OAK RIDGE National Laboratory

Models for Resilience Design Patterns

Mohit Kumar and Christian Engelmann

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

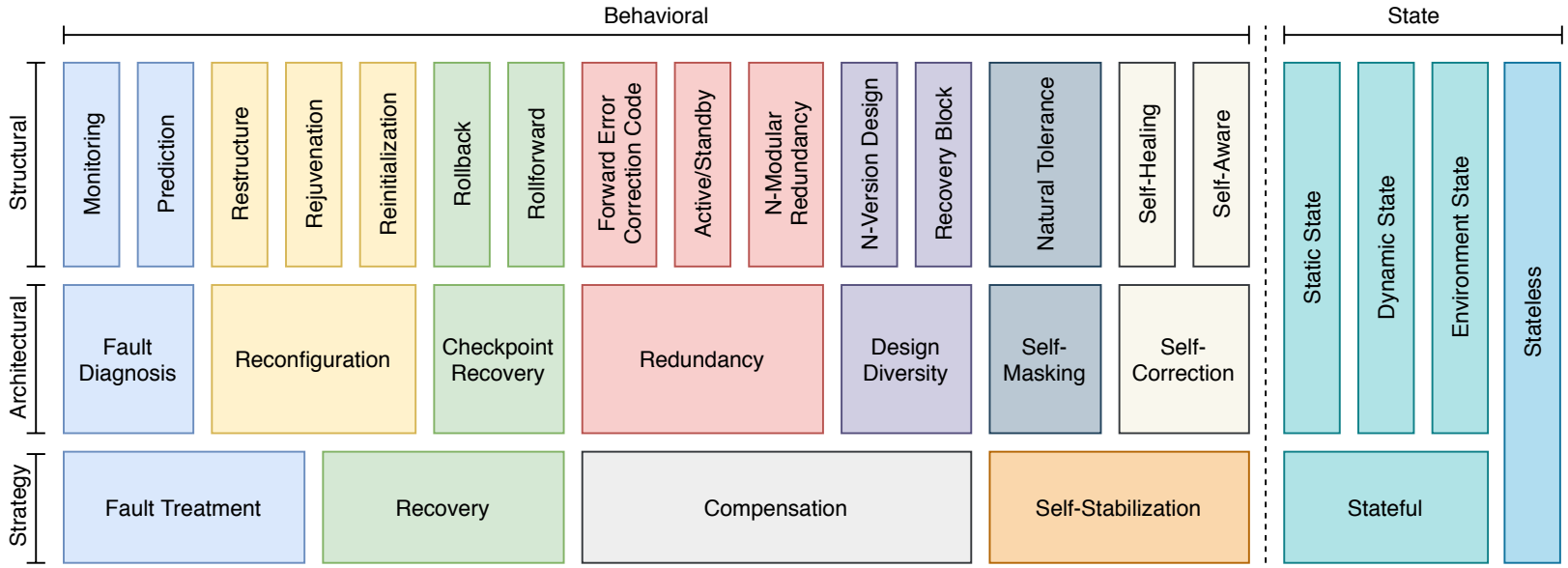
U.S. DEPARTMENT OF ENERGY

Resilience in HPC

- Resilience allows continue operation of extreme scale HPC systems
  - component counts increases resulting in decreased reliability
  - hardware and software complexity increases
  - unexpected issues, such as bad solder, dirty power, and early wear-out

- Resilience Design Patterns
  - identifies and evaluates repeatedly occurring resilience problems
  - coordinates solutions throughout hardware and software

Resilience Design Patterns



Previous Work

S. Hukerikar and C. Engelmann, "**Resilience design patterns: A structured approach to resilience at extreme scale**," *Journal of Supercomputing Frontiers and Innovations*, vol. 4, no. 3, pp. 4–42, Oct. 2017.

R. Ashraf, S. Hukerikar, and C. Engelmann, "**Pattern-based modeling of multi resilience solutions for high-performance computing**," in *ACM/SPEC International Conference on Performance Engineering*, 2018, pp. 80–87.

S. Hukerikar and C. Engelmann, "**Pattern-based modeling of high-performance computing resilience**," in *Lecture Notes in Computer Science: Workshop on Resiliency in High Performance Computing in Clusters, Clouds, and Grids*, vol. 10659, 2017, pp. 557–568.

