Compute System Simulator:
Modeling the Impact of Allocation Policy and Hardware Reliability on HPC Cloud Resource Utilization

Jarrod Leddy, Huseyin Yildiz
jarrodleddy@microsoft.com
Microsoft Corporation

Microsoft Azure

SC25 St. Louis, MO | hpc ignites.

1. Introduction

Motivation:
- As cloud computing clusters become increasingly complex and geared towards HPC, fragility of the systems can lead to lower than optimal compute utilization
- Choice of how to distribute workloads accounting for hardware outages significantly impacts compute utilization outcomes

Method:
- Discrete timestep simulation of resource allocation/deallocation and hardware failures
  - Probabilistic hardware failures
  - Multiple allocation policies
  - Workload checkpointing
  - Diagnostics for detailed analysis

We present a simulation tool developed at Microsoft that can be used to optimize utilization of cloud infrastructure and resource management, accounting for infrastructure reliability.

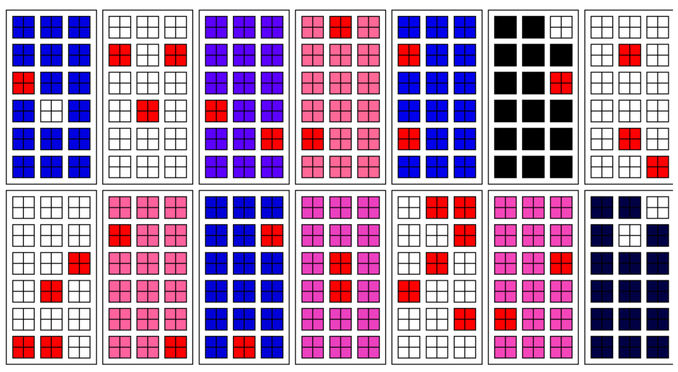
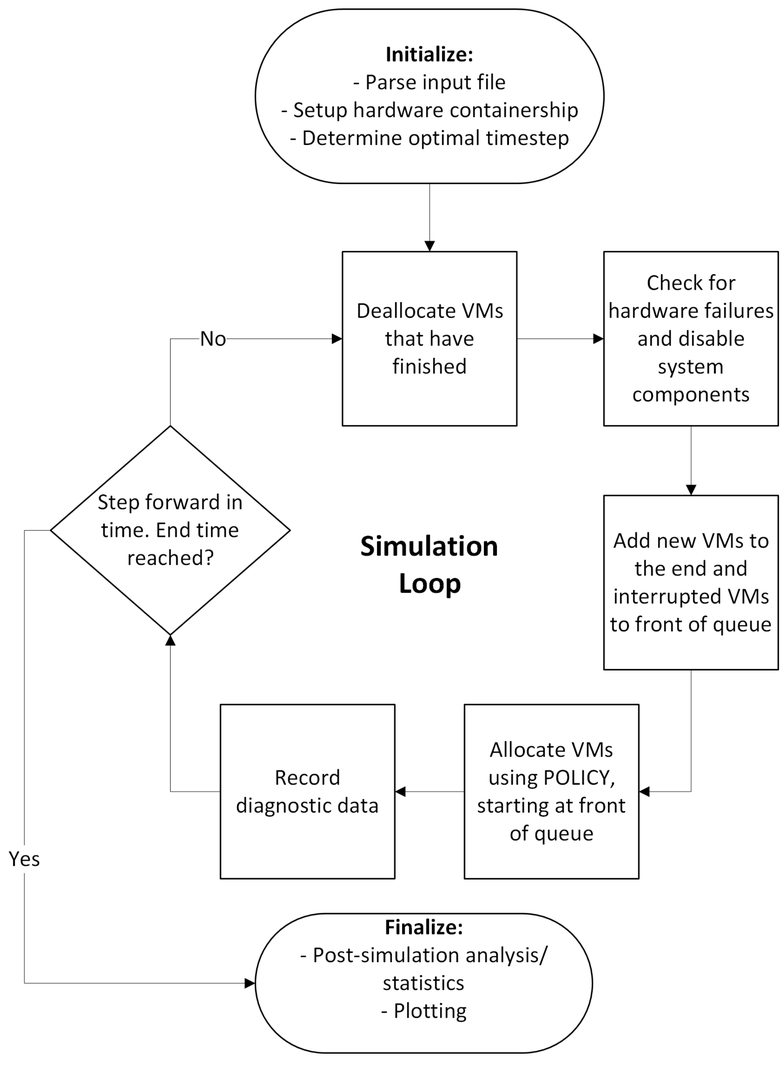


Fig 1. Workload allocations (blue, pink, black) and hardware failures (red) preventing allocations on racks where the workload size no longer fits.

2. Simulator Design

Simulator is explicit, time-domain with probabilistic hardware failures/repairs
- Hardware Components - know their health status and workload, contain lower hardware objects
- Scheduler - Assigns workloads to hardware based on allocation policy (available policies: single, whole, chunked, adjacent)
- Failure Events – sampled from Weibull or fixed distributions. Each hardware component type can have different failure frequency



3. GB200 NVL72 Experiments

GB200 NVL72
System:
- Cutting edge CPU/GPU from Nvidia
- 4 GPUs per node, 18 nodes per rack, 28 racks per cluster
- High bandwidth NVLink connections allow entire rack to function as an effective-single node

Failure modes:
- Failure of any one component leads to rack failure, workload restart required
- Common fatal failures:
  - NVLink connectivity
  - GPU failure/missing
  - Xid 145/149
  - NVMe failures
  - Thermal failures/leaks
- Common non-fatal failures:
  - NVLink/IB bandwidth degradation
  - IB/NVLink Flaps
  - ECC spikes

Workload decisions:
- GPUs in NVLink pod for workload
- Checkpoint frequency
- Whole/split rack allocations
- Multi-priority workloads

Simulations:
- Perform many simulations filling out parameter space:
  - mean-time-to-failure (MTTF) (100-1600 days)
  - mean-time-to-repair (MTTR) (1-14 days)
  - workload size (40-72 GPUs)
- Fixed:
  - allocation policy (single rack)
  - workload duration (2 weeks)
  - checkpoint frequency (12hr)
  - total duration (2 years)

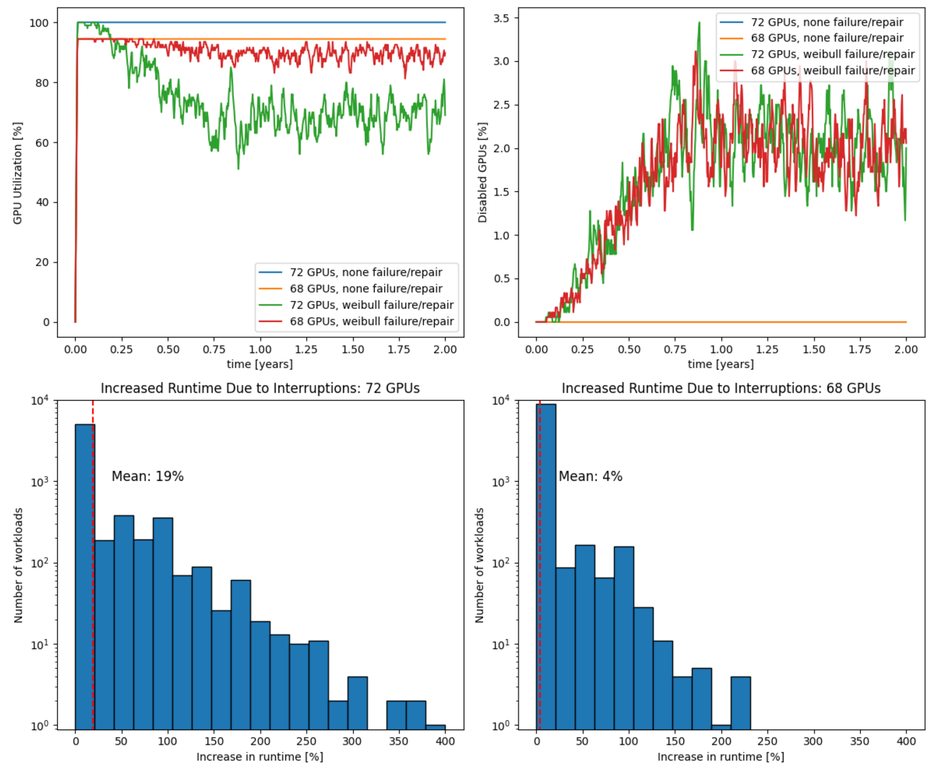


Fig 2. GPU utilization (top left), availability (top right), and increased runtime distributions for 72 (left) and 68 (right) GPU NVLink domains. Mean workload runtime increase improves from 19% to 4% when using the smaller NVLink domain.

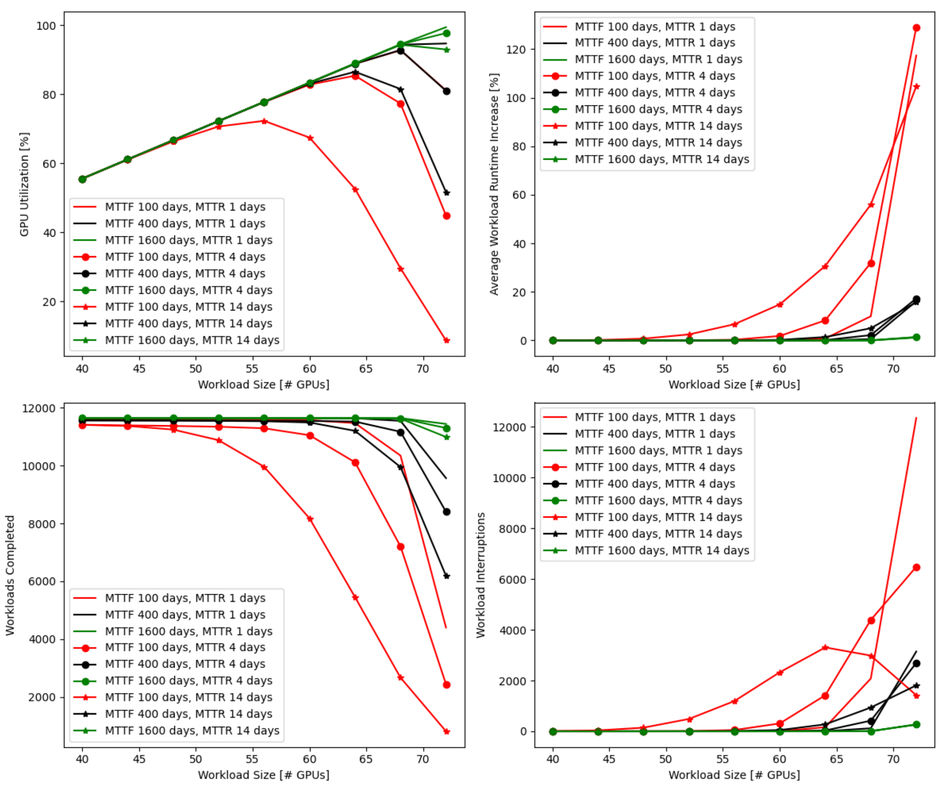
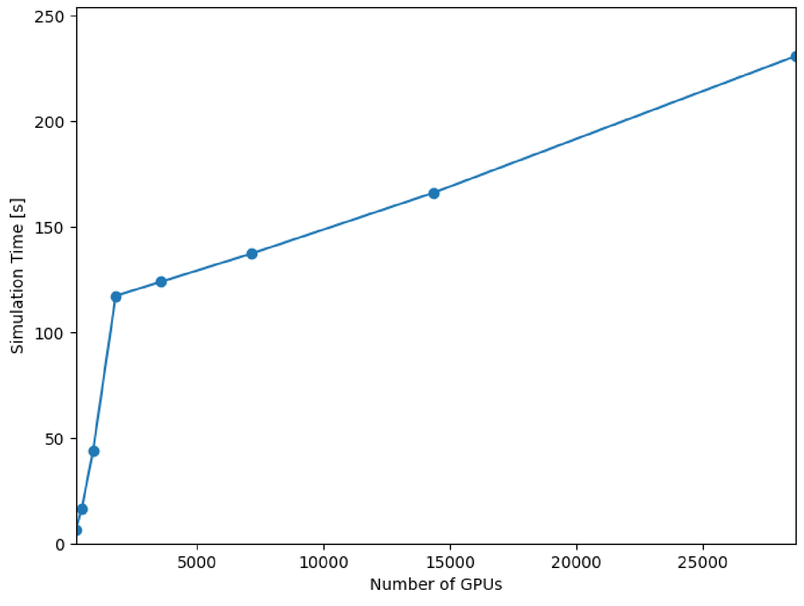


Fig 3. Comparison of GPU utilization (top left), runtime increase (top right), completed workloads (bottom left), and workload interruptions (bottom right) for various values of MTTF and MTTR.

4. Scalability

What happens if we have 1000 clusters we want to analyze, how does the simulator scale?
- Simulator scales with the number of compute components – in the case of the GB200 NVL72 scenario above this would be GPUs
- Time-step automatically calculated to be as large as possible to resolve highest frequency events, so number of steps varies for fixed total time
- Where hardware failure/repair period << VM duration, simulator scales as *NlogN*, otherwise N^2



5. Conclusions

- Choosing an allocation policy is a challenging prospect with significant compute utilization and financial ramifications
- Many system parameters (often out of the control of the user) can also have significant impact on the utilization (e.g. hardware failure/repair rates)
- Hardware failures can effectively increase workload run-time due to both stoppage time and checkpoint reversion
- A tool, such as the one developed here, can provide data toward making informed decisions on workload size, allocation policy, distribution, and prioritization – maximizing resource utilization
- Scalability limited by component search during reallocation in event of hardware failure. Improvements can be made by optimizing data structure organization/lookup.