CPE 633
Chapter 6 – Checkpointing

Dr. Rhonda Kay Gaede

UAH

Introduction

- There still are some applications that take a very long time to execute.
  - _________ ____________ – Large ____________ of three-dimensional cells __________, examples include weather and climate modeling
  - _____________ – Optimal deployment under a ________ ___ __ __________ – examples include airlines
  - ___________ – N-body simulations, ____ ___________, ____________ __________ of the universe
  - _____________ – study of ________ _________
- When a program takes very long to execute, the ____________ ___ ___ ________ and the ____ ___ ___ ________ become significant.

Analytical Model

- Consider a program that takes T time units to execute if ___ ________ occur during its execution.
- Suppose the system suffers ________ failures according to a _________ ________ with a rate of λ failures per time unit.
- Assume here that ________ failures are point failures that _______ __ _____ in the system and then __ _____.
- All the ____________ done by the program _____ ___ ___ _____ is lost.
- The system takes __________ _____ to recover from the failure.
- Let E be the ________ ________ _____, including any computational work ____ ___ _______.
- Two cases
  1) ___ _______
  2) ___ _______ ___ _______

Analytical Model

- Case 1
  - Probability of __ _________ $=e^{-\lambda T}$, contribution to average execution time is ________
- Case 2
  - The first failure _______ __ time units into the execution time T, τ time units are _____ and the _________ execution time is _____.
  - The probability that the ___ ________ falls in the ________ $[\tau, \tau+d\tau]$ is $\lambda e^{-\lambda\tau} d\tau$, τ in ______
  - Contribution to average execution time

$$\int_{\tau=0}^{T} (\tau + E)\lambda e^{-\lambda\tau} d\tau = \frac{1}{\lambda} + E - e^{-\lambda T}\left\{\frac{1}{\lambda} + T + E\right\}$$

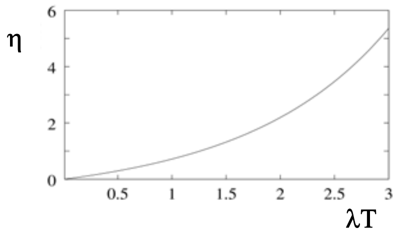
Analytical Model

- Adding the contributions of case 1 and case 2 and

$$E = Te^{-\lambda T} + \frac{1}{\lambda} + E - e^{-\lambda T}\left\{\frac{1}{\lambda} + T + E\right\} \qquad E = \frac{e^{\lambda T} - 1}{\lambda}$$

- ____________ the _______ imposed by the failure process _______ by the __________ time

$$\eta = \frac{E - T}{T} = \frac{E}{T} - 1 = \frac{e^{\lambda T} - 1}{\lambda T} - 1$$



η starts quite small and then _____ _________

6.1 What is Checkpointing?

- Checkpointing is _____ ____________ results.
- Consider adding up a long ____ ___ ________ with a calculator. Record the ______ ___ on paper every ___ numbers. If a mistake is made, at most ___ _______ need to be repeated.
- In general, a checkpoint is a ________ of the _____ _____ of the process at the _______ it was taken, it represents all the information needed to _______ the process ____ ____ ______.
- ______ the checkpoint on ____ ________, i.e., disks.
- Checkpoint _______ is the _______ in application execution time.
- Checkpoint _______ is the time needed to ____ ___ _________.
- The size of the checkpoint is _____________ and may ___ _______.

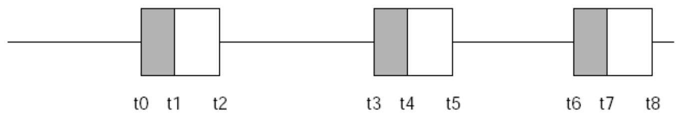
6.1.1 Why is Checkpointing Nontrivial?