Background - Terminology

- Performance is the total execution time of an application (T)
- Reliability is the probability of a system not experiencing a fault, error, or failure during operation

$$R(t) = 1 - F(t)$$

$$R(t) = e^{-\lambda t} \qquad MTTF = 1/\lambda$$

- Serial/Parallel reliability

$$R(t)_s = R(t)^N = e^{-\lambda t N} \qquad R(t)_p = 1 - (1 - R(t))^N = 1 - (1 - e^{-\lambda t})^N$$

Background - Terminology

- 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}

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}} \qquad A = \frac{MTTF}{MTTR + MTTF}$$

- Serial/ Parallel availability

$$A_s = A^N \qquad A_p = 1 - (1 - A)^N$$

Resilience Design Patterns Models

- Flowchart
- Performance
- Reliability
- Availability

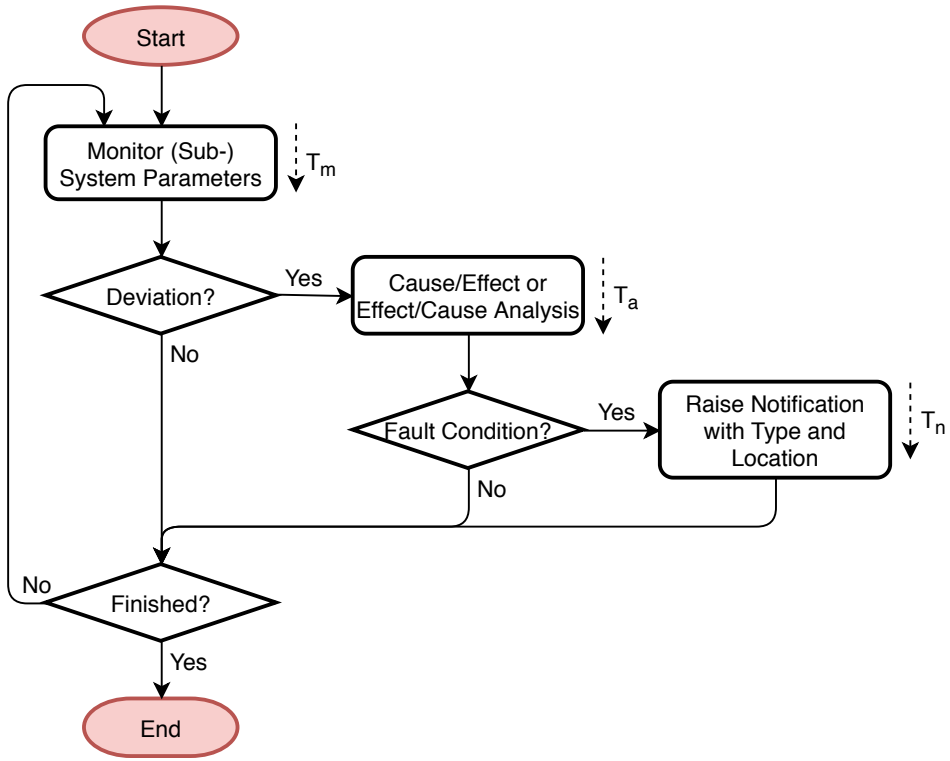
Monitoring

Monitoring pattern supports methods to recognize the presence of a defect or anomaly within a monitored system.

$$T_{f=0} = T_E + P(t_m)$$

$$T = T_E + P(t_m) + \frac{T_E}{M}(T_a + T_n)$$

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



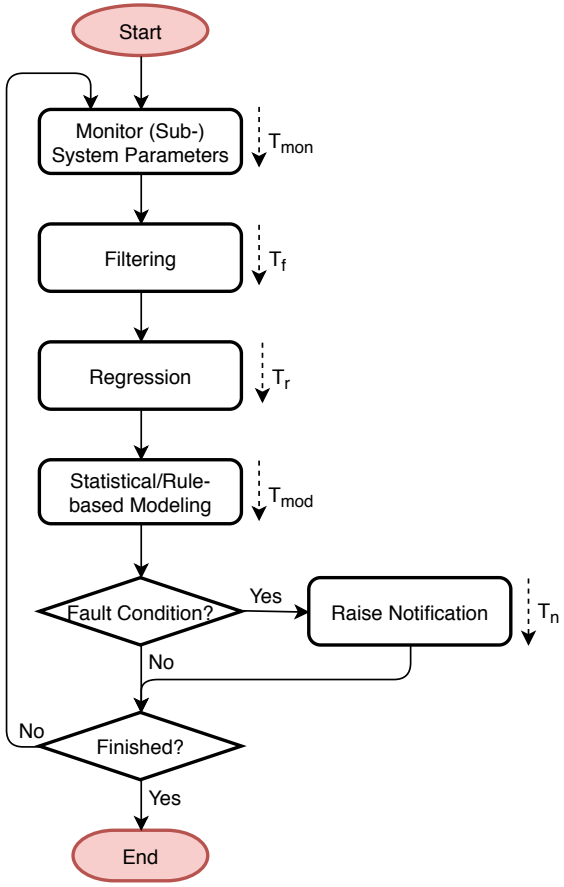
Prediction

Prediction supports methods to recognize the potential of a future defect or anomaly within a monitored system.

$$T_{f=0} = T_E + P\left(t_{mon} + t_f + t_r + t_{mod}\right)$$

$$T = T_E + P\left(t_{mon} + t_f + t_r + t_{mod}\right) + \frac{T_E}{M}(T_n)$$

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Restructure

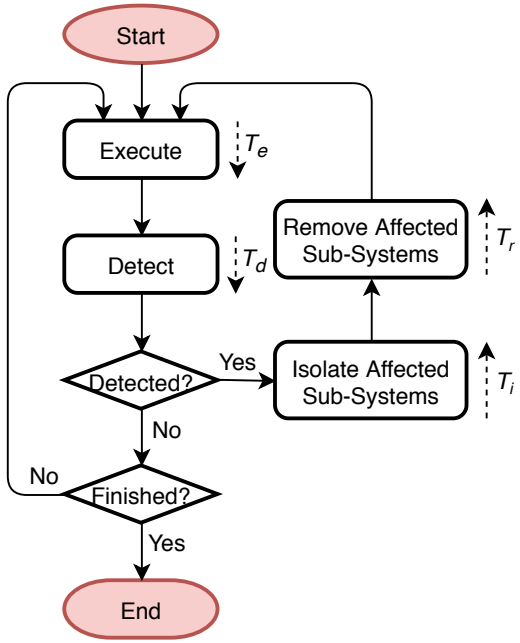
Restructure alleviates the impact of a fault, error, or failure on system operation by changing the interconnection between the subsystems in the overall system.

$$T_{f=0} = T_E + P(t_d)$$

$$T = T_E + P(t_d) + \frac{T_E}{M}(T_i + T_r)$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Rejuvenation

Rejuvenation alleviates the impact of a fault, error, or failure on system operation by restoring the affected subsystem or system to a known correct state.

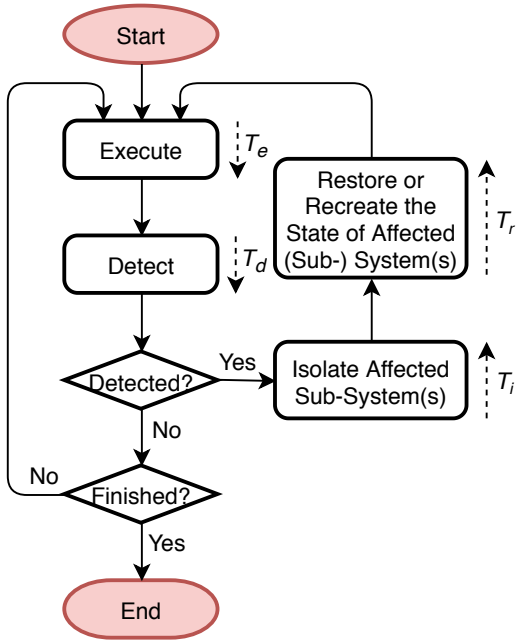
$$T_{f=0} = T_E + P(t_d)$$

$$T = T_E + P(t_d) + \left(\frac{T_E}{\tau} - 1\right) T_s + \frac{T_E}{M} T_{e,f}(\tau + T_s) + \frac{T_E}{M}(T_l + T_r)$$

$$\tau = \sqrt{2MT_s}$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Reinitialization

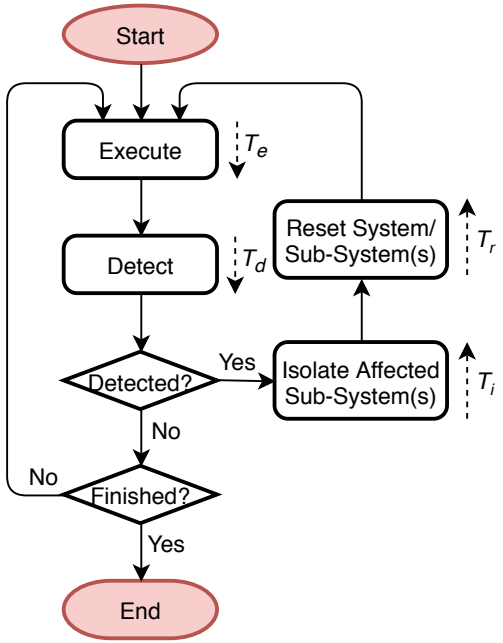
Reinitialization alleviates the impact of a fault, error, or failure on system operation by restoring the affected subsystem or system to its initial state.

$$T_{f=0} = T_E + P(t_d)$$

$$T = T_E + P(t_d) + \frac{T_E}{M}(T_i + T_r + T_E * 0.5)$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Rollback

Rollback supports resilient operation by restoring the system to the time when the last checkpoint occurred in the event of an error or failure.

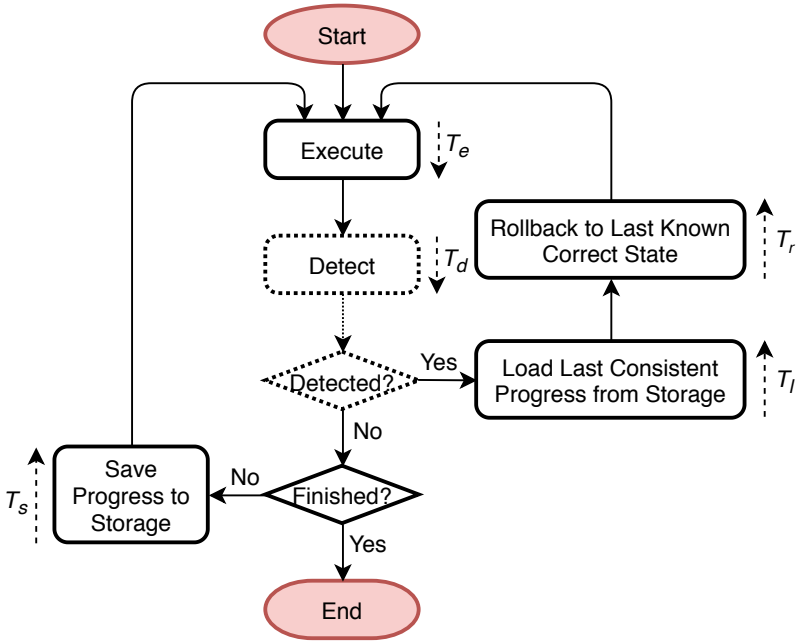
$$T_{f=0} = T_E + \left(\frac{T_E}{\tau} - 1\right) T_s$$

$$T = T_E + \left(\frac{T_E}{\tau} - 1\right) T_s + \frac{T_E}{M} T_{e,f}(\tau + T_s) + \frac{T_E}{M}(T_l + T_r)$$

$$\tau = \sqrt{2MT_s}$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Rollforward

Rollforward supports resilient operation by restoring the system to the time when the error/failure event occurred in the event of an error or failure.

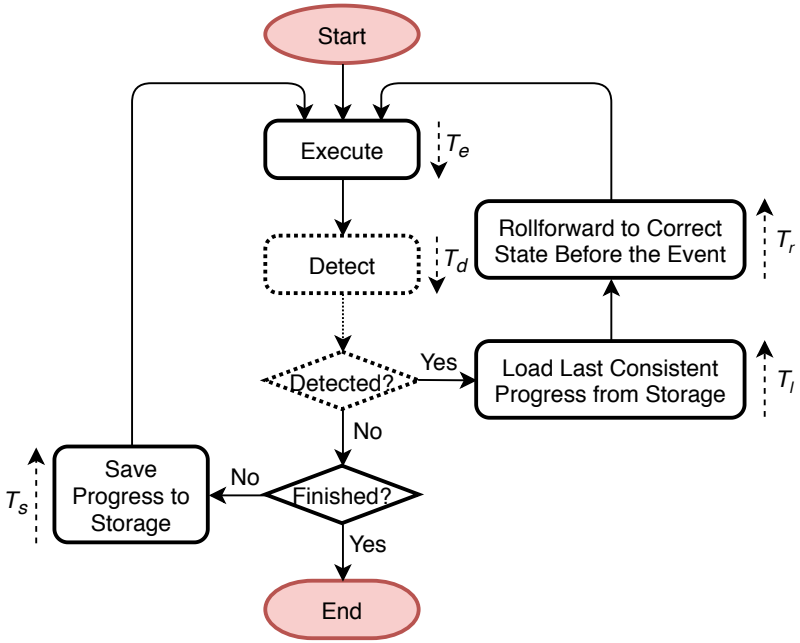
$$T_{f=0} = T_E + \left(\frac{T_E}{\tau} - 1\right) T_s$$

$$T = T_E + \left(\frac{T_E}{\tau} - 1\right) T_s + \frac{T_E}{M}(T_l + T_r)$$

$$\tau = \sqrt{2MT_s}$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Forward Error Correction Code

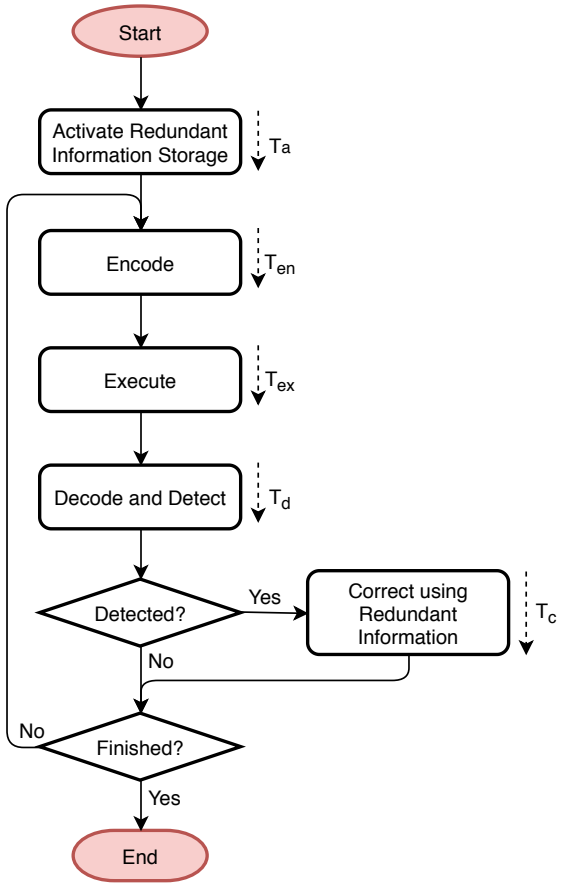
Forward Error Correction Code (FECC) supports resilient operation by applying redundancy to system state and optionally to system resources in the form of encoded system state.

$$T_{f=0} = T_E + T_a + P(t_{en} + t_d)$$

$$T = T_E + T_a + P(t_{en} + t_d) + \frac{T_E}{M}(T_c)$$

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

$$A = \frac{t_{pu}}{t_{pu} + t_{sd} + t_{ud}}$$



Active/Standby

Active/Standby supports resilient operation by applying redundancy in the form of N functionally identical replicas, using redundancy in space and potentially in time.

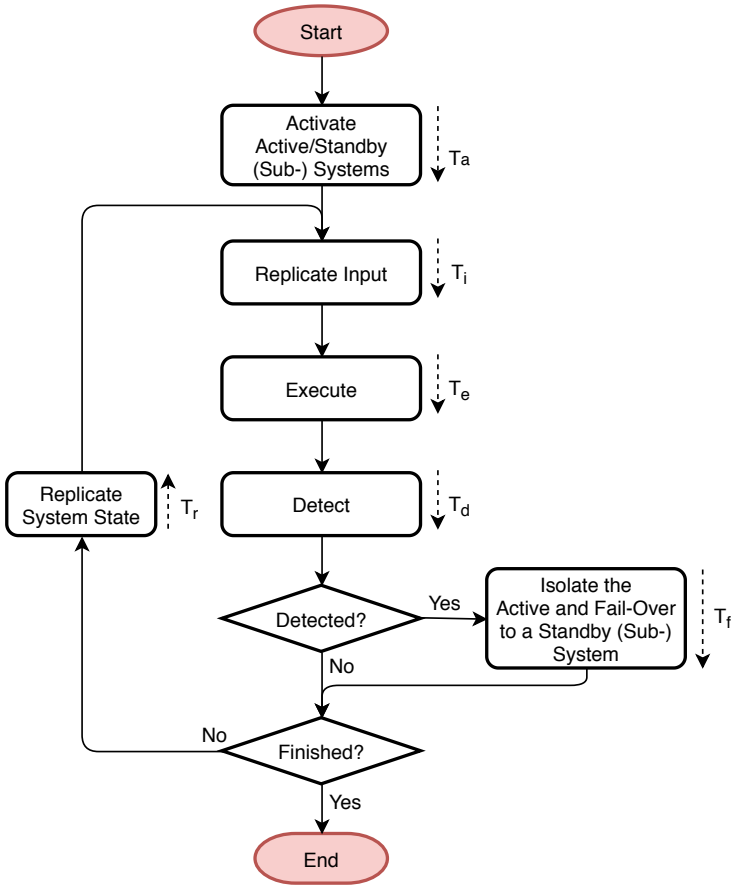
$$T_{f=0} = T_E + T_a + P(t_i + t_d + t_r)$$

$$T = T_E + T_a + P(t_i + t_d + t_r) + \frac{T_E}{M}(T_f)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_i + t_d + t_r) + \frac{T_E}{M}(T_f)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



N-modular Redundancy

N-modular Redundancy enables the continuous correct operation of a system by applying redundancy in the form of N functionally identical replicas.

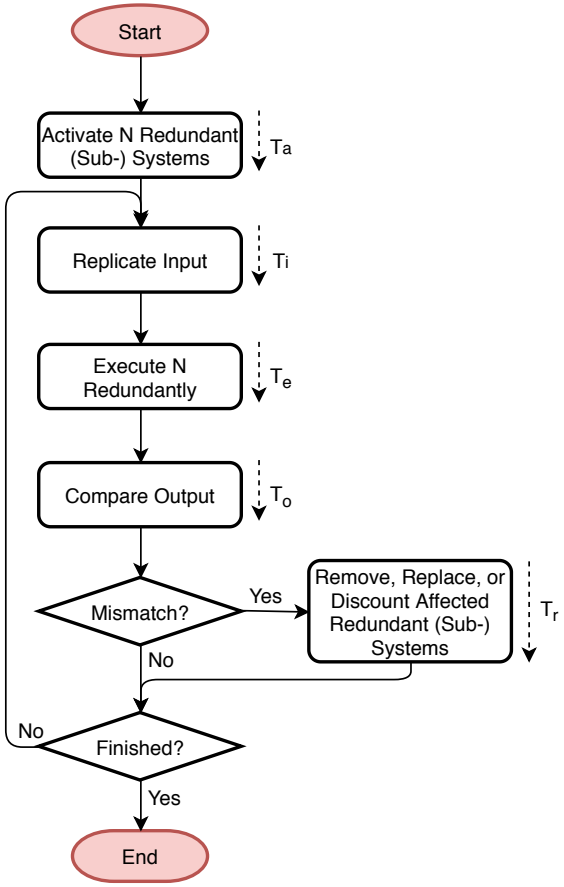
$$T_{f=0} = T_E + T_a + P(t_i + t_o)$$

$$T = T_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



N-Version Design

N-Version Design supports resilient operation by applying redundancy in the form of N functionally equivalent alternate system implementations.

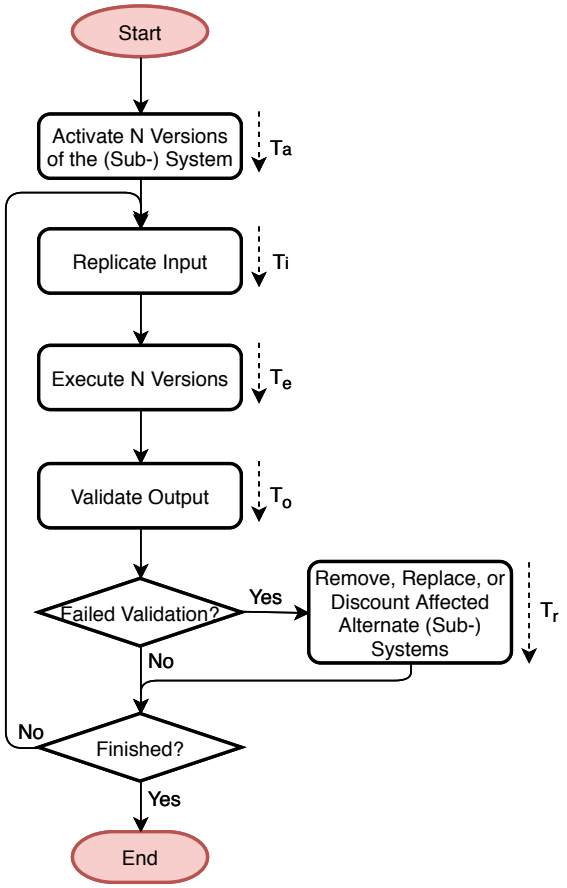
$$T_{f=0} = T_E + T_a + P(t_i + t_o)$$

$$T = T_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



Recovery Block

Recovery Block supports resilient operation by applying redundancy in the form of a functionally equivalent alternate system implementation encapsulated in a recovery block.

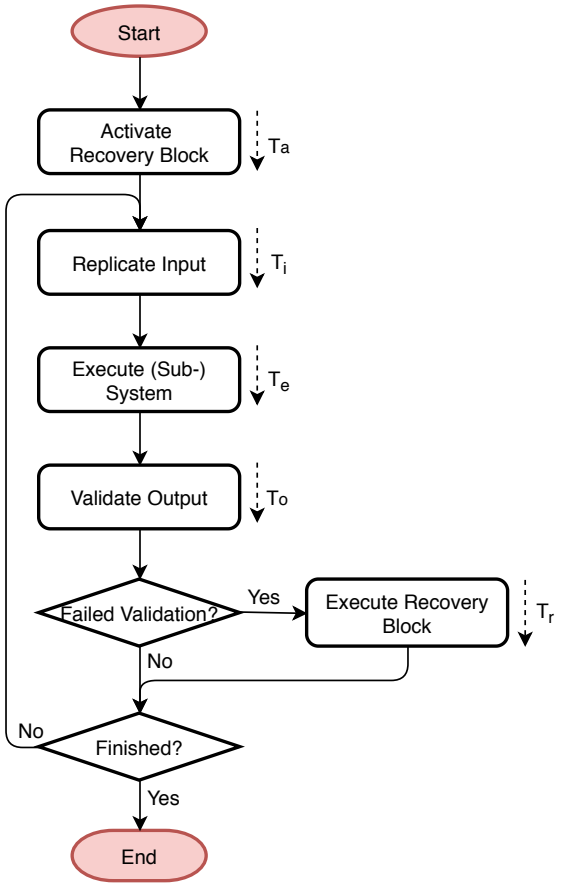
$$T_{f=0} = T_E + T_a + P(t_i + t_o)$$

$$T = T_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_i + t_o) + \frac{T_E}{M}(T_r)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



Natural Tolerance

Natural Tolerance relies on the capability of reaching a correct system state from an illegal system state after a finite number of execution steps using implicit error/failure detection and self-masking.

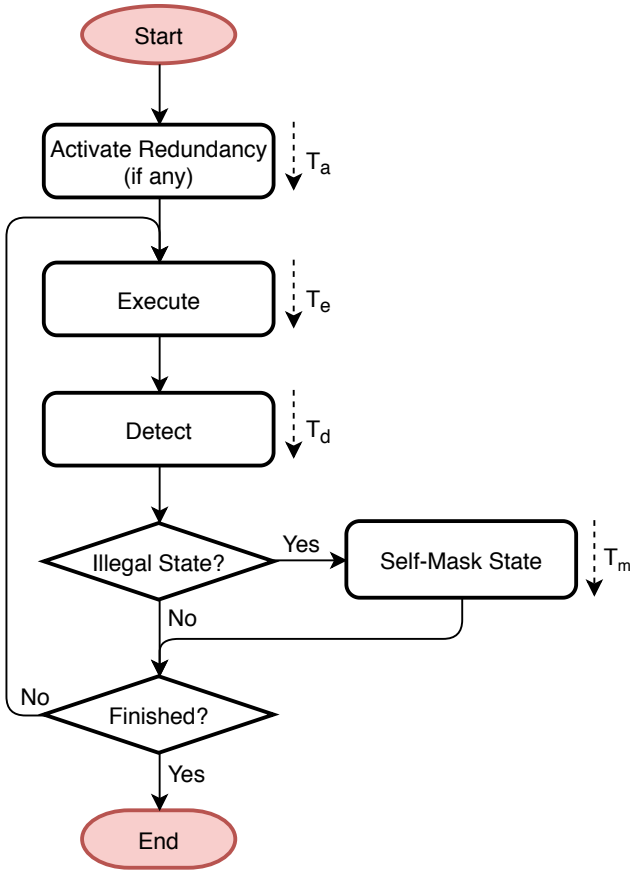
$$T_{f=0} = T_E + T_a + P(t_d)$$

$$T = T_E + T_a + P(t_d) + \frac{T_E}{M}(T_m)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_d) + \frac{T_E}{M}(T_m)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



Self Healing

Self-Healing relies on the capability of reaching a correct system state from an illegal system state after a finite number of execution steps using explicit error/failure detection and self-correction.

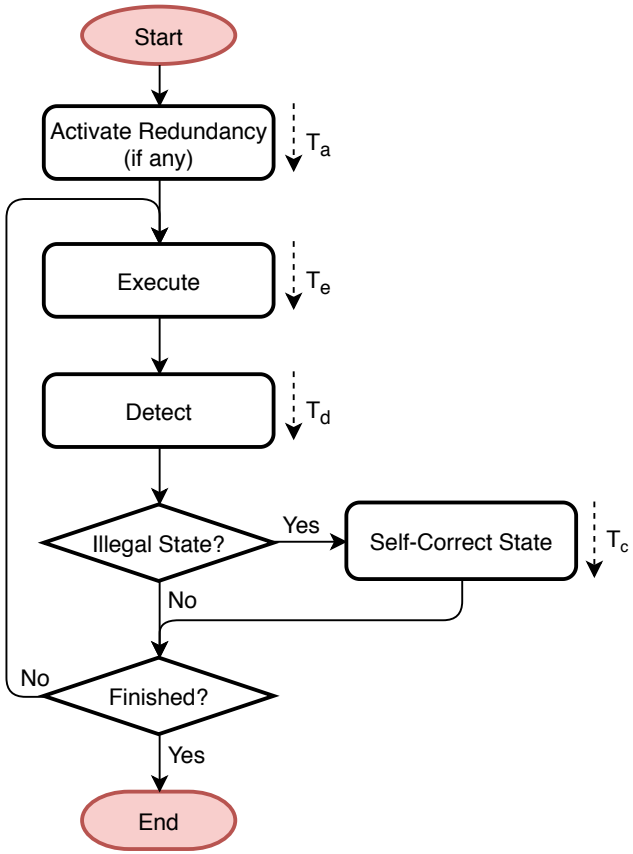
$$T_{f=0} = T_E + T_a + P(t_d)$$

$$T = T_E + T_a + P(t_d) + \frac{T_E}{M}(T_c)$$

$$T = \alpha T_E + (1-\alpha)NT_E + T_a + P(t_d) + \frac{T_E}{M}(T_c)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

$$A = 1 - (1 - A)^N$$



Self Aware

Self-Aware relies on the capability of reaching a correct system state from an illegal system state after a finite number of execution steps using explicit error/failure detection and self-correction.

