Modeling and Simulation of Error Recovery in Concurrent Processing Systems

Dr. Silvia Heubach

Department of
Mathematics and Computer Science

California State University Los Angeles

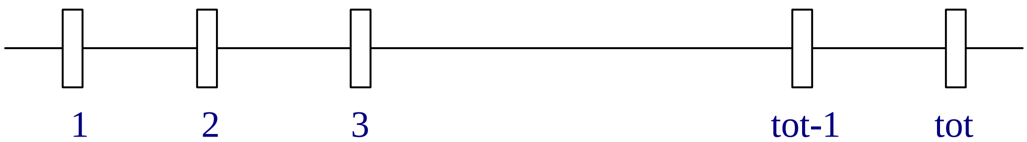
Introduction

- Model and Parameters
- Iterative versus Selective Rollback
- Simulation
- Cost Derivation
- Results of Comparison
- Conclusion

Model Description

- Global checkpointing with n processes
- Inter-communications between processes after exponential times with parameters λ_{ij}
- Failures after exponential times with parameters φ_i
- Acceptance tests after exponential times with parameters α_i

Iterative versus Selective Rollback



- Iterative Rollback
  - **Rolls back to most recent checkpoint (CP) and attempts recovery**
  - **If recovery fails from CP k, processes are rolled back to CP k-1**

- Selective Rollback
  - **Selects checkpoint for recovery based on distribution of latency times**
  - **Pairs of checkpoints are compared for smaller expected cost of recovery**

- Both methods recover eventually if failure occurred after CP 1

Cost of Recovery from Checkpoint k

- CT = cycle time = C+CL
  - **C = time between checkpoints**
  - **CL = time to load checkpoint**

- tot = # of currently established checkpoints

- d = time between acceptance test and last checkpoint

- T(k) = cost of recovery from CP k
  - **T(k) = (tot – k) CT + CL + d**

Total Cost of Recovery (Iterative Rollback)

- CP r = checkpoint to which processes must be rolled back for recovery
  - **Failure between CP k and CP k+1 => r = k**
  - **Failure before CP 1 => r = 1**

- Total cost of recovery
  - **TOI =T(r) + T(r+1) + …+ T (tot)**

Selective Rollback

- Choice of checkpoint based on latency distribution

- Simulation creates empirical distribution function for latency distribution

- Simulation based on events with exponential waiting times
  - **next event after exponential time with parameter**

$$\mu = \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \neq j}}^{n} \lambda_{ij} + \sum_{i=1}^{n} \varphi_i + \sum_{i=1}^{n} \alpha_i$$

Simulation

- 5 Simulations (up to time 200 hours)

- Ranges for average times (in hours) between events

- Simulation based on events with exponential waiting times
  - **Inter-communications [0.25, 1]**
  - **Failures [10, 40]**
  - **Acceptance tests [1, 2]**

Simulation

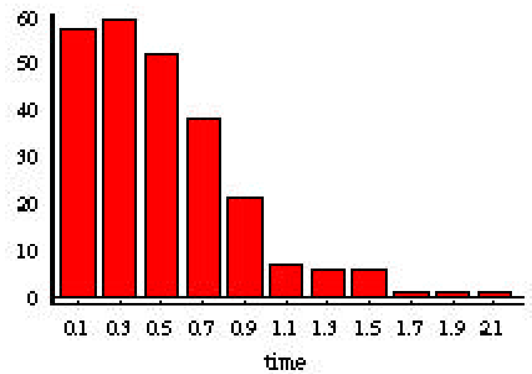
- Sample parameter values for simulation 1

λ_{ij}	1	2	3	4
1	*	3.50	1.85	3.92
2	1.84	*	2.30	3.30
3	3.34	1.16	*	2.80
4	3.77	3.05	1.43	*

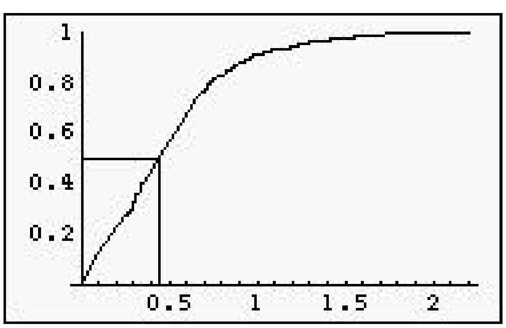
i	φ_i	α_i
1	0.080	0.087
2	0.098	0.949
3	0.080	0.971
4	0.066	0.770

Latency Times

- Combined Latency Times



- Latency Distribution
  - **median ~ .43**



Latency Times

- P(k) := P(failure occurred after CP k) = P(latency < (total - k + 1) C)
- C(l,k) = cost of successful recovery from CP l given that the first recovery attempt starts at CP k > l
- EC(k) = expected cost of recovery given recovery from CP k was unsuccessful
  - **EC(k) = [P(1) - P(2)] C(1,k-1) + [P(2) - P(3)] C(2, k-1) +…+[P(k-1) - P(k)] C(k-1,k-1)**

Selective Rollback Algorithm

- **Step 1** (Initialization)
  - **k = min { m, total}, P = 0, TOS = 0**
- **Step 2** (Termination Condition)
  - **If k = 1 , go to Step 4**
- **Step 3** (Comparison)
  - **Compare expected cost of recovery from CP k and k-1. If cost from CP k-1 smaller, then k:=k-1; go to Step 2**
- **Step 4** (Rollback; updating of variables)
  - **Roll back to CP k and attempt recovery**
  - **P:= P(k), TOS = TOS + T(k)**
  - **If recovery successful, resume normal operation; otherwise**
    - if k > 1, go to Step 2
      else indicate that recovery is not possible

Selective Rollback Algorithm

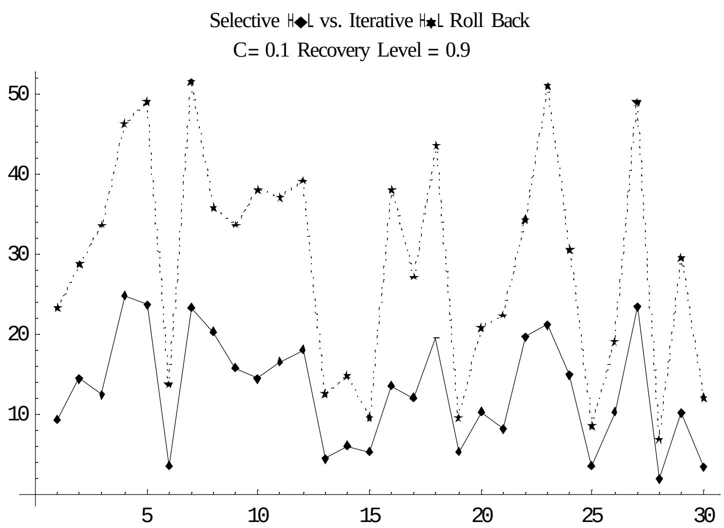
- **Step 3** (Comparison)
  - **If (P(k-1) - P) T(k-1) + (1 - P(k-1)) EC(k-1) <= (P(k) - P) T(k) + (1 - P(k)) EC(1) , then k:=k-1; go to Step 2**

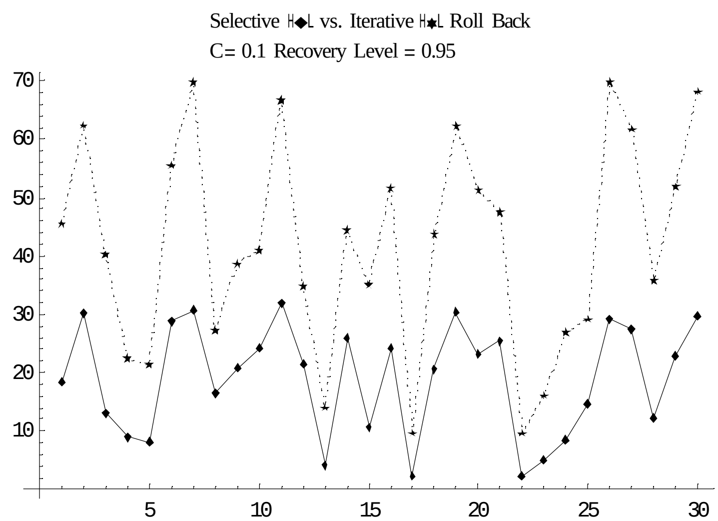
- **Step 4** (Rollback; updating of variables)
  - **P:= P(k), TOS = TOS + T(k)**
    - **If recovery is unsuccessful from CP k, the probabilities P(l) are replaced by conditional probabilities (P(l)-P(k))/(1-P(k)) and the expected values EC(k) are likewise divided by (1-P(k)). This is achieved by setting P = P(k).**
    - **Cost is updated to reflect the accumulated cost**

Cost Comparison

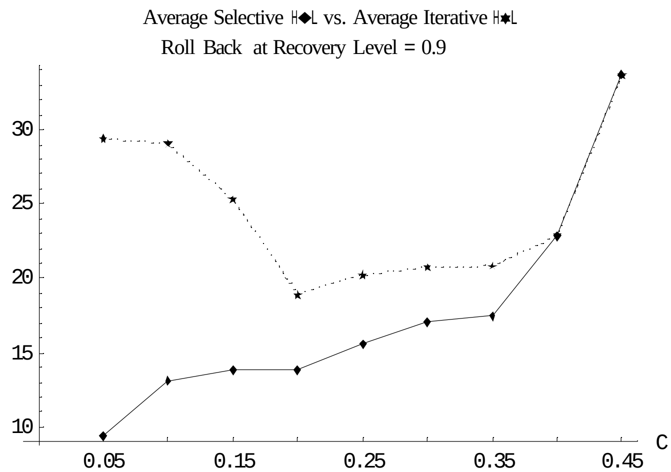
- Number of checkpoints selected to achieve a given recovery level (0.9 or 0.95) using quantiles of latency distribution
- Values for C ranging from 0.05 to 0.45 (~ median of latency distribution)
- 30 simulations for each level of recovery and choice of C
- If no recovery was possible, simulation stopped

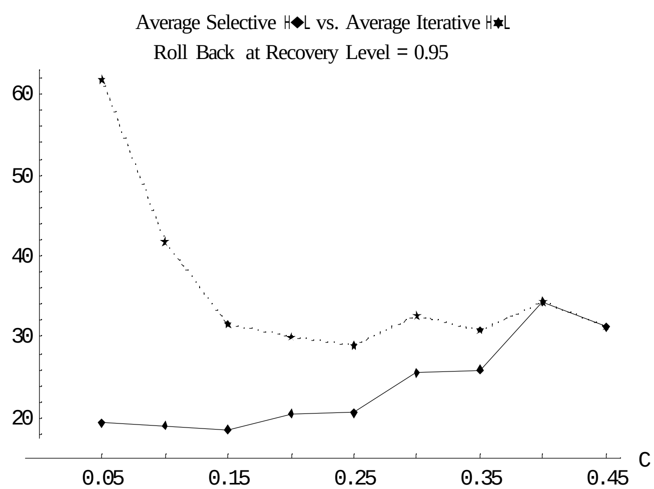
Results C = 0.1


Selective H◆L vs. Iterative H★L Roll Back
C= 0.1 Recovery Level = 0.9
50
40
30
20
10
5
10
15
20
25
30


Selective H◆L vs. Iterative H★L Roll Back
C= 0.1 Recovery Level = 0.95
70
60
50
40
30
20
10
5
10
15
20
25
30

Results for Averaged Cost vs. C





Conclusion

- Selective rollback has smaller or equal cost in all cases
- Difference most pronounced for small cycle time
- Checkpoint selection
  - m = total number of checkpoints
  - m = checkpoint to which processes roll back

90%			95%		
C	*m*	$\tilde{m}$	*C*	*m*	$\tilde{m}$
.05	**19**	**4**	**.05**	**24**	**6**
.10	**11**	**4**	**.10**	**13**	**5**
.15	**8**	**4**	**.15**	**9**	**4**
.20	**6**	**3**	**.20**	**8**	**5**
.25	**5**	**3**	**.25**	**6**	**4**
.30	**5**	**3**	**.30**	**6**	**4**
.35	**4**	**2**	**.35**	**5**	**3**
.40	**4**	**4**	**.40**	**5**	**5**
.45	**4**	**4**	**.45**	**4**	**4**

Practical Issues

- Simulation can be enhanced by dynamically adapting the latency distribution
- Actual Implementation
  - **Initially use iterative method with small C to create data for approximate latency distribution**
  - **Use approximate distribution function to implement selective rollback**

Future Work

- Error bounds on difference between actual latency distribution and approximate distribution
- Theoretical distribution and related parameter estimation?