Resilience Structural Design Pattern Modeling

Mohit Kumar, and Christian Engelmann – Oak Ridge National Laboratory

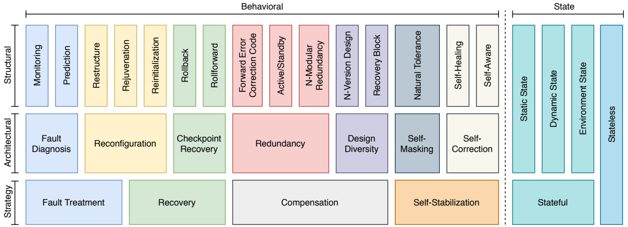
Motivation

Resilience in extreme-scale high-performance computing (HPC) systems is a critical challenge.

- High component counts
- Lower component reliability
- Hardware complexity
- Software complexity

Resilience Design Patterns

Design patterns describe generalizable solutions to recurring problems. Resilience design patterns address the issues of dealing with faults, errors, and failures in extreme-scale HPC.



Terminology and Metrics

- Fault is a defect in a system that has the potential to cause an error.
- A fault becomes an error when it is activated and results in an illegal system state.
- A failure occurs when an error reaches the service interface of a system, resulting in system inconsistent behavior with its specification.
- Reliability is the probability of a system not experiencing a fault, error, or failure during operation.

$$R(t) = 1 - F(t) = \int_t^{\infty} f(t)d(t)$$

- ➢ λ is the frequency at which a system experiences fault, error or failure.

$$MTTF = \int_0^{\infty} R(t)d(t) = 1/\lambda$$

- ➢ λ displays the “bathtub curve” which results in a normalized exponential probability density function (PDF).

$$R(t) = e^{-\lambda t}$$

- ➢ N systems depending on each other exhibit serial reliability and N systems redundant to each other have parallel reliability.

$$R(t)_s = \prod_{n=1}^{N} R_n(t)\,, R(t)_p = 1 - \prod_{n=1}^{N} (1 - Rn(t))$$

- Availability is the proportion of time a system provides a correct service, with planned uptime (PU) t_{pu}, scheduled downtime (SD) t_{sd} and unscheduled downtime (UD) t_{ud}.

$$A = \frac{t_{pu}}{t_{pu} + tsd + tud} = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

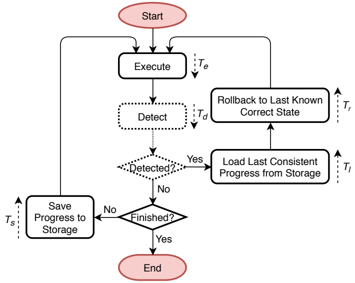
- ➢ Serial and parallel availability can be defined as:

$$A_s = \prod_{n=1}^{N} A_n, Ap = 1 - \prod_{n=1}^{N} (1 - An)$$

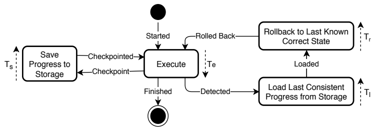
- Performance is the time required to successfully execute a task, including PU, SD and UD.

Rollback Resilience Design Pattern

- A derivative of the Checkpoint Recovery architectural pattern which supports resilient operation by restoring the system to a known correct state in the event of an error or failure.



Rollback pattern flowchart



Rollback pattern state diagram

- Parameters
  - T_e - Time to execute system progress
  - T_d - Time to detect an error/failure
  - T_l - Time to load consistent system state and progress from storage
  - T_r - Time to rollback to the last known correct state
  - T_s - Time to save system state and progress to storage

- Performance

$$T = Me^{(Tl+Tr)/M}\left(e^{(\tau+Ts)/M} - 1\right)\frac{T_E}{\tau}$$

$$\tau = \sqrt{2MTs}\left[1 + \frac{1}{3}\left(\frac{T_s}{2M}\right)^{1/2} + \frac{1}{9}\left(\frac{T_s}{2M}\right)\right] - Ts$$

N-Modular Resilience Design Pattern

- A derivative of the Redundancy architectural pattern enables the continuous correct operation of a system by applying redundancy to system state and optionally to system resources.
- Parameters
  - T_a - Time to activate N replicas of the system
  - T_i - Time to replicate the input to the N replicas

