CPE 633
Chapter 5 – Software Fault Tolerance

Dr. Rhonda Kay Gaede

UAH

Electrical and Computer Engineering

Introduction

- The difficulties in producing correct software are both _____ and _____.
- _____ difficulties include
  - Inherent complexity of a structure with _____ _____ and complex _____ _____
  - Frequency of _____ as ____ _____ _____ _____, _____ _____, _____ _____ _____
  - _____ between interacting _____ _____
- _____ difficulties include
  - People error in translating _____ _____ into _____ _____
- Even though much work has been and continues to be done, a _____ _____ is that any large piece of software currently in use _____ _____

5.1 Acceptance Tests

- Acceptance tests are used in _____ and _____ _____ and are essentially a check of _____.
- Categories of acceptance tests
  - Timing Checks – ___ _____ _____
  - Verification of Output – to save time, may restrict to _____ _____
  - Range Checks – must balance _____ and _____
    - _____ is the conditional probability that the test _____ ___ _____, given the output is _____.
    - _____ is the conditional probability that, given the acceptance test _____ ___ _____, it is _____ ___ _____.

5.2.1 Single-Version Fault Tolerance – Wrapper Introduction

- A wrapper is a piece of _____ that _____ the given program when it is being executed.
- ___ _____ of software can be wrapped, _____ and _____ are _____ by the wrapper.
- Wrappers are used in _____ applications when adapting _____ software components.
- Wrappers _____ _____ and perform _____ _____ on outputs.

5.2.1 Single-Version Fault Tolerance – Wrapper Examples

- Dealing with Buffer Overflow –
  - C does not perform _____ _____ for arrays, subject to _____ or _____ damage
  - A wrapper can perform the check
- Checking the Correctness of the Scheduler
  - For real-time scheduling, Earliest Deadline First may be the scheduling priority
  - Wrapper _____ that the _____ _____ is being executed correctly

5.2.1 Single-Version Fault Tolerance – Wrapper Examples Continued

- Using Software with Known Bugs
  - Software fails for a _____ __ __ _____
  - Wrapper checks _____ ____ __, may redirect the input to an _____ _____
- Using a Wrapper to Check for Correct Output
  - Wrapper includes an _____ _____

5.2.1 Single-Version Fault Tolerance – Wrapper Issues

- The ability to successfully wrap a piece of software depends on
  - _____ of the Acceptance Tests
  - _____ of Necessary Information from the Wrapped Component
    - Components are often _____ _____
    - _____ is the Holy Grail
    - _____ would really be nice
  - _____ to which the Wrapped Software Module Has Been _____
    - Testing identifies _____ _____ of the input space.

5.2.2 Software Rejuvenation

- _____ _____ ___ is an example of software rejuvenation
- Rejuvenation Level
  - _____ – Suspend individual application, Clean up its state by _____ _____, _____ of data structures, then _____
  - _____ – Reboot all applications.
    - In a _____, it is beneficial to _____ rejuvenations to maximize _____ _____ ____
- Timing of Rejuvenation based on
  - _____
  - _____

5.2.2 Software Rejuvenation - Time-based Rejuvenation

- Rejuvenate at _____ _____
- Want to balance _____ versus _____
- Mathematical model notation
  - N(t) - Expected number of errors over an interval of length t (without rejuvenation)
  - C_e – Cost of each error
  - C_r – Cost of each rejuvenation
  - P – Inter-rejuvenation period

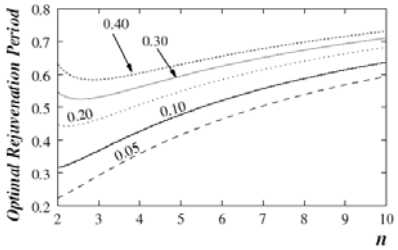
$$C_{rate}(P) = \frac{C_{rejuv}(P)}{P} = \frac{\tilde{N}(P)C_e + C_r}{P}$$

5.2.2 Software Rejuvenation - Time-based Rejuvenation

- Consider three cases of N(P)
- $N(P) = \lambda P$
- $N(P) = \lambda P^2$
- $N(P) = \lambda P^n$

5.2.2 Software Rejuvenation - Time-based Rejuvenation

- To set the period P appropriately, we need to know the values of Cr/Ce and N(t).
  - Obtain from _____
  - Make system _____, choosing default _____ _____, then _____ period



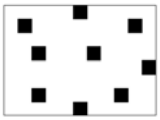
$\lambda = 1$; Time units are arbitrary;
Curve labels indicate C_r/C_e.

5.2.2 Software Rejuvenation - Prediction-based Rejuvenation

- Monitor the _____ _____ (amount of _____ _____, number of _____ _____ _____, etc.)
- Rejuvenate just _____ ___ _____
- The _____ must have access to enough _____ _____ to make such predictions
  - Part of _____ _____ – great
  - Built on top of ___ depends on ___ _____
    - vmstat, iostat, netstat, nfsstat
- Do _____ _____ or _____ _____ ___ of a polynomial
- Can combine _____ and _____ rejuvenation

5.2.3 Data Diversity

- The _____ _____ of a program can be divided into _____ and _____ regions
- _____ _____ typically have a large number of _____, but we can visualize them only in the _____ input space case.
- Data diversity is _____ ___ _____ slightly hoping to nudge it from a _____ region into a _____region.
- Consider the figures below, same _____ faulty



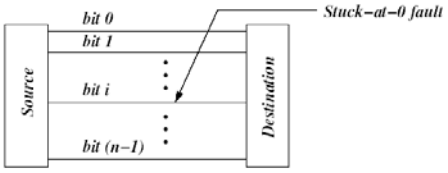
5.2.3 Data Diversity

- _____ of data diversity depends on the _____ mechanism
  - One copy with acceptance test – _____ with perturbed inputs and recheck _____ _____
  - Massive redundancy - apply _____ _____ input sets to different _____ of the program and then ____ on their output
- Perturbation of the input data can be done either
  - _____ – Add a small deviation
  - _____
    - _____ ___ _____ with inherent deviation – pressure and temperature
    - _____ _____
- The output may have to be _____.

5.2.4 Software Implemented Hardware Fault Tolerance (SIHFT)

- _____ _____ can be combined with ____ _____ to construct techniques for detecting _____ _____.
- A SIHFT can provide an _____ _____ to hardware and/or information redundancy especially on top of _____ _____.
- Suppose the program has all _____ _____ and _____.
- It can be _____ by multiplying ____ _____ and _____ by k and expecting the final result to be k times the _____ _____.
- k should result in a ____ _____ of detecting a fault and should not create _____ or _____, also $k=2^n$ is preferred

5.2.4 Software Implemented Hardware Fault Tolerance (SIHFT) - Example



- Consider the n-bit bus shown with the ith bit stuck-at-0.
- If a transformed program with ____ is executed on the same hardware, the ith bit will now use line ____ of the bus and will not be affected by the fault. The fault will not be detected if _____ ____ __ ____ _____ ___ __, that happens with probability 0.25
- If k=-1 is used, almost all 0s in the original program will turn into 1s, greatly reducing the probability of an _____ _____.

5.2.4 Software Implemented Hardware Fault Tolerance (SIHFT) - Overflow

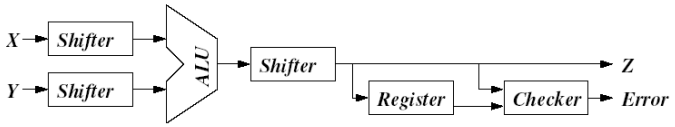
- The risk of _____ exists even for _____ values of k.
- Thus, the transformed program should take appropriate _____ by using range analysis to scale up the type of _____ _____

```
i = 0;
x = 3;
y = 1;
while (i < 5) {
    y = y * (x + i);
    i = i + 2;
}
z = y;
```

```
i = 0;
x = 6;
y = 2;
while (i < 10) {
    y = y * (x + i)/2;
    i = i + 4;
}
z := y;
```


5.2.4 Recomputing with Shifted Operands (RESO)

- This approach is similar to SIHFT, with the main _____ being that the hardware is _____ to support _____ _____.
- In this approach, each unit that executes either an _____ or a _____ operation is modified so that it first executes the operation on the _____ _____ and then re-executes the same operation on _____ _____.
- Here, the transformations are limited to _____ _____.
- The hardware can be to prevent ____.



5.3 N-Version Programming – Introduction

- N independent _____ __ _____ develop software to the same specifications.
- These N versions of software are then run in _____, and their output is _____ ____.
- The probability of no more than __ _____ versions out of __ _____, under the defect/bug _____ assumption, is

$$p_{ind}(N,m,q) = \sum_{i=0}^{m} \binom{N}{i} q^i (1-q)^{N-i}$$

5.3.1 Consistent Comparison Problem

- Consider N _____ _____ software versions, V_1, ...,V_N, for some _____.
- Suppose the _____ _____ of each version involves _____ some quantity, x, and _____ __ with a constant, c.
- Let x_i denote the value of x as computed by version V_i.
- The comparison is said to be _____ if either $x_i \geq c$ for all i=1,..,N, or $x_i < c$ for all i=1,...,N

5.3.1 Consistent Comparison Problem

- Consider an application
  - For f(p, t) < c take action A1
  - For f(p, t) ≥ c take action A2
- Consider concrete example, let _____ and _____
  - V_1 outputs _____, V_2 outputs _____ and V_3 outputs _____
  - _____
- As a result, _____ will order action A1 to be taken and _____ will order action A2 *even though all three versions are* _____ _____.

5.3.1 Consistent Comparison Problem

- If _____ versions can _____ in their output, the system has __ _____ __ determine whether the outputs are in disagreement because they are _____ or because of the _____ _____ _____.
- The _____ with which the consistent comparison problem _____ and the _____ __ _____ for which it lasts depend on the nature of the application.
- In applications where _____ _____ _____ is not used, the consistent comparison problem may occur _____ and go away _____ _____.

5.3.2 Version Independence

- _____ _____ between versions can increase the overall error probability by _____ ___ _____.
- Consider the case N=3, which can tolerate up to ___ _____ _____ for any input.
- Suppose that the probability of an incorrect output is _____, error probability is _____
- Now, suppose that they are not independent and there is one defect mode _____ ___ _____ ___ ___ _____ versions that occurs at a rate of 10^{-6}, the system error probability is now _____, more than 30 times that of the independent case.

5.3.2 Version Independence – More About Correlation

- Quite often, the _____ _____ can be subdivided into _____ according to the _____ that an input from that region will cause a version to fail. The error rate for that _____ may be greater than the _____ _____ _____ over the entire input space.
- If _____ in each subspace,
- Prob{V_1, V_2 both fail|input is from subspace S_i} = Prob{V_1 fails|input from S_i}•Prob{V_2 fails|input from S_i}
- Prob{V_j fails} = ΣProb{V_j fails|input is from S_i} • Prob{input is from S_i} (j=1,2)
- Prob{V_j, V_2 both fail} = ΣProb{V_1 fails|S_i} • Prob{V_2 fails|S_i} • Prob{input is from S_i}

5.3.2 Version Independence – Numerical Example

- The _____ failure probabilities are as follows

Version	S_1	S_2
V_1	0.010	0.001
V_2	0.020	0.003

- Prob{V_1 fails} =
- Prob{V_2 fails} =
- If independent,
  - Prob{V_1 fails} • Prob{V_1 fails} =
- The actual _____ _____ is
  - Prob{V_1, V_2 both fail} =
- The two versions are _____ correlated

5.3.2 Version Independence – Another Numerical Example

- The conditional failure probabilities are as follows

Version	S_1	S_2
V_1	0.010	0.001
V_2	0.003	0.020

- If independent,
  - Prob{V_1 fails} • Prob{V_1 fails} =
- The actual joint probability is
  - Prob{V_1, V_2 both fail} =
- The two versions are _____ correlated

5.3.2 Version Independence – Independence Compromising Factors

- _____ _____
- _____ _____ ___ ____ _____
  - one _____ of the input space may be particularly _____
- _____ _____
- _____ _____
  - programmers who are trained to _____ __ _____ ____ can make _____ _____
- _____ _____ ____ _____ _____
  - Faults/defects affect all versions

5.3.2 Version Independence – Independence Producing Factors

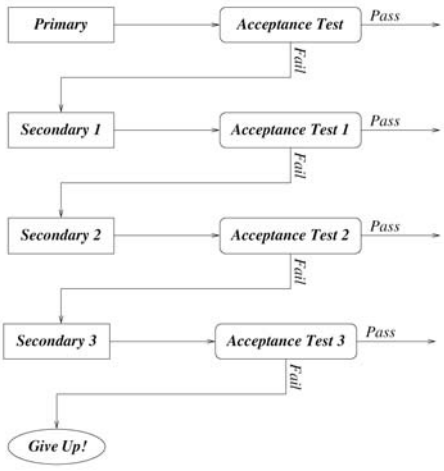
- _____ Diversity
  - The byproduct of _____ the developers of different modules to work _____ of one another, questions are sent to a _____ _____
- _____ Diversity
  - Use Diverse _____
    - The _____ may be expressed in _____ _____
    - One version may be simple enough to _____ _____
  - Use Diverse _____ _____
    - Errors with _____, not uncommon in C, will not occur in _____
    - Different _____, _____
  - Use Diverse _____ _____ and _____
  - Use _____ _____ Teams

5.3.2 Version Independence – Other Issues in N-Version Programming

- Back-to-Back Testing
  - Compare _____ for the same _____
  - Can also compare intermediate variables – increases _____, decreases _____
- Cost of N-Version Programming
  - Costs can be kept under control by carefully identifying the _____ _____ _____ of the code and only developing __ _____ for them
- Producing Single Good Versus Many Versions
  - No good _____ at this time
- Experimental Results
  - Available only for work done at _____
  - One done in same language, same hardware had ___ correlated bugs, expected no more than __
  - No correlation was observed between the _____ of the programs produced and the _____ of the programmer

5.4.1 Recovery Block Approach - Basic Principles

- Execute the _____ initially
- If the _____ _____ fails, roll back system state and execute _____
- If acceptance test 1 fails, roll back system state and execute _____
- If acceptance test 2 fails, roll back system state and execute _____
- If acceptance test 3 fails, _____ ___



5.4.2 Success Probability Calculation - Notation

- E the _____ of a version is _____
- T the ____ _____ that the output is _____
- f failure _____ of a version
- s test _____
- σ test _____
- n number of _____ software versions

- Thus, f = P{E}, s = P{T|E}, σ = P{E|T}
- For the scheme to succeed, it must succeed at some stage i, $1 \le i \le n$
  - Prob{Success in stage i} = $[P\{T\}]^{i-1}P\{E' \cap T'\}$

$$\Pr ob\{Success\} = \sum_{i=1}^{n} [P\{T\}]^{i-1} P\{\overline{E} \cap \overline{T}\}$$

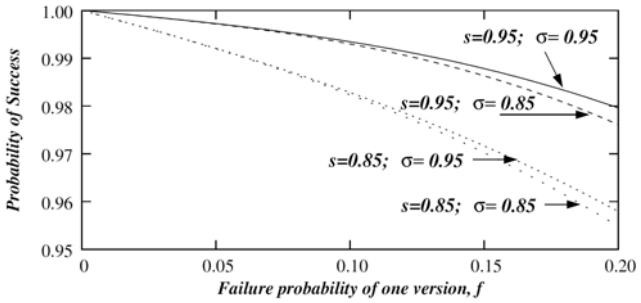
5.4.2 Success Probability Calculation - Formulation

$$\begin{aligned}
P\{E \cap T\} &= P\{T|E\}P\{E\} = sf \\
P\{T\} &= \frac{P\{E \cap T\}}{P\{E|T\}} = \frac{sf}{\sigma} \\
P\{\bar{E}|T\} &= 1 - P\{E|T\} = 1 - \sigma \\
P\{\bar{E} \cap T\} &= P\{\bar{E}|T\}P\{T\} = (1-\sigma)\frac{sf}{\sigma} \\
P\{\bar{E}\} &= 1 - P\{E\} = 1 - f \\
P\{\bar{E} \cap \bar{T}\} &= P\{\bar{E}\} - P\{\bar{E} \cap T\} = (1-f) - (1-\sigma)\frac{sf}{\sigma}
\end{aligned}$$

5.4.2 Success Probability Calculation - Example

$$\text{Prob\{Scheme is successful\}} = \sum_{i=1}^{n} \left[\frac{sf}{\sigma}\right]^{i-1} \left[(1-f)-(1-\sigma)\frac{sf}{\sigma}\right]$$

$$= \frac{1-\left(\frac{sf}{\sigma}\right)^{n}}{1-\frac{sf}{\sigma}} \left[(1-f)-(1-\sigma)\frac{sf}{\sigma}\right]$$



5.4.3 Distributed Recovery Blocks – Structure

- Consider the _____ ____ of only one secondary.
- Two nodes carry _____ _____ of the primary and secondary.
- Node 1 executes the _____ while node 2 executes the _____ in _____
- If node 1 _____ the acceptance test, the output of node 2 is used provided it ___ _____ the acceptance test
- The output of node 2 can also be used if there is a _____ _____ and node 1 fails to produce an output within a _____ time.
- Once the _____ ____ fails, the roles of primary and secondary are _____.
- No _____ is required, saving _____.
- Useful for _____ _____

5.5 Preconditions, Postconditions, and Assertions

- These are widely used in _____ _____ to improve _____ _____.
- A _____ of a method is a _____ _____ that must be true when that method __ _____.
  - Example – square root function over ___ _____, input must be _____
- A _____ associated with a method _____ is a condition that must be true when we _____ _____ the method
  - Example – $e^Y = X$ (within _____ _____) for _____ _____ method called with input X
- _____ are a generalization of _____ and _____, they test for a condition that must be true at the point at which that _____ is made

5.6 Exception-Handling

- An _____ is raised to indicate that something has happened during execution that _____ _____.
- When an _____ is raised, control is generally _____ to an _____ _____, a routine that takes the _____ _____.
- Effective _____ can make a _____ _____ to system fault tolerance.

5.6 Exception-Handling– Types of Exceptions

- _____ and _____ Error
  - A _____ error occurs when an _____ _____ is used
  - A _____ error occurs when the program _____ ___ _____ or _____ _____ __ _____ that is seen to be incorrect in some way.
    - Encountering EOF when reading a file
    - Producing a result that violates embedded acceptance test
    - Trying to print a line that is too long
    - Generating arithmetic overflow or underflow
- _____
  - Example, exhausted inputs from file
- _____
  - Missing a deadline in a real-time system

5.6.1 Requirements from Exception-Handlers - Characteristics

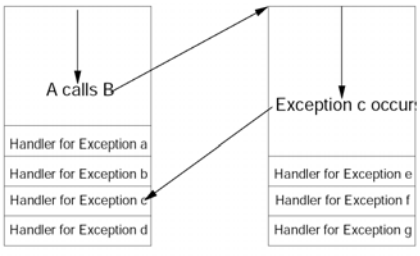
- Should be _____ to program and use, _____.
- Should not impose a _____ _____ on the normal functioning of the system – keep the _____ _____ fast.
- Must not compromise the _____ _____ – want to be able to resume.

5.6.2 Basics of Exceptions and Exception-Handling

- When an exception occurs, it is said to be _____, _____ or _____.
  - _____ – notifying the _____ _____ which it occurred
  - _____ – the notification _____ to another module
- Exceptions can be either
  - _____ – handled within the module in which it was raised
  - _____ – propagates elsewhere, i.e., incorrect module call.

5.6.2 Basics of Exceptions and Exception-Handling – Propagation of Exceptions

- Module A calls module B, which executes normally until it encounters _____ ___. B does not have the handler for this exception, so it propagates the exception back to its _____ _____, A, which executes the appropriate handler. If __ _____ can be found, the execution is _____.



5.7 Software Reliability Models

- The area of modeling error rates and software reliability is _____ _____ and _____ _____.
- Since software is often the _____ _____ of system unreliability, our _____ ___ _____ reliability causes concern.
- Three models given here that give _____ _____.
- We don't yet know which models are good for what _____ ___ _____.
- Distinguish between
  - _____ (bug) – exists in written software
  - _____ – deviation of _____ from exact requirements

5.7 Software Reliability Models – Definitions and Assumptions

- Definition of software reliability – the probability of _____ _____ of a computer program in a _____ _____ for a _____ _____.
- Need to introduce _____ _____ ____.
- Software reliability models attempt to predict the software error rate as a function of the _____ ___ _____ ___ ___ _____, and their purpose is to determine the _____ ___ _____ required until the predicted future error rate of the software goes below some _____ _____.
- Common assumptions
  - _____ _____ number of bugs
  - When a bug occurs during testing, it is fixed in _____ _____ without causing any ___ _____

5.7.1 Jelinski-Moranda Model

- Assumes _____ number of bugs, N(0), at time 0, out of which ____ bugs remain at time t
- The _____ _____ is a Poisson process with a rate $\lambda(t)$ that _____ with time, $\lambda(t) = cN(t)$
- $\lambda(t)$ _____ ___ ___ whenever an error occurs and is _____ between errors
- For _____ _____ testing time between consecutive errors, the reliability at time t is $R(t) = e^{-\lambda_0 t}$
- Given an error occurred at time τ, the _____ _____ that the following interval of length t will be _____ is $R(t|\tau) = e^{-\lambda(\tau)t}$
- As the software runs for longer and longer, more bugs are caught and _____ from the system, and so the error rate _____ and the future reliability _____
- The _____ to this model is that all bugs _____ _____ to the error rate.

5.7.2 Littlewood-Verrall Model

- Again, assume ____ and _____ number, N(0) of initial bugs, out of which ___ remain at time t.
- Difference, consider M(t) - the number of bugs _____ ___ _____ during [0,t] - M(t) = N(0) – N(t)
- The errors occur according to a nonhomogeneous Poisson process with $\lambda(t)$ considered a _____ _____ with a Gamma density function. The Gamma density function has two parameters α and ψ, where the parameter ψ is a _____ _____ function of M(t)

$$f_{\lambda(t)}(\ell) = \frac{[\psi(M(t))]^{\alpha} \ell^{\alpha-1} e^{-\psi(M(t))\ell}}{\Gamma(\alpha)}$$

where

$$\Gamma(x) = \int_0^{\infty} e^{-y} y^{x-1} dy$$

5.7.2 Littlewood-Verrall Model - Reliability

- The Gamma density function was chosen because it lends itself to _____ and its two parameters provide a _____ _____ of differently shaped density functions.
- The _____ _____ of the Gamma density function is $\alpha/(\psi(M(t))$, so that the predicted error rate will _____ and the reliability _____ as the software is run for longer periods of time and _____ _____ ___ _____.
- Reliability

$$R(t) = \left(1 + \frac{t}{\psi(0)}\right)^{-\alpha}$$

$$R(t \mid \tau) = \left(1 + \frac{t}{\psi(M(\tau))}\right)^{-\alpha}$$

5.7.3 Musa-Okumoto Model

- This model assumes an _____ number of initial bugs in the software, and uses M(t).
- Notation – λ_0 – error rate at time 0, c – constant of proportionality, $\mu(t)$ – the _____ _____ of errors experienced during [0,t] ($\mu(t) = E(M(t))$)
- The _____ ___ after testing for a time t is $\lambda(t) = \lambda_0 e^{-c\mu(t)}$
- Intuitively, when testing first starts, the _____ bugs are caught quickly. The remaining bugs are _____ _____ to catch. As a result, the rate at which an as-yet-undiscovered bug causes errors drops _____ as testing proceeds.

5.7.3 Musa-Okumoto Model - Reliability

- Starting with the differential equation

$$\frac{d\mu(t)}{dt} = \lambda(t) = \lambda_0 e^{-c\mu(t)}$$

- We get

$$\mu(t) = \frac{\ln(\lambda_0 ct + 1)}{c} \qquad \lambda(t) = \frac{\lambda_0}{\lambda_0 ct + 1}$$

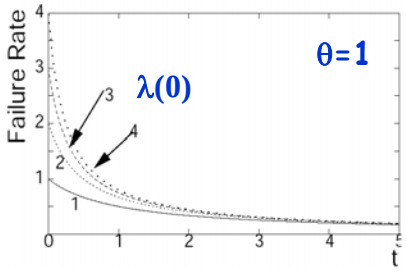
- Giving a reliability

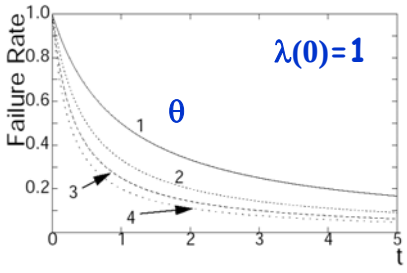
$$R(t) = e^{-\int_0^t \lambda(z)dz} = e^{-\mu(t)} = (1 + \lambda_0 ct)^{-\frac{1}{c}}$$

$$R(t \mid \tau) = e^{-\int_\tau^{\tau+t} \lambda(z)dz} = e^{-(\mu(\tau+t)-\mu(\tau))} = \left(1 + \frac{\lambda_0 ct}{1 + \lambda_0 c\tau}\right)^{-\frac{1}{c}}$$

5.7.3 Musa-Okumoto Model - Error Rates

- The error rate decays _____ _____
- _____ _____ of testing are required





5.7.4 Model Selection and Parameter Estimation

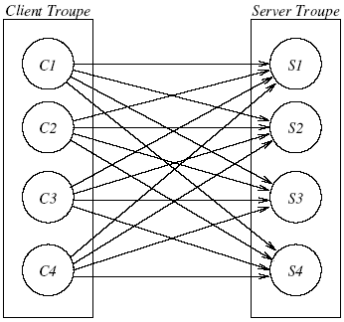
- The literature on software error models is vast and varied, a small subset has been presented.
- If using a model, two questions
  - _____ _____
    - The American Institute of Aeronautics and Astronautics (AIAA) recommends using one of the three mentioned here or the _____ model
    - _____ hardware reliability models, __ _____ _____ data is available
    - The best that one can suggest is to ____ ___ _____ ____ and _____ which model it follows
  - _____ _____
    - After model selection, can estimate parameters using the _____ _____ _____

5.8.1 Fault-Tolerant Remote Procedure Calls - Primary-Backup Approach

- A Remote Procedure Call (___) is a _____ by which one process can __ _____ _____ executing on some other processor.
- Each process is implemented as _____ and _____ processes, running on _____ _____.
- _____ are sent to both copies, _____ primary executes them, secondary is _____ should primary ____
- Two flavors
  - _____– a _____ RPC can be executed _____ _____ without violating correctness (____ __ _____)
  - _____ – should be completed _____ ____ (any bank transaction other than _____), may use _____ (Chapter 6)

5.8.2 Fault-Tolerant Remote Procedure Calls – The Circus Approach

- The circus approach involves _____client and server processes, have client and server _____.
- Consider four clients and four servers, each client makes a call to _____, each call has a _____ _____
- A server waits until it has _____ ____ _____ ____ or _____ ____ before executing the RPC, then sends results with sequence number to ___ _____
- A client may wait until _____ _____ replies or _____ ____ before accepting the input or take _____, _____ ____



5.8.2 Fault-Tolerant Remote Procedure Calls – Ensuring Order

- Complication – multiple client troupes sending _____ ____ to the server troupe
- Optimistic Approach – Let everything run freely and _____ to see whether _____ _____, if not, abort and try again
- Pessimistic Approach – _____ _____ ensure that order is preserved.