as docker containers in an AWS EC2 instance and performed additional systems-level profiling. To evaluate overhead of Groundhog, we also obtained baseline TC performance on a cloud environment by adapting running DiSE implementation from Visa [1], without using any reboot-based mechanisms. We ran both these scenarios using upto $n = 18$ docker containers.

We plot the baseline performance in **Fig. 1** and the performance with Groundhog in **Fig. 2**. In both these graphs, the x-axis depicts total number of nodes (n), the y-axis depicts number of nodes participating in the TC application (*i.e.,* threshold t) and the z-axis plots the TC performance in encryption per second. We note that for lower thresholds (*e.g.,* $n = 8$, $t = 2$), the performance in both scenarios are close. However, at higher thresholds (*e.g.,* $n = 8$, $t = 6$), the performance with Groundhog exceeds that of vanilla DiSE. This happens because of higher communication overhead while gathering the encryption responses in DiSE. More specifically, for the case of DiSE, the initiator contacts all the nodes and waits for the first threshold number of responses, whereas in the case of Groundhog, the reboot sequences are known to the initiator via the uptime container. This difference in communication costs is measured and plotted in **Fig. 3** – the upward sloping lines as threshold increases explain the performance variation seen between the case of DiSE and that of Groundhog atop DiSE. Thus, we conclude that with Groundhog, there is no noticeable performance overhead while ensuring availability.

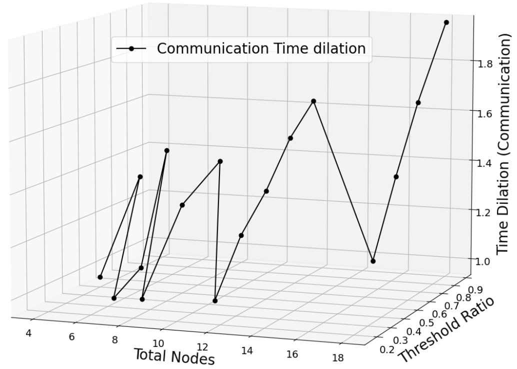

Figure 3: Communication-time differences between baseline DiSE and DiSE with Groundhog shows that for a given n, as the threshold increases, the communication cost overhead increases in vanilla DiSE as the initiator polls all the containers as opposed to with Groundhog, where the reboot sequences and hence the participating nodes are known.

References

[1] Implementation of DiSE: Distributed Symmetric-key Encryption. https://github.com/visa/dise, May 2020.