Use Two-Level Rejuvenation to Combat Software Aging and Maximize Average Resource Performance

Chunhui Guo, Hao Wu, Xiayu Hua, Douglas Lautner, Shangping Ren
Email: {cguo13, hwu28, xhua, dlautner}@hawk.iit.edu, ren@iit.edu

Illinois Institute of Technology

August 24, 2015

Outline

Outline

Introduction

Software aging is a well-known phenomenon and has two effects:

- failure rate increase
- performance degradation

Software rejuvenation is a preventive and proactive maintenance solution for handling system aging effects.

- Rejuvenation Models: four-state model [Huang et al., 1995], two-level model [Koutras and Platis, 2011]
- Handle Failure Rate Increase: maximize reliability [Guo et al., 2015] and availability [Koutras and Platis, 2011]
- Handle Performance Degradation: P^2-resource model with one-level rejuvenation [Hua et al., 2015]

Introduction

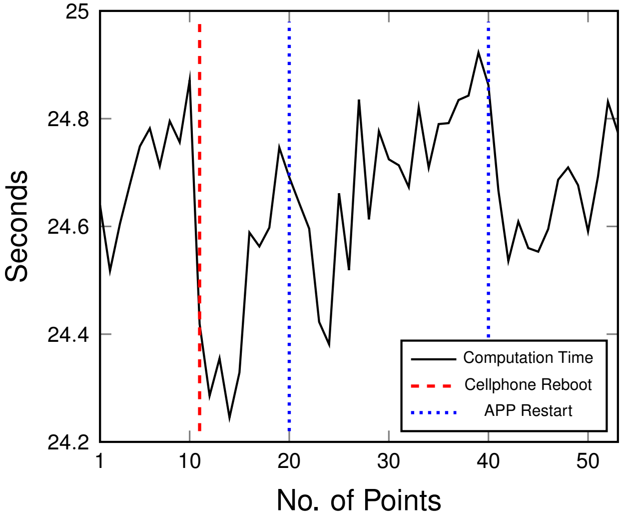


Figure: Aging Effect of Matrix Multiplication Time on Cellphone

Outline

Resource Performance Function

Due to aging, the resource performance decreases with time.

We assume $f(t) = 1 - a \times t$, where a is a constant and $0 \leq a < 1$.

If $a = 0$, the resource's performance does not degrade.

Rejuvenation Model

The resource can perform two-level rejuvenations

- Cold Rejuvenation: $f(t_0 + \Phi_C) = 1$
- Warm Rejuvenation: $f(t_1 + \Phi_W) = f_s \times p$, where f_s denotes the resource performance after previous rejuvenation and $0 < p < 1$
- Time Cost: $\Phi_C > \Phi_W$

Resource Rejuvenation Pattern

- Rejuvenation Threshold: $f(t) \leq r$, where $0 \leq r < 1$
- Rejuvenation Pattern: n ($n \in \mathbb{N}$) warm rejuvenations followed by one cold rejuvenation
- Periodic Rejuvenations: repeatedly rejuvenated by the pattern with period Π (rejuvenation hyperperiod)
- Max n: $N_{\max} = \lfloor \log_p r \rfloor$

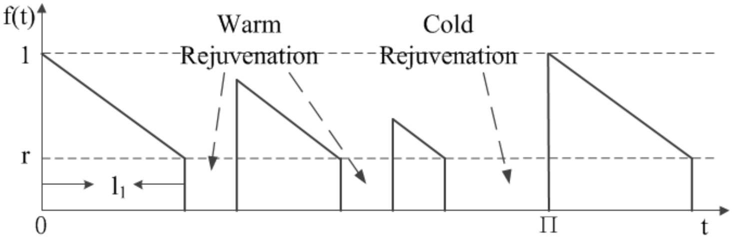


Figure: Resource Rejuvenation Pattern

Resource Model

$$R = (f(t), r, p, \Phi_W, \Phi_C, n)$$

- $f(t)$: resource performance function
- r: resource performance threshold to rejuvenate
- p: resource performance restore factor of a warm rejuvenation
- Φ_W: warm rejuvenation time cost
- Φ_C: cold rejuvenation time cost
- n: number of warm rejuvenations before a cold rejuvenation

Average Resource Performance

$$f_{\mathrm{ave}} = S_L / L$$

where L is system longevity and S_L is total resource supply within L.

Problem Formulation

Problem Definition

Given a resource $R(f(t), r, p, \Phi_W, \Phi_C, n)$, decide n that maximizes the average resource performance, i.e., f_{ave}, within its operational interval $[0, L]$.

Strategy

- First, we analyze the total resource supply S_L with a given n.
- Second, we present the MAX-AVE-PERFORMANCE algorithm to determine the optimal n with respect to maximizing average resource performance.

Outline

Resource Supply Analysis

- First, we analyze the resource supply S_Π within a rejuvenation hyperperiod Π.
- Second, we formalize the total resource supply S_L within the system longevity L on the basis of S_Π.

Resource Supply within Rejuvenation Hyperperiod Π

$$S_i = \int_{f^{-1}(p^{i-1})}^{f^{-1}(r)} f(t)dt = \int_{\frac{1-p^{i-1}}{a}}^{\frac{1-r}{a}} f(t)dt$$

$$S_\Pi = \sum_{i=1}^{n+1} S_i$$

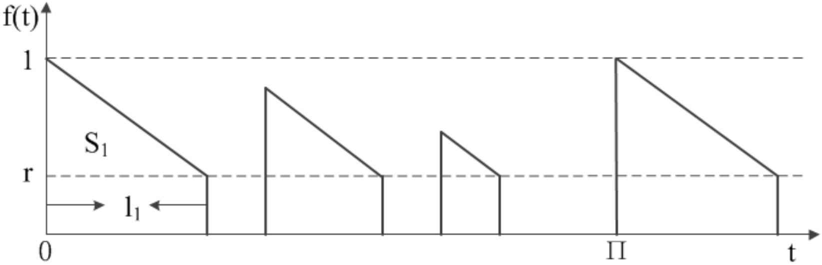
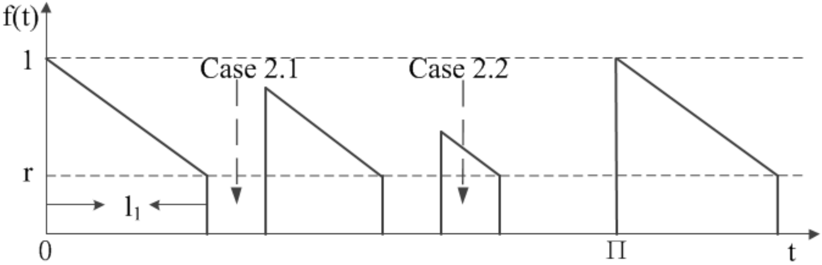


Figure: Resource Supply Analysis

Resource Supply within System Longevity L

- Case 1 ($L \bmod \Pi = 0$): $S_L = S_\Pi \cdot \frac{L}{\Pi}$
- Case 2 ($L \bmod \Pi \neq 0$): $S_L = S_\Pi \cdot \lfloor \frac{L}{\Pi} \rfloor + S_R$
  - Case 2.1 (l_R ends during a rejuvenation): $S_R = \sum_{i=1}^{j} S_i$
  - Case 2.2 (l_R ends when the resource is available):

$$S_R = \sum_{i=0}^{j-1} S_i + \int_{f^{-1}(p^{j-1})}^{f^{-1}(p^{j-1})+l_R-\sum_{i=0}^{j-1} l_i-(j-1)\Phi_W} f(t)dt$$



Outline

Average Resource Performance Maximization

As $n \in \mathbb{N}$ and $0 \leq n \leq N_{\max}$, the possible choices of n are limited. We present a linear search method to determine N^* maximizing $f_{\texttt{ave}}$.

Algorithm 1 MAX-AVE-PERFORMANCE

```
N* = 0
f_max = 0
N_max = ⌊log_p r⌋
for n = 0 to N_max do
    Calculate S_L
    f_ave = S_L / L
    if f_ave > f_max then
        N* = n
        f_max = f_ave
    end if
end for
return N* and f_max
```

Outline

Rejuvenation Strategy Impact Factors Evaluation

We evaluate the rejuvenation strategy impact factors from two aspects:

- the relationship between warm rejuvenation number n and average resource performance $f_{\mathtt{ave}}$;
- the impacts of warm/cold rejuvenation time cost on the optimal warm rejuvenation number N^* that maximizes the average resource performance $f_{\mathtt{ave}}$.

Relationship between n and f_{ave}

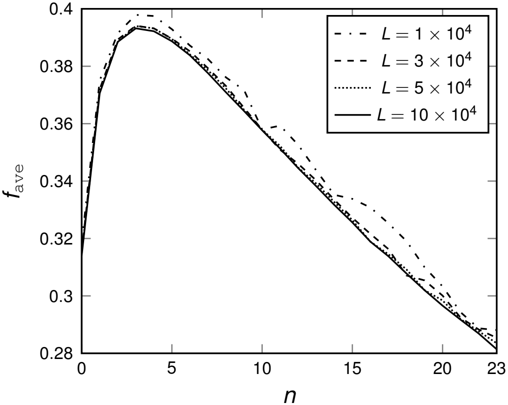
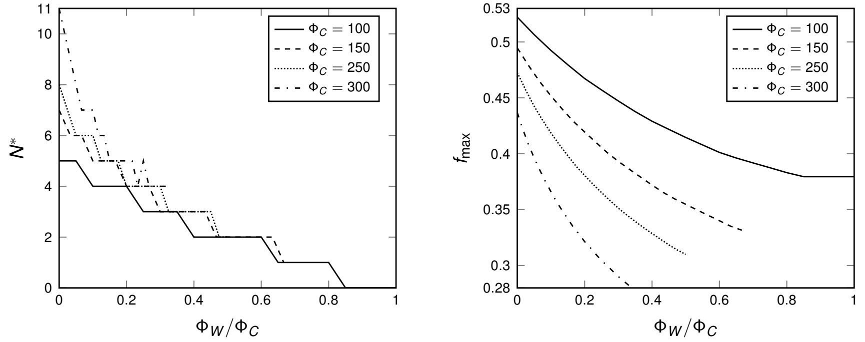


Figure: Average Resource Performance vs Warm Rejuvenation Number

The resource model with two-level rejuvenations achieves 25.22% higher average resource performance than the resource model with one-level rejuvenations ($n = 0$).

Warm/Cold Rejuvenation Time Cost Impact



(a) Optimal Number of Warm Rejuvenations

(b) Maximal Average Resource Performance

Figure: Warm/Cold Rejuvenation Time Cost Impact

Outline

Conclusion

- Propose the resource model using a two-level rejuvenation strategy to combat resource performance degradation due to software aging.
- Formally analyze the resource supply function of the proposed resource model.
- Present the MAX-AVE-PERFORMANCE algorithm to maximize the average resource performance.
- Validate the resource supply analysis through simulations.
- Compared with the resource model with one-level rejuvenations, the proposed resource model with two-level rejuvenations achieves 25.22% higher average resource performance.

Future Work

- Analyze task schedulability of the resource model with two-level rejuvenations, and study the optimal rejuvenation pattern maximizing the task schedulability.
- Obtain resource performance degradation function from experiments.

References

Guo, C., Wu, H., Hua, X., Ren, S., and Nogiec, J. (2015).
Maximize system reliability for long lasting and continuous applications.
In *New Contributions in Information Systems and Technologies*, volume 353 of *Advances in Intelligent Systems and Computing*, pages 603–612. Springer International Publishing.

Hua, X., Guo, C., Wu, H., and Ren, S. (2015).
Schedulability analysis for real-time task set on resource with performance degradation and periodic rejuvenation.
In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2015 IEEE 21th International Conference on*.

Huang, Y., Kintala, C., Kolettis, N., and Fulton, N. (1995).
Software rejuvenation: analysis, module and applications.
In *Fault-Tolerant Computing, 1995. FTCS-25. Digest of Papers., Twenty-Fifth International Symposium on*, pages 381–390.

Koutras, V. and Platis, A. (2011).
Applying partial and full rejuvenation in different degradation levels.
In *Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on*, pages 20–25.

Questions?

Thank You