- Issues
  - At ____ _____ should we checkpoint? How _________ to the ____ should the checkpointing process be?
  - _____ ______ checkpoints should we have?
  - At _____ ______ in the program should we checkpoint?
  - How can we _______ checkpointing ________?
  - How do we checkpoint systems in which there may or may not be a ______ __________, and in which _________ _____ between individual processes?
- There is also the question of __________ the ________________ at a ___________ _____ if that becomes necessary.
- Furthermore, program ____________ with the _______ ______ should be carefully considered because some of them cannot be undone, for example, _____________ ________ __ ____________.

6.2 Checkpoint Level

- ______________ Checkpointing
  - Checkpointing procedures are included in the ________, _____________ to the _____
- ____________ Checkpointing
  - A ___________ _________ is provided to do the checkpointing, applications are ______ to the __________, transparent to ______
- __________________ Checkpointing
  - The ______________ is responsible, this approach provides the user with the ___________ _______ but is expensive to ____________ and _______
- The ____________ available to _______ _____ is different.
- ____________ ___, ________ ___ and __________ ___ files may have to be managed.

6.3 Optimal Checkpointing- An Analytical Model



- ____________ is the checkpointing activity that cannot be done __ ___________ with the application execution.
- Notation
  - T_{lt} ,the ________, is the time interval between the ____ and ____ of checkpointing, assumeT_{lt} = t_2-t_0=t_5-$_{t3}$=t_8-t_6
  - T_{ov}, the ________, is the part of the T_{lt} interval during which the application execution is ________ ____ _________________, assume T_{ov} = t_1-t_0=t_4-t_3=t_7-t_6
  - Tr, _________ time, includes the time spent __ __ _______ state plus the time it takes to ________ ___ ___ _________ state.

6.3 Optimal Checkpointing- An Analytical Model

- Consider the _________ between i and i+1 checkpoint
  - E_{int}, time between the ith and i+1st checkpoint,
  - T_{ex}, __________ _____ during this period
  - If E_{int} is ____________ ___ ___________, E_{int} = T_{ex}+T_{ov}
  - If there is a ________ τ units into ________, work done since the ________ _____________ is lost
    - This work is the _______ _____ done during the latency period, ________ and the work done since the _______ ______. Lost work = __________
  - ___________ time must be added
    - Extra time = ____ _____ + ________ = ________________

6.3.1 Time Between Checkpoints – A First-Order Approximation

- Assume at most ____ ________ between checkpoints
  - Case 1 – no failure $(T_{ex} + T_{ov})e^{-\lambda(T_{ex}+T_{ov})}$
  - Case 2 – one failure

$$\left(1 - e^{-\lambda(T_{ex}+T_{ov})}\right)\left\{T_{ex} + T_{ov} + \frac{T_{ex}+T_{ov}}{2} + T_r + T_{lt} - T_{ov}\right\}$$

$$= \left(1 - e^{-\lambda(T_{ex}+T_{ov})}\right)\left\{\frac{3T_{ex}}{2} + \frac{T_{ov}}{2} + T_r + T_{lt}\right\}$$

  - Total

$$E_{int} \approx \frac{3}{2}T_{ex} + \frac{T_{ov}}{2} + T_r + T_{lt} - \left(\frac{T_{ex}}{2} + T_r + T_{lt} - \frac{T_{ov}}{2}\right)e^{-\lambda(T_{ex}+T_{ov})}$$

$$\frac{dE_{int}}{dT_{ov}} \gg \frac{dE_{int}}{dT_{lt}}$$

6.3.2 Optimal Checkpoint Placement

- How do we select T_{ex} so as to ________ expected _____ ___________ time, or the ______ __ ______

$\eta = E_{int}/T_{ex} - 1$?

- Using $e^{-\lambda(T_{ex}+T_{ov})} \approx 1 - \lambda(T_{ex} + T_{ov})$, we obtain

$$\eta = \frac{\frac{3}{2}T_{ex} + \frac{T_{ov}}{2} + T_r + T_{lt} - (\frac{T_{ex}}{2} + T_r + T_{lt} - \frac{T_{ov}}{2})(1 - \lambda(T_{ex} + T_{ov}))}{T_{ex}} - 1$$

$$= \frac{(T_{ex} + T_{ov})\left[1 + \lambda(\frac{T_{ex}}{2} + T_r + T_{lt} - \frac{T_{ov}}{2})\right]}{T_{ex}} - 1$$

- Differentiating to _________ η

$$T_{ex}^{opt} = \sqrt{\frac{2T_{ov}}{\lambda} + 2T_{ov}\left(T_r + T_{lt} - \frac{T_{ov}}{2}\right)} \qquad N_{opt} = \frac{T}{T_{ex}^{opt}} - 1$$

UAH Chapter 6 CPE 633

6.3.3 Time Between Checkpoints – A More Accurate Model

- To relax the assumption of ___ _____ ____ failure in an intercheckpoint interval, deal more accurately with Case 2. Suppose a failure occurs at time τ ($\tau < T_{ex} + T_{ov}$) with probability $\lambda e^{-\lambda\tau} d\tau$, The amount of time ________ due to the ________ is $\tau + T_r + T_{lt} - T_{ov}$, after which the computation will take an added ______ _____ of E_{int}. The contribution of case 2 is

$$(T_{ex} + T_{ov})e^{-\lambda(T_{ex}+T_{ov})}$$

- Since T_{ov} appears in the ____________, E_{int} is more sensitive to T_{ov} than to T_{lt}

UAH Chapter 6 CPE 633

6.3.3 Time Between Checkpoints – A More Accurate Model

- Consider again the figure of merit **$\eta = E_{int}/T_{ex} - 1$**
- Look for T_{ex} that minimizes η, it is one that satisfies the following equation

$$e^{\lambda(T_{ex}+T_{ov})} = \frac{1}{1 - \lambda T_{ex}}$$

- Thus, the optimal value, does not depend on the ___________ or the ____________ _____, just the __________.

$$N_{opt} = \frac{T}{T_{ex}^{opt}} - 1$$

- In ____________ checkpointing, the application cannot be executed ___ ___________ with the checkpointing. We therefore have $T_{lt} = T_{ov}$ and the overhead ratio is

$$\eta = \frac{(T_r + \frac{1}{\lambda})(e^{\lambda(T_{ex}+T_{ov})} - 1)}{T_{ex}} - 1$$

6.3.4 Reducing Overhead - Buffering

- Write the checkpoint into _____ _________ and initiate a _____ of checkpoint from _____ _________ to disk
- A refinement is called ____________ buffering.
  - _______ _________ of the process state remain _________ since the last checkpoint.
  - Change ______________ of process pages to ________, _________ ___________ triggered if try to write
- This is an example of ____________ checkpointing
  - Advantage – The _____ of the ____________ checkpoints will be _____ ______.
  - Disadvantage – The process of recovery is _____ _____________, one has to build the system state by examining a __________ of ______________ checkpoints.

6.3.4 Reducing Overhead – Memory Exclusion

- Another approach to lowering the checkpointing __________ attempts to reduce the _______ of ________________ that must be stored upon a checkpoint.
- Two types of variables unnecessary to record
  - Those that have ____ _____ _________
  - Those that are ______
    - _______ _____ _______
    - _____ ________ ___ ___ ______
  - Finding _____ variables
    - Code segment – ______________________
    - Stack segment – the contents of addresses _______ the _____ ________ are _____
    - Heap segment – the contents of the _____ _____ are ________ and some user-level checkpointing packages (______) specify regions _________ or _________

6.3.5 Reducing Latency

- Checkpoint ______________ has been suggested as one way to reduce ___________.
- How well this works depends on
  - The ______ of the __________ (_____)
  - The _____ ________ to execute the compression algorithm

6.4 Cache-Aided Rollback Error Recovery (CARER)

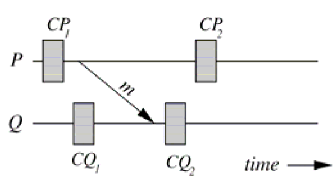
- Reducing checkpointing ___________ allows us to increase the checkpointing _____________, thereby reducing the _______ of a _________ upon failure.
- CARER seeks to reduce the checkpointing time by marking the _______ _________ in main memory and the cache as part of the _____________ ________.
- The checkpoint consists of the __________ of the ________ in main memory, together with any lines of the cache which are _________ as part of the ________________.
- The cache ____________ has to have an extra ____________ ___, when set, that line is _____________.

6.4 Cache-Aided Rollback Error Recovery (CARER)

- We are forced to take a checkpoint when
  - A _____ _____ marked unmodifiable is to ___ ________
  - The _____ _________ is to be updated
  - An ____ ___________ is executed or an _______ ________ occurs
- Taking a checkpoint involves
  - Saving the __________ ________ in memory
  - Setting to 1 the __________ ____ associated with each _____ cache line
- The _____________ consists of the _________ of the __________ in the main memory, all the _____ ______ that are marked unmodifiable, and the _______ _______.
- Rolling back involves restoring the __________ from __________ and marking all cache lines with checkpoint bit 0 ________.

6.5 Checkpointing in Distributed Systems

- Logically, we consider a distributed system to consist of a _______ of _________ connected together by means of directional ________. Assume _________ are ___________.
- The ________ of a channel at time t is the ___ ___ ________ carried by the channel up to time t, ___ _____.
- The ________ of a distributed system is said to be _________ if, for every ________ _________ recorded in the state, there is a corresponding ___________ _______.
- A message that violates ___________ is said to be an _______.

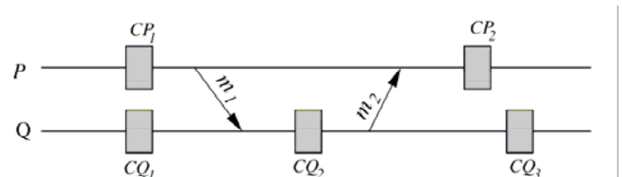


6.5 Checkpointing in Distributed Systems

- The following sets of checkpoints represent a _____________ system state:
  - $\{CP_1, CQ_1\}$: Neither knows about m
    - Action
  - $\{CP_2, CQ_1\}$: CP_2 knows that m was sent, CQ_1 hasn't received m
    - Action
  - $\{CP_2, CQ_2\}$: CP_2 knows that m was sent, CQ_2 has received m
    - Action
- The following checkpoint _____ ____ represent a consistent state.
  - $\{CP_1, CQ_2\}$: CP_1 knows nothing about m, CQ_2 has received m
- A set of checkpoints that represent a consistent system state is said to form a ________ ____, the system can be restarted from it.

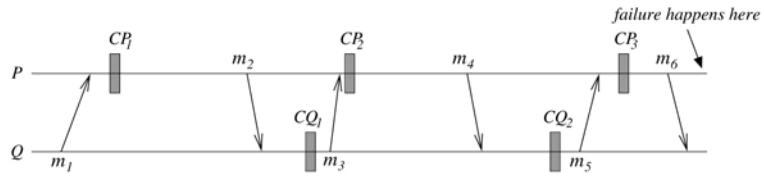
6.5 Checkpointing in Distributed Systems

- Sometimes checkpoints can be placed in such a way that they will _____ form part of a recovery line.
  - $\{CP_1, CQ_2\}$:
  - $\{CP_2, CQ_2\}$:



6.5.1 The Domino Effect and Livelock

- Without _______________, a single failure could cause a sequence of rollbacks that send _____ _______ back to its starting point, the __________ ______.

