Rebuttal: Address reviewers suggestions and comments

1. Reviewer 1

1.1. *What is new in this paper from a conceptual perspective?*

First of all, ULFM can only detect process failures. If a node failure occurs the whole job will be revoked and need to be restarted. RDaemon# can detect not only process failures but also node failures.

Secondly, ULFM is using one level of event notification among processes. For RDaemon# we use two steps notification, for the global node level propagation we use **BMG** for reliability broadcast, for process level notification we use **Star Topology** to notify locally.

Also ULFM is limited to Open MPI, but RDaemon# can run

1.2. *Evaluation could use a more in-depth study at large-scale: the authors only focus on the heart-beat overhead for the NERSC machine (32k ranks/PEs).*

Figure 9 shows the comparison between RDaemon# and SWIM, for 4K processes, the stabilization of RDaemon# is still below the range of the heartbeat period and timeout. SWIM latency shows a linear increase when the number of processes increase which will be the bottleneck when scaling up.

Figure 12 displays the failure detection and notification latency comparison of ULFM and RDaemon# with 4K processes hosted on Cori, which demonstrates that scalability trend remains logarithmic with the number of nodes (not processes). It is not easy and costly to allocate larger number of allocation on Cori. Also we are still working on this work, even we could not test with a larger scale on Cori, but we are trying to test on other clusters.

2. Reviewer 2

2.1. *It is not clear for each experiment how many runs where taken and which parameters were used*

The paper provides information about experiment setting in 5.1, which shows "Each experiment is repeated 30 times and we present the average". Also we added detailed description for Figures.

2.2. *why you chose $\delta = 50ms$ for deployment first and then always use 500ms ?*

For the first experiment we intend to test the accuracy and the lower bound of the the heartbeat and timeout. There is no specific reason why we choose 500s for all other. Actually, in real applications failures are much less frequently happened compare to our failure injection rate, which means 500ms is sensitive enough for real application runs.

2.3. ***Showing the overhead of ULFM***

I think we include both RDaemon$^{\#}$ and ULFM overhead in Figure 7 with the same experiments setting for fair comparison.

2.4. ***5.3 Figure 7: You show overheads for*** $\delta = 1ms$ ***and for 10ms, even you state before in section 5.2, that this values leads to false positive results. Would not it be more feasible to show values starting from 20ms increasing?***

Figure 7 shows the overhead without failure injection, the overhead comes from the ring topology of heartbeats sending and receiving. If there is no node failures, we can set heartbeat to a smaller value, this is the reason why we could have 1ms and 10ms results included.

2.5. ***20 milliseconds whereas Figure 8 says lower than 40ms Heartbeat period. Maybe Figure 8 shows the timeout interval*** η***, as also suggested in the text We set a constant ratio*** $\eta = \delta * 2$***. This methodology exposes the behavior in normal deployment (100ms period) ... and in the caption. 5.2 Figure 8*** **RDaemon$^{\#}$** ***daemon failure is not a good label, please choose something which describes the curve better as detection latency***

We regenerated Figure 8 and fix both issues the reviewer suggested.

2.6. ***Figure 9 and 10: A comparison with ULFM would be nice too!***

Figure 9 and 10 are comparison between RDaemon$^{\#}$ and SWIM. If the review means we should compare ULFM and SWIM. Actually we did, but this paper is mainly focus on RDaemon$^{\#}$ instead of ULFM. And Figure 11 and 12 show the comparison between RDaemon$^{\#}$ and ULFM.

2.7. ***Both broadcast algorithms scale the same (by num of nodes or num of processes) as log number of ppn is a constant factor.***

Yes, the scale is the same. But if we increase the scale to a larger scope, the latency increase of ULFM follows the trend of log(num of processes), and the latency increase for RDaemon$^{\#}$ follows log(num of nodes).

2.8. ***5.5 Figure 11 and 12: What is the used heartbeat rate? The graphs indicates a rate below 20ms which you state in 5.2 is leading to false positives. If the detection time is not measured please change the captions and the text accordingly. Otherwise use reasonable heartbeats for both ULFM and*** **RDaemon$^{\#}$.**

Those two Figure show the latency of **process failures** detection and propagation. And heartbeat is used to detect node failures, heartbeat rate doesn't affect the detection of processes failures.

2.9. ***Figure 11: The labels for the log curves do not make sense, maybe just use gray for all of them and one label log x 5.5 Figure 12: Please but the log curve behind the other curves.***

This actually answers question 2.7. We can see log(Processes)_PPN12 line is higher than log(Processes)_PPN1, which demonstrates that ULFM latency trend is affected by the number of processes. If you have the same number of nodes but with different PPN, ULFM may have different latency with different PPN. But for RDaemon$^{\#}$ with the same amount of nodes, no matter how many processes is hosted on each node, the latency is constant and only related to number of nodes.

2.10. ***5.6 Figure 13: What is the difference between RDAEMON and RDEAMON Detected? Both seem to have the same values.***

We regenerated this figure and changed the labels to RDAEMON_detected (node failure is detected by a daemon) and RDAEMON_notified (every process get notified that an error happened.

2.11. ***5.6 in the text to Figure 14 you use $\delta = 0.5s$, whereas in 5.2 you state 0.05s is for normal use. The graph only shows the effect of the detection, as the propagation part is 2 magnitudes faster when looking ag Fig. 11***

Figure 14 shows the latency of single node failure detection and propagation. We changed the description of the figure to make it more clear.

2.12. ***5.6 Figure 15: What is the number of nodes used in this experiment. What is the used heartbeat rate? (0.5s?)***

We change the description of the Figure 15 and added heartbeat value, the number of nodes.

3. Reviewer 3

3.1. ***Grammar/Syntax***

Checked and fixed all the Grammar/Syntax issues.

3.2. ***Consistency***

Checked and fixed. (Alignment of figures and algorithms, enumeration using the same style).