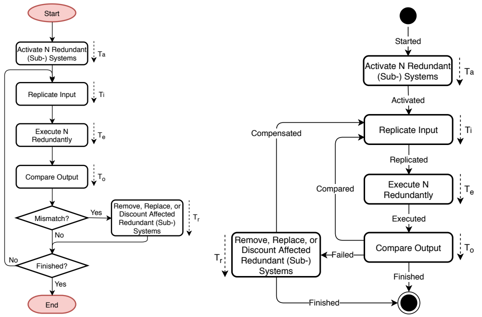
Resilience Structural Design Pattern Modeling

Mohit Kumar and Christian Engelmann – Oak Ridge National Laboratory

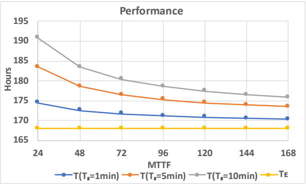
- T_e - Time to execute system progress in the N replicas
- T_o - Time to compare the outputs from the N replicas
- T_r - Time to remove, replace, or discount the affected replica(s)

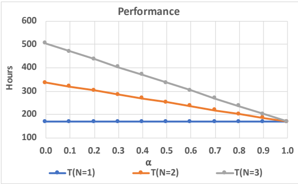
• Performance

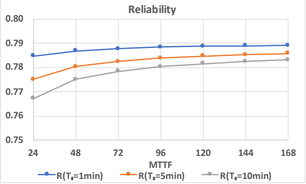
$$T = \alpha TE + (1-\alpha)NT_E + P(t_i + to) + TR$$

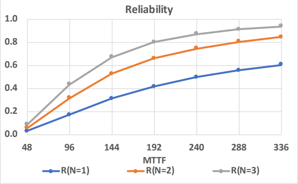


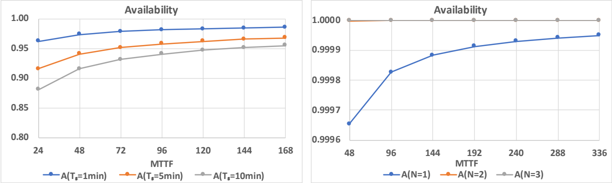
N-modular Redundancy pattern flowchart and state diagram











Rollback and N-modular Redundancy pattern performance, reliability, and availability

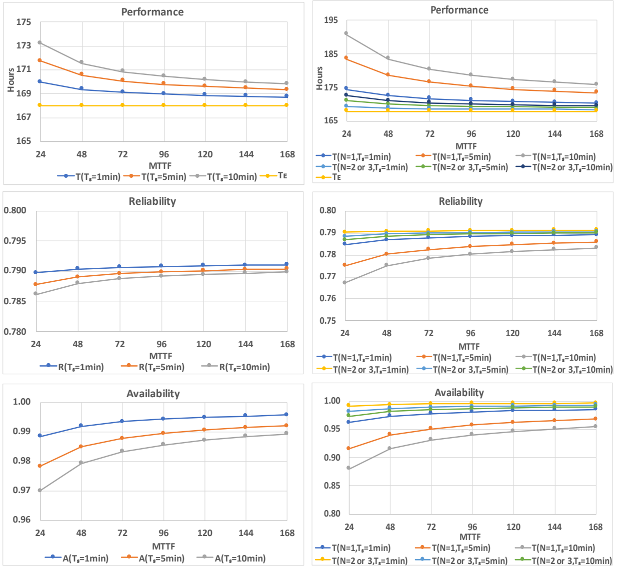
Multi-level Rollback

- A new approach for offering a separate resilience strategy for computation offloaded to a general-purpose computing graphics processing unit (GPGPU) accelerator.
- Rollback pattern for the application (level l = 0), Rollback pattern for the offloaded computation (level l = 1)
- 80% of the task's execution time T_E offloaded to a GPGPU.
- T_s and T_{l+r} are of 1 second.
- Performance

$$T = Tl_{=0} + Tl_{=1}$$

Rollback and N-modular Redundancy

- GPGPU errors and failures are detected and potentially corrected using redundancy.
- GPGPU redundancy N is 1, 2, or 3 and in time (α = 1).
- Time to replicate the input T_i and to compare the outputs T_o are 0.
- Time to reboot a GPGPU and use it again for redundancy T_r and the MTTR R are 1 minute.



Multi-level Rollback, and Rollback and N-modular Redundancy pattern performance, reliability, and availability

Future Work

- Provide models for other structural resilience design patterns.
- Develop a command line tool to generate plots for different parameters of specific design patterns model.
- Focus on models for power consumption and energy.

ACKNOWLEDGEMENTS

Work supported by the Early Career Program of the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing.