University of Pittsburgh

U.S. DEPARTMENT OF ENERGY | Office of Science

Partial Redundancy in HPC Systems with Non-Uniform Node Reliabilities

Zaeem Hussain, Taieb Znati, Rami Melhem

Department of Computer Science
University of Pittsburgh

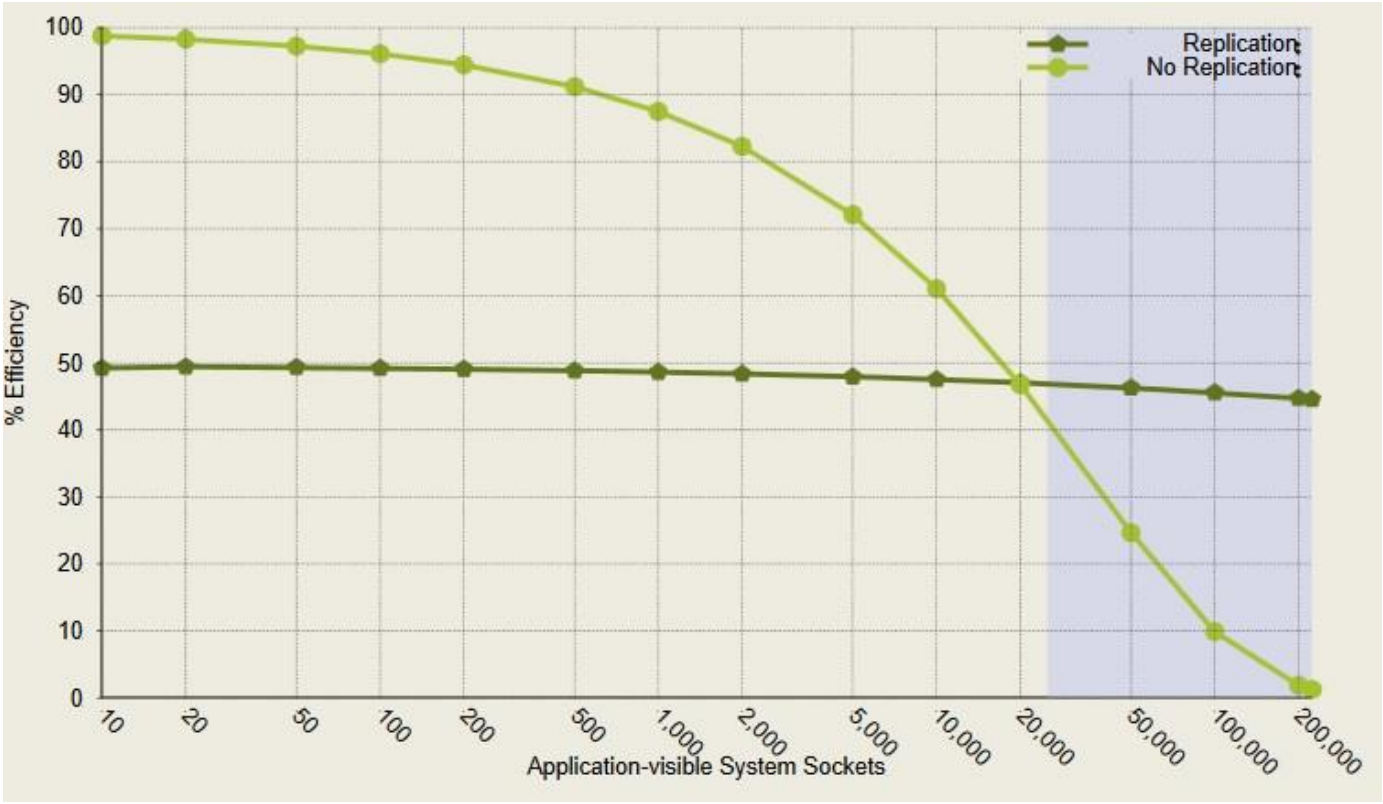
SC'18 Dallas

Fault Tolerance for HPC

- Traditional: Checkpoint/Restart without Replication
  - Upside: All nodes perform useful work most of the time during failure free execution
  - Downside: Every failure causes re-execution of lost work
- Replication (with checkpoints) [1]
  - Downside: Duplicated work on half the system nodes
  - Upside: Most failures do not trigger a rollback, requiring much fewer checkpoints

[1] Ferreira, Kurt, et al. "Evaluating the viability of process replication reliability for exascale systems." *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011.

To replicate or not to replicate?



Ferreira, Kurt, et al. "Evaluating the viability of process replication reliability for exascale systems." *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011.

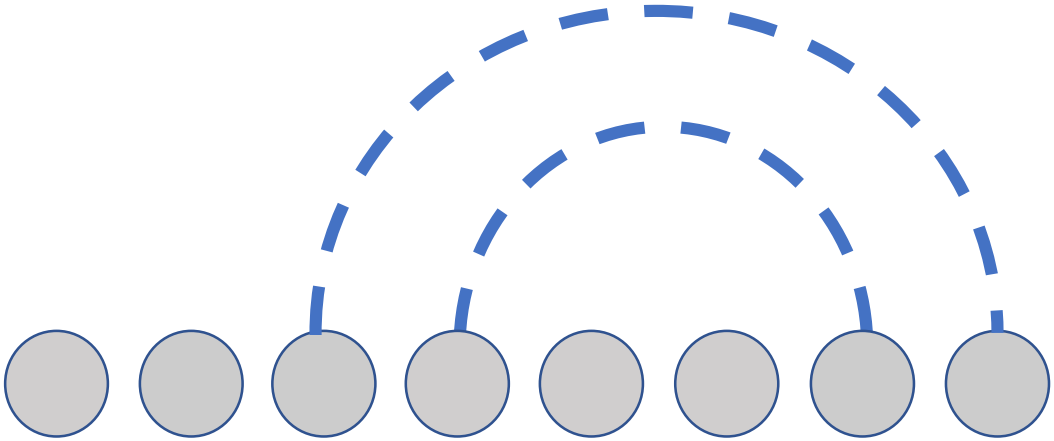
Is there something in between?

Is there something in between?

- Answer: Partial Replication [2]

System nodes

Pairing of node and its replica



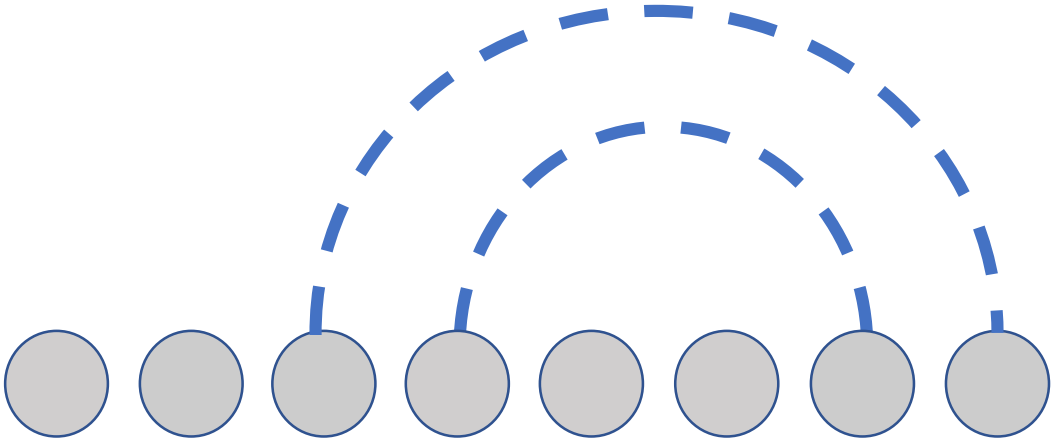
[2] Elliott, James, et al. "Combining partial redundancy and checkpointing for HPC." *2012 IEEE 32nd International Conference on Distributed Computing Systems*. IEEE, 2012.

Is there something in between?

- Answer: Partial Replication

System nodes

Pairing of node and its replica



Replication Factor, $r = \frac{\#\ Total\ nodes}{\#\ Application\ visible\ nodes} = \frac{8}{6} = 1.33$

But does partial replication pay off? [3]

- Conclusion of [2] based on simulation results:

  '… while partial replication may "pay off" (yield a shorter runtime than perfect strong scaling), the best overall value is offered by full replication.'

- Our findings:
  - MTTI with exponentially distributed failures approximately proportional to $1/(2-r)$
  - Minimum of normalized expected completion time using Young's[4] formula occurs at one of the extremes, i.e. $r = 1$ or $r = 2$
  - Similar numerical findings for Weibull distribution

[3] Stearley, Jon, et al. "Does partial replication pay off?." *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*. IEEE, 2012.

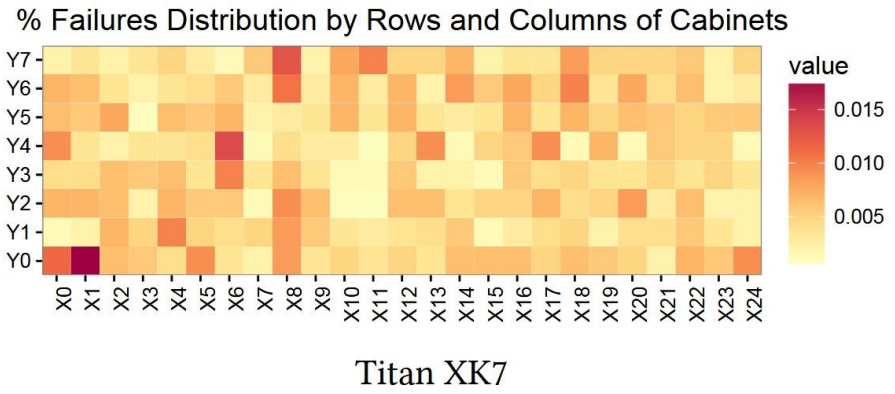
[4] Young, John W. "A first order approximation to the optimum checkpoint interval." *Communications of the ACM* 17.9 (1974): 530-531.

The IID assumption

- Assumption underlying the analysis so far:
  - All nodes in the system have the same individual likelihood of failure.
- Nearly universal assumption in all theoretical work on fault tolerance for HPC
- But is it true in practice?

Failures in HPC Systems

- "The spatial distribution of failures is not uniform at any compute granularity across systems."[4]



- Similar conclusions drawn by other studies on failures in Supercomputers.

[4] Gupta, Saurabh, et al. "Failures in large scale systems: long-term measurement, analysis, and implications." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2017.

Consequences of assuming Non-Uniform Failure Likelihoods

- Which nodes to replicate?
- How to form replica groups?
- Does partial replication pay off in such a setting?

Our Contribution

We use theoretical and numerical analyses to answer these questions for $1 \leq r \leq 2$.

Optimally Reliable Configuration

Our Theoretical Result

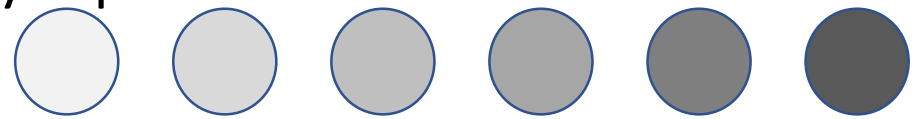
Given N independent nodes with different failure likelihoods, and a fixed r (i.e. knowing how many nodes should be replicated), system reliability is maximized when:

1. The least reliable nodes are replicated among themselves
2. Within the nodes to be replicated, the least reliable node is paired with the most reliable node, the next least reliable with the next most reliable, and so on.