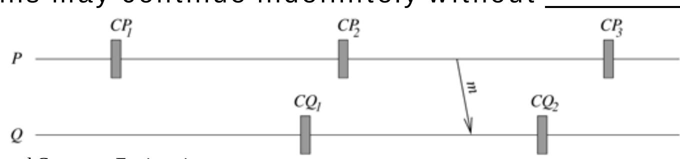


P suffers a transient failure

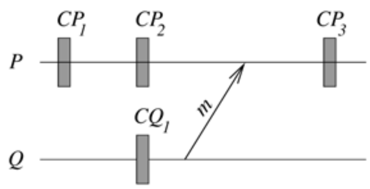
- Rolls back to checkpoint CP_3
- Q rolls back to CQ2 (so m_6 will not be an orphan)
- P rolls back to CP2 (so m_5 will not be an orphan)
- This continues until both processes have rolled back to their starting positions

6.5.1 The Domino Effect and Livelock

- Q sends P a message m_1; P sends Q a message m_2
- P fails _______ receiving m_1
- Q rolls back to CQ_1 (otherwise ____ is orphaned)
- P __________, rolls back to CP_2, sends another copy of ___, and then receives the copy of m_1 that was sent before all the rollbacks began
- Because Q has rolled back, this copy of m_1 is now ________, and P has to _____ its rollback
- This action orphans the second copy of m_2 and Q must repeat its rollback
- This may continue indefinitely without _____________



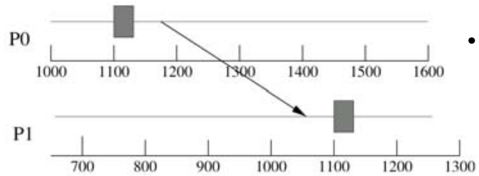
6.5.2 A Coordinated Checkpointing Algorithm



- Suppose that P wants to __________ __ _________ at CP_3 that will record, among other things, that message m was received from Q.
- To prevent m from ever being _________, Q must checkpoint also, thus _____________.

- Process P records its current state in a __________ checkpoint then sends messages to all other processes from whom it _______ __ _______ since taking its last checkpoint, with the id of the last message it received, m_{qp}. If others haven't recorded their message, they are asked to take a __________ checkpoint. If all checkpoint, the checkpoints become _____________. If not all checkpoint, the _________ checkpoints are _____________.

6.5.3 Time-Based Synchronization



- Orphan messages _______ _____ if each process checkpoints at exactly the ____ _______ time.

- Can't do it at the same time, try ________ _____ _____.
- Suppose the _________ skew between any two clocks in the distributed system is δ, and each process is asked to checkpoint when its local clock reads τ . Wait until _____ to send a message to make sure the other process has ____________. You can also include a lower bound on ______________ ______________ time, ε, and wait $\tau + \delta - \varepsilon$.
- Similarly, the process can receive a message but not ___ ___ __ if it _______ ________ an orphan.

6.5.4 Diskless Checkpointing

- Main memory is ________, and by itself, often unsuitable as a medium for storing a checkpoint, but writes to it are _____ _______ than to disks.
- Implement _______ checkpointing by having redundant processors using ___________ techniques to deal with failure.
- Have one extra processor hold ______ of all ______________ or have an extra processor per processor to _____ each _____________. Parity computations could also be _____________ to avoid ____ ______.

6.5.5 Message Logging

- __________ consists of rolling back to the ______ ___________ and taking up the computation from that point.
- In a ______________ system, the recovering process may require all the messages it __________ since that ________________, in __________ order.
- _________ ___________ is a consequence of coordinated checkpointing.
- An alternative is ______ ________ into stable storage.
  - ___________ Message Logging
  - _________ Message Logging

6.5.5 Message Logging - Pessimistic

- Pessimistic message logging _________ that rollback will not _______ to other processes
- The simplest approach is for the __________ of a message to log each message onto stable storage and then _______ ____________.
- Recovery consists of rolling back to the _________ __________ and playing the messages back __ ______.
- Logging all messages to stable storage is ___________overhead.
- Alternative is ____________ message logging.

6.5.5 Message Logging - Pessimistic

- The _______ of a message records it in a log (initially, a _______ _______).
- Each process has a _________ and a ___________.
- Each sent message has a Send Sequence Number (SSN), each received message has a Receive Sequence Number (RSN).
- The receiver sends an _______________, including the RSN and then sends no other messages until the sender _____________ the ____________________.
- A message is said to be ______ _______ when the sending node knows both its SSN and its RSN; and __________ logged when only the SSN is known.

6.5.5 Message Logging - Pessimistic

- When a process rolls back and restarts computation from the latest checkpoint, it sends out to the other processes a message listing the SSN of their latest message that it recorded in its checkpoint.
- When this message is received by a process, it knows which messages to __________.
- The recovering process now has to use these messages in the ____ _______ they were used before it failed.
- For ______ ________ messages, RSNs are available, sort by that.
- Because the receiver was forbidden from sending out any messages until acknowledgment was acknowledged, the order of _________ __________ messages cannot affect any other process.

6.5.5 Message Logging - Optimistic

- Optimistic message logging creates recovery scenarios sufficiently __________ as to render it of not much more than _______________ interest.
- When messages are ___________, they are written into a high-speed volatile buffer, that is copied into stable storage at a __________ ______, leading to low overhead.
- However, the buffer can be lost upon _________, leading to _________ processes being rolled back.
- A scheme is needed to compute the _____________ ______.

6.5.5 Message Logging – Staggered Checkpointing

- Many checkpointing algorithms can result in a _____ ________ of processes taking checkpoints at nearly the same time, leading to congestion at the ____ or ____________ or both.
- Two approaches
  - Write the checkpoints into a local buffer and stagger ______ ___ _____, requires large buffer.
  - _______ the checkpoints in time.
    - Ensure that at ___ __ process is taking its checkpoint
    - Have a ____________ _____ to avoid orphan messages

6.5.5 Message Logging – Staggered Checkpointing (Checkpointing Phase)

- The algorithm has two phases: checkpointing and message-logging

```
/* Checkpointing Phase */
for (i=0; i<=n-1; i++){
      P_i takes a checkpoint.
      P_i sends a message to P_(i+1) mod n, ordering the
      latter to take a checkpoint.
}
```

- At the end of the loop, P_0 gets a message from P_{n-1} ordering it to take a checkpoint, this is the cue for P_0 to initiate the second phase. It does this by sending out a marker message on each of its outgoing channels.
- When a process gets a marker message, it initiates the second phase.

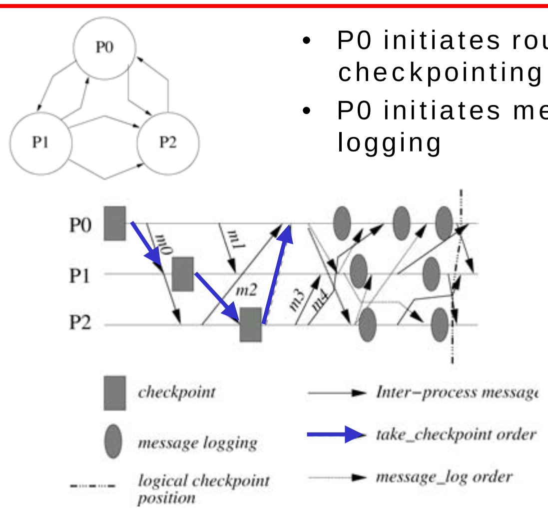
UAH Chapter 6 CPE 633

6.5.5 Message Logging – Staggered Checkpointing (Message Logging Phase)

```
/* Message Logging Phase */
if (no previous marker message was received in this round by Pi) then {
     Pi sends a marker message on each of its outgoing channels.
     Pi logs all messages received by it after the preceding checkpoint
     and before the marker was received.
}
else
     Pi updates its message log by adding all the messages received by it
     since the last message log and before the marker was received.
end if
```


UAH Chapter 6 CPE 633

6.5.5 Message Logging – Staggered Checkpointing (Example)



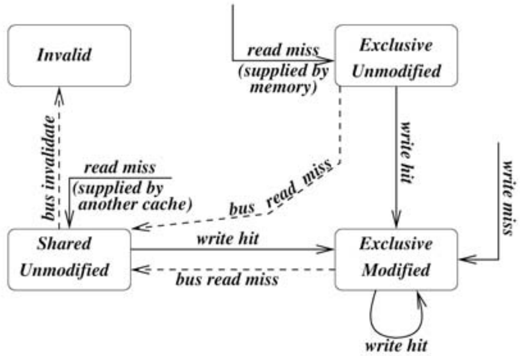
- P0 initiates round of checkpointing
- P0 initiates message logging

6.6 Checkpointing in Shared-Memory Systems

- Use a variant of the CARER scheme for ___________, ___________ multiprocessors with _______ caches.
- A checkpoint ___________ is associated with each cache line.
- A checkpoint ________ keeps track of the current checkpoint ________.
- Whenever a line is _________, we set $C_{id} = C_{count}$.
- If a line has been __________ since being brought into the cache and $C_{id} < C_{count}$, the line is part of the checkpoint state and is ____________.

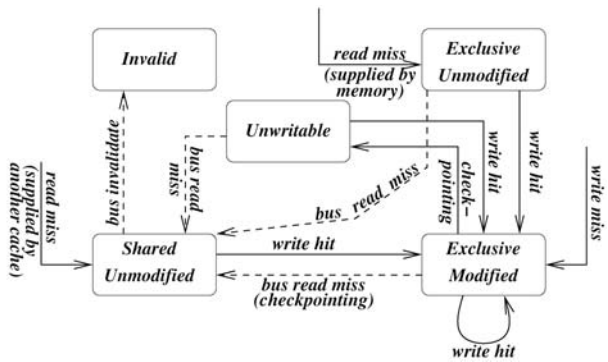
6.6.1 Bus-Based Coherence Protocol

- A cache line can be in one of the following states: ________, _______ ___________, ____________ ________, and __________ ____________



6.6.1 Bus-Based Coherence Protocol – Modified for Checkpointing

- The original __________ __________ state now splits into two: __________ __________ and ____________.



6.6.2 Directory-Based Protocol

- A ________ directory is maintained which records the status of each line, it is controlled by a ______________ controller.
- This controller handles all read and write misses and all other operations that ______ _____ _______.
- Caches ______ the controller of any ________ to be made.
- The __________ then changes the state in the ____________.
- _____________________ can be implemented atop such a _________.

6.7 Checkpointing in Real-Time Systems

- A real-time system is characterized by the need to meet ____________.The performance of a real-time system is related to the _____________ that the system will _________ its ___________ deadlines.
- Therefore, the goal of checkpointing in a real-time system is to __________ this ____________ and not to _____________ the mean execution time, in fact it may well ___________ the average execution time.
- We present an analytical model that calculates the ___________ _______ of the execution time of a task instead of the average execution time.
- We place a checkpoint after every T_{ex} units of ___________; each ones takes overhead. We are assuming here that checkpoint __________ and ___________ are identical, the system is so simple that the CPU has no other unit to which to delegate the checkpointing task.

6.7 Checkpointing in Real-Time Systems

- ____________ faults occur at a constant rate λ.
- When a transient failure hits the processor, it goes down for time T_r (including ____________ if necessary).
- Let $f_{int}(t)$ be the probability density function of the time taken between ____________ ______________ of checkpoints. There are two cases.
- Case 1 – no failure over the interval $T_{ex} + T_{ov}$
- Case 2 – at least one failure over the interval $T_{ex} + T_{ov}$
- When case 1 occurs, which it does with the probability $e^{-\lambda(T_{ex}+T_{ov})}$, the interval between checkpoint initiations will be $T_{ex} + T_{ov}$.
- In case 2, the time will be greater than $T_{ex} + T_{ov}$.

UAH Chapter 6 CPE 633

6.7 Checkpointing in Real-Time Systems

- To analyze Case 2, let us condition on the _____ _________, τ time units into the interval. At that time, τ units of computation ____ _____ and T_r units are taken to _________.
- Thus, the processor is ready to restart execution _______ units later.
- _____________ __ _________, the density function of the rest of the execution of this interval will be __________ to the unconditional density function.
- The conditional density function is $f_{int}(t - [\tau + Tr])$
- The probability of the first failure happening t units into the interval $[\tau, \tau + d\tau]$ is $\lambda e^{-\lambda\tau} d\tau$

$$f_{\text{int}}(t) = \int_{\tau=0}^{Tex+Tov} \lambda e^{-\lambda\tau} f_{\text{int}}(t - [\tau + Tr] d\tau \qquad \text{if } t > T_{ex} + T_{ov} + T_r$$

UAH Chapter 6 CPE 633

6.7 Checkpointing in Real-Time Systems

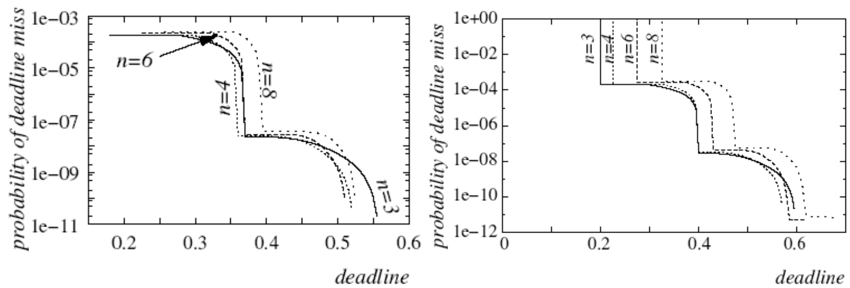
- Three cases for execution time, t
  - t can never be _____ than _________
  - t can never fall in the interval [_______, ____________]
  - t will be exactly __________ in the common case of __

$$f_{\text{int}}(t) = \begin{cases} e^{-\lambda(T_{ex}+T_{ov})}\delta(t - [T_{ex} + T_{ov}]) & \text{if } t = T_{ex} + T_{ov} \\ 0 & \text{if } t \neq T_{ex} + T_{ov} \text{ and } t \leq T_{ex} + T_{ov} + T_r \\ \int_{\tau=0}^{T_{ex}+T_{ov}} \lambda e^{-\lambda\tau} f_{\text{int}}(t - [\tau + T_r])d\tau & \text{if } t > T_{ex} + T_{ov} + T_r \end{cases}$$

- This equation can be solved numerically.
- For N checkpoints, the density function of the overall time is the (N+1)-fold ___________ of the density function per intercheckpoint interval: $f_{exec}(t) = f_{int}^{*(N+1)}(t)$
- If the real-time deadline is t_d, the probability of _______ ___ is given by:

6.7 Checkpointing in Real-Time Systems - Tradeoffs

- Consider a specific numerical example. Let T = 0.15 time units and $\lambda = 10^{-3}$ per time unit. The recovery time is Tr = 0.1 unit. The probability of missing a deadline is plotted for two cases: $T_{ov} = 0.015$ and $T_{ov} = 0.025$.



$T_{ov} = 0.015$ $T_{ov} = 0.025$

6.8 Other Uses for Checkpointing

- ________ __________
  - Since a checkpoint represents a ________ ______, ___________ a _________ simply involves ________ the checkpoint.
  - Used for __________ ____ _____ and ____ __________.
- __________
  - Provides the programmer with __________ of the __________.
  - These ___________ can be useful to study _______ __________.