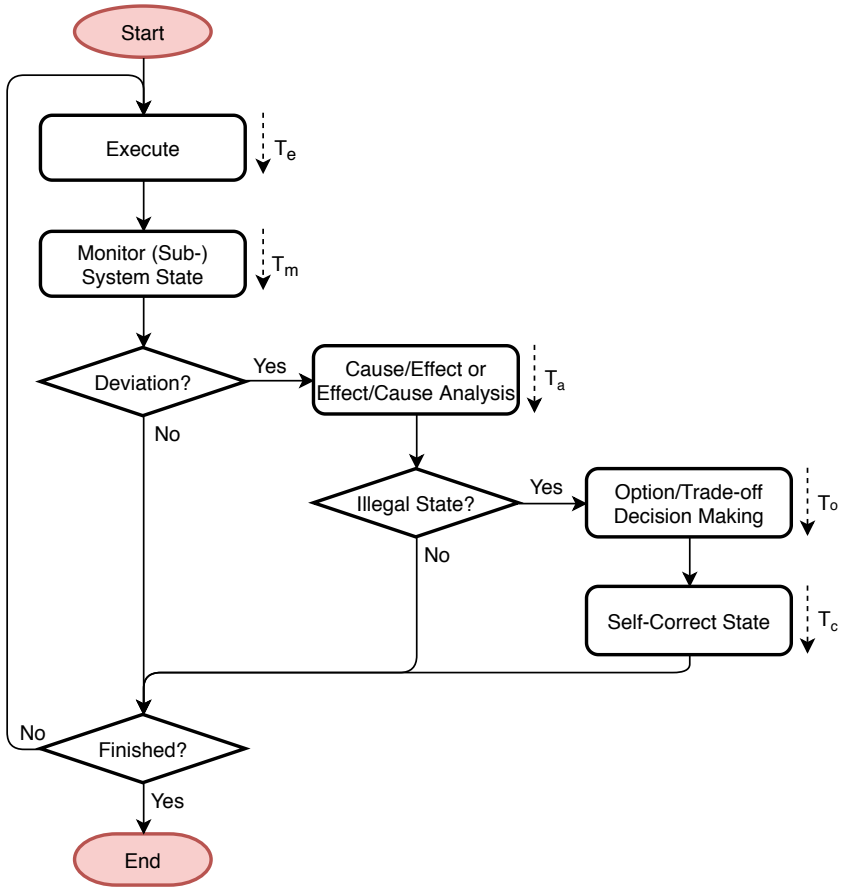
$$T_{f=0} = T_E + P(t_m)$$

$$T = T_E + P(t_m) + \frac{T_E}{M}(T_a + T_o + T_c)$$

$$T = \alpha T_E + (1-\alpha)NT_E + P(t_m) + \frac{T_E}{M}(T_a + T_o + T_c)$$

$$R(t) = 1 - (1 - e^{-\lambda t})^N$$

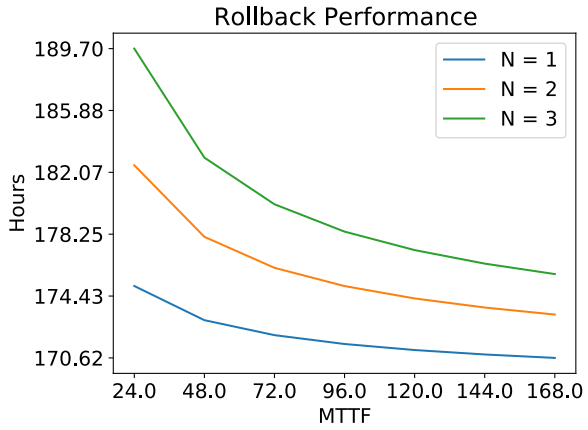
$$A = 1 - (1 - A)^N$$

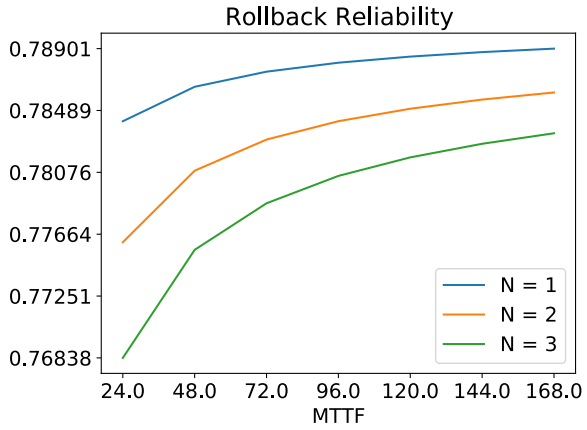


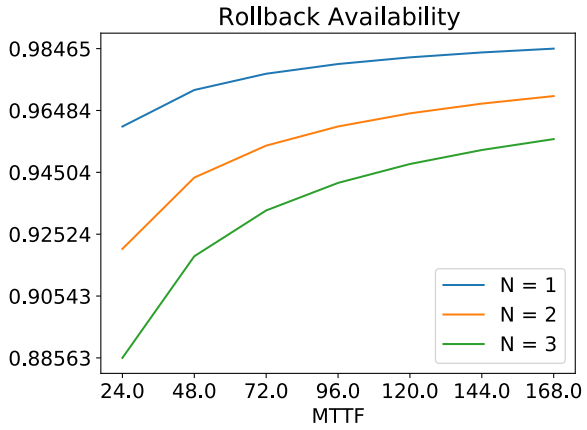
OAK RIDGE National Laboratory

Resilience Design Pattern Modeling Tool[1]

- Implements performance, reliability, and availability models
- Plots performance, reliability, and availability metrics
- Patterns objects are created and configured using an XML file
- Python – based







OAK RIDGE National Laboratory

[1]https://code.ornl.gov/6hk/rdpm

Conclusion

- Described performance, reliability, and availability models for all 15 structural patterns
- Provided flow charts and state diagrams
- Introduced the RDPM tool to study the characteristics of patterns and pattern combinations

Future Work

- Models for power consumption and energy
- Validates the models and verify the RDPM tool