Example

$N = 6$, $r = 1.5$

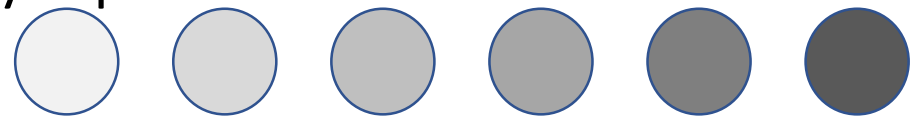
Color intensity represents node failure likelihood



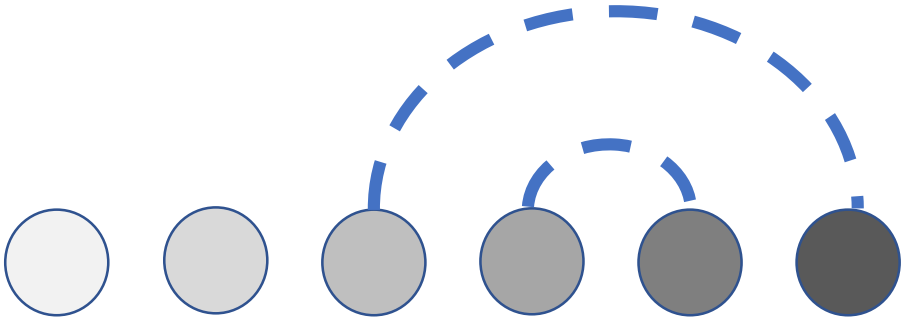
Example

$N = 6$, $r = 1.5$

Color intensity represents node failure likelihood



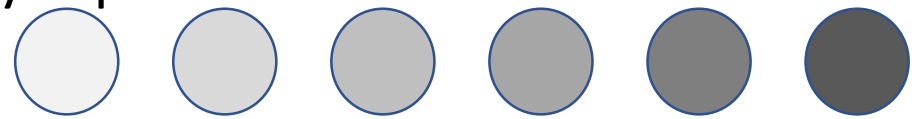
Configuration with maximum reliability:



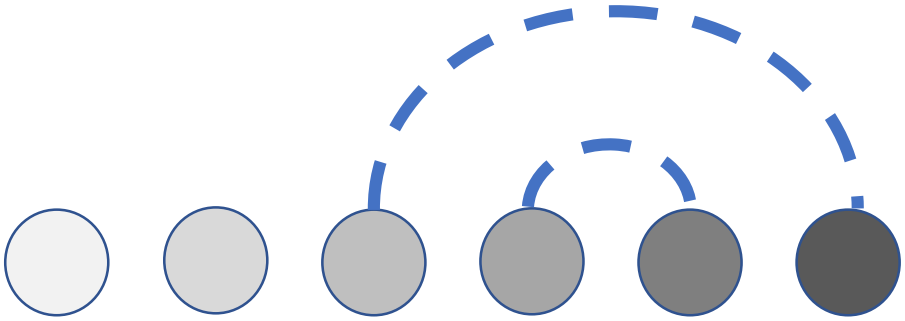
Example

$N = 6$, $r = 1.5$

Color intensity represents node failure likelihood



Configuration with maximum reliability:



Observation: We need not know the exact values of the failure likelihoods of the nodes

- Only the relative ordering of nodes based on their likelihood of failing is enough

Feasibility of Partial Replication

- The earlier result only tells which nodes to replicate, *knowing how many to replicate*
- We can evaluate the best r by comparing $E_N(r)$, the normalized expected completion time (i.e. expected time using partial replication r to finish work that takes unit time on N nodes).

Feasibility of Partial Replication

- The earlier result only tells which nodes to replicate, *knowing how many to replicate*
- We can evaluate the best r by comparing $E_N(r)$, the normalized expected completion time (i.e. expected time using partial replication r to finish work that takes unit time on N nodes).
- Optimization problem

$$\min_{a,b} E_N(r)$$

$$\text{subject to} \quad a + 2b \leq N$$

$$n = a + b \geq 1$$

where $r = (a + 2b)/(a + b)$

$a =$ number of non-replicated nodes

$b =$ number of replicated pairs

Toy Examples

- 8 nodes with exponential failure rates
  $C = 0.01, \alpha = \gamma = 0$

Example 1: Failure rates $= [0.9, 1.0, 1.1, 1.2, 4.9, 5.0, 5.1, 5.2]$

Example 2: Failure rates $= [0.9, 1.0, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2]$

Toy Examples

- 8 nodes with exponential failure rates
  $C = 0.01, \alpha = \gamma = 0$

Example 1: Failure rates $= [0.9, 1.0, 1.1, 1.2, 4.9, 5.0, 5.1, 5.2]$

Answer: Optimal $r = 8/6$

$[0.9, 1.0, 1.1, 1.2, 4.9, 5.0, 5.1, 5.2]$

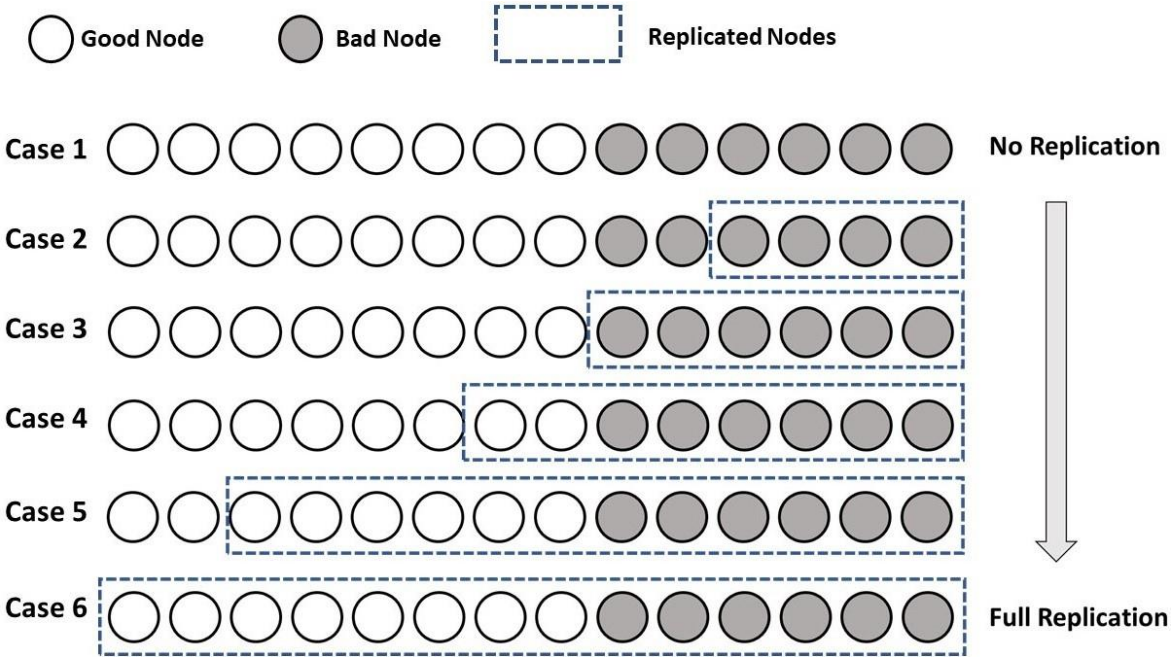
Example 2: Failure rates $= [0.9, 1.0, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2]$

Answer: Optimal $r = 8/5$

$[0.9, 1.0, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2]$

Systems with Two Kinds of Nodes

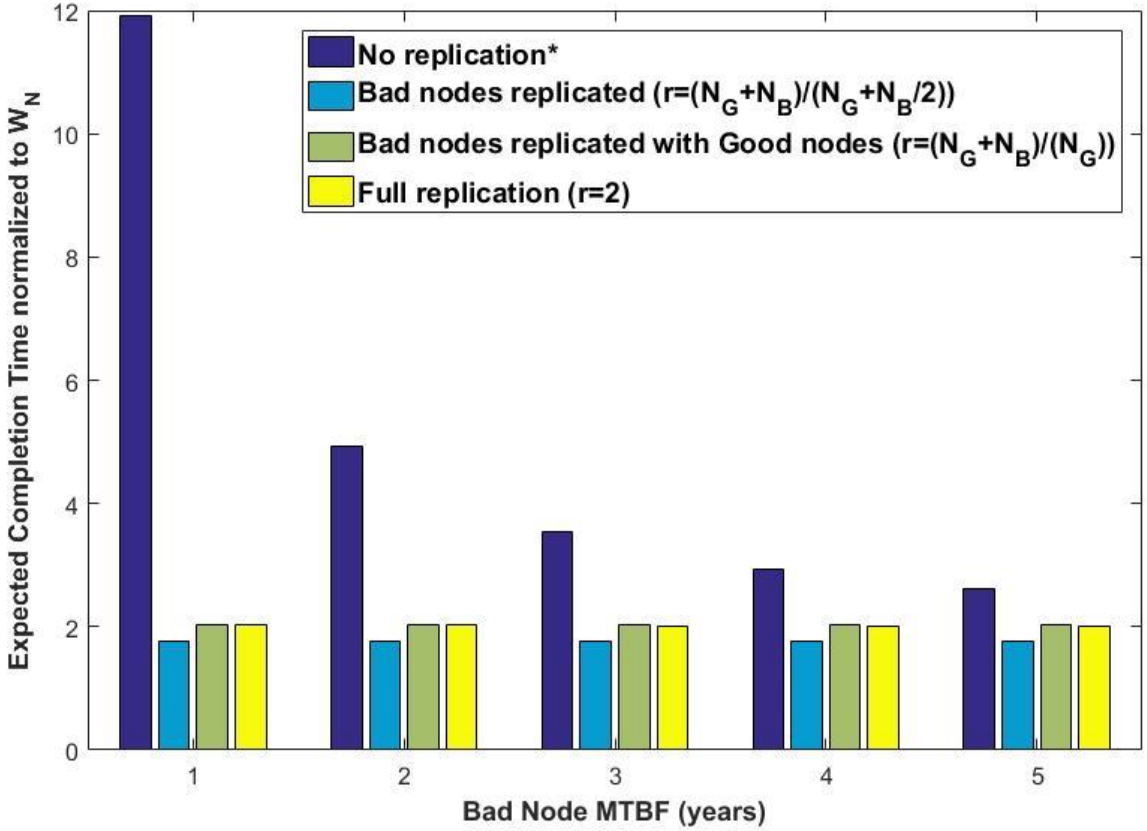
- Possible cases when using all nodes in the system



- Approximate analysis for Case 2 similar to IID scenario
- Optimal completion time usually achieved by the boundary cases only

Results

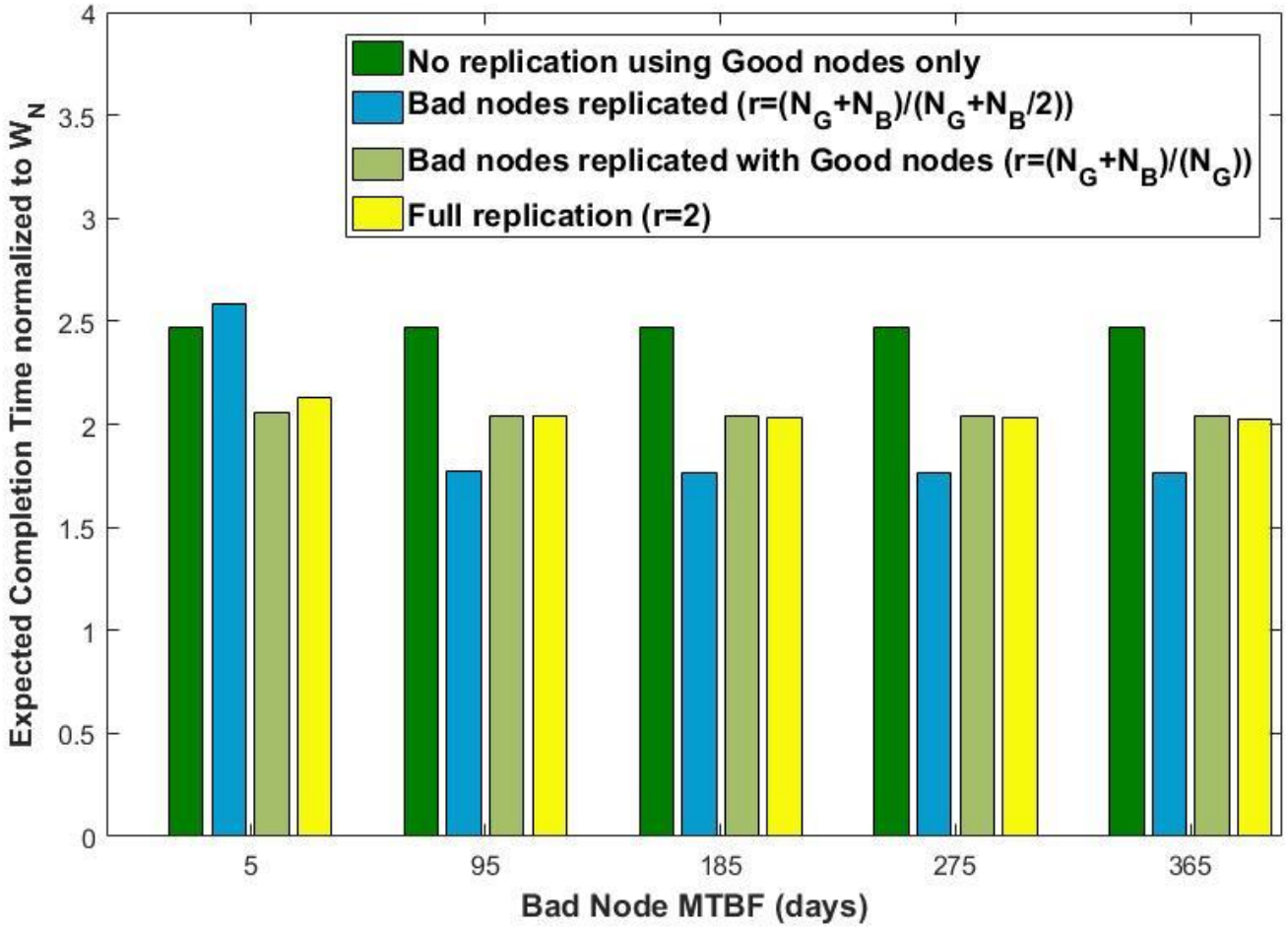
- Replicating all bad nodes among themselves usually sufficient



Good Nodes MTBF: 50 years, C = 60 seconds, $\alpha = \gamma = 0$

Results

- What about not using the bad nodes at all?



Takeaways for Such Systems

- Minimum expected completion time usually achieved by either
  - No replication at all, or
  - Replicating all bad node among themselves
- For realistic values of node MTBFs, utilizing bad nodes (via replication) achieves faster average completion times when compared to not using the bad nodes at all.

Conclusions

- Provided theoretically optimal way of arranging system nodes into non-replicated nodes and replica groups.
- Demonstrated that partial replication can provide optimal performance when node failure distributions are not identical.
- Theoretical result and problem formulation are general enough to handle systems with arbitrary kinds of nodes

Acknowledgements

US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research

Center for Research Computing, University of Pittsburgh