Report Number: WUCS-92-07

1992

Multicomputer Checkpointing

Ken Wong and Mark Franklin

This paper examines the performance of synchronous checkpointing in a distributed computing environment with and without load redistribution. Performance models are developed, and optimum checkpoint intervals are determined. We extend earlier work by allowing for multiple nodes, state dependent checkpoint intervals, and a performance metric which is coupled with failure-free performance. We show that the optimum checkpoint intervals in the presence of load redestribution has a numerical solution in all cases and a closed form in many reasonable cases. These new results are then used to determine when performance can benefit load redistribution.
... Read complete abstract on page 2.

Multicomputer Checkpointing

Ken Wong and Mark Franklin

Complete Abstract:

This paper examines the performance of synchronous checkpointing in a distributed computing environment with and without load redistribution. Performance models are developed, and optimum checkpoint intervals are determined. We extend earlier work by allowing for multiple nodes, state dependent checkpoint intervals, and a performance metric which is coupled with failure-free performance. We show that the optimum checkpoint intervals in the presence of load redestribution has a numerical solution in all cases and a closed form in many reasonable cases. These new results are then used to determine when performance can benefit load redistribution.

Multicomputer Checkpointing

Ken Wong
Mark Franklin

WUCS-92-7

January 1992

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis, MO 63130-4899

Multicomputer Checkpointing

Ken Wong and Mark Franklin

The Computer and Communications Research Center
Washington University
One Brookings Drive, Campus Box 1115
St. Louis, MO 63130-4899
kenw@wuccrc.wustl.edu, jbf@wuccrc.wustl.edu

Submitted to

Eleventh Annual ACM Symposium on

Principles of Distributed Computing

Abstract

This paper examines the performance of synchronous checkpointing in a distributed computing environment with and without load redistribution. Performance models are developed, and optimum checkpoint intervals are determined. We extend earlier work by allowing for multiple nodes, state dependent checkpoint intervals, and a performance metric which is coupled with failure-free performance. We show that the optimum checkpoint intervals in the presence of load redistribution has a numerical solution in all cases and a closed form in many reasonable cases. These new results are then used to determine when performance can benefit from load redistribution.

(EXTENDED ABSTRACT)

Optimum Multicomputer Checkpointing*

Ken Wong and Mark Franklin

The Computer and Communications Research Center
Washington University, St. Louis, MO 63130
kenw@wuccrc.wustl.edu, jbf@wuccrc.wustl.edu

1. Introduction

The emerging technologies of gigabit networks [1] and very high-speed processors, suggest the possibility for tackling large, computationally difficult problems by coupling these technologies into a somewhat ad hoc distributed computing environment. The applications which make good candidates for this environment will produce results only after many hours even when multiple computers are employed. A fundamental problem which must be addressed in this environment is providing effective computational progress in the face of resource failures.

In order to achieve maximum speed, the computational tasks must be assigned to the resources to exploit maximum parallelism. However, the possibility for a system failure (and therefore a complete restart) increases as larger numbers of processors are brought to bear on the application. Thus, fault tolerant techniques must be used to insure finishing times which are comparable with fault-free performance.

One approach to providing higher reliability is to periodically checkpoint or save the system state. When a failure occurs, the computation returns to the latest checkpoint. The state of the system at this latest checkpoint is recovered from checkpoint storage and the computation then proceeds forward. However, the checkpoint frequency must be chosen wisely. Checkpointing frequently in a highly reliable system results in unnecessary overhead while checkpointing infrequently in a highly unreliable system results in large quantities of lost work which must be repeated. Another approach is to isolate the faulty processor by redistributing the load onto the operational processors. When the faulty processor is repaired, the load is redistributed back onto the fixed processor. This approach allows for graceful performance degradation in the face of failures.

---

* This research has been sponsored in part by funding from the NSF under Grant CCR-9021041.

This paper examines the performance of synchronous checkpointing in a distributed computing environment with and without load redistribution. Performance models are developed, and optimum checkpoint intervals are determined. We extend earlier work by allowing for multiple nodes, state dependent checkpoint intervals, and a performance metric which is coupled with failure-free performance. We show that the optimum checkpoint intervals in the presence of load redistribution has a numerical solution in all cases and a closed form in many reasonable cases. These new results are then used to determine when performance can benefit from load redistribution.

2. The Optimum Checkpointing Problem

Optimum checkpointing for the single-node case has been studied extensively [2,3,4,5,6,7,8]. Optimum checkpoint intervals have been found by maximizing availability or minimizing response time. The objective function is typically convex and analytic or numerical solutions can be found in many cases [8]. Gelenbe has shown that the optimum checkpoint interval which minimizes the response time must be smaller than the one that maximizes the availability in order to avoid queue build-up due to processor unavailability [4]. In most cases, a transaction-oriented environment is assumed where jobs or requests arrive from a Poisson source.

The optimum selection of checkpoint intervals for the multicomputer case has been sparsely studied. Gelenbe, et. al., developed a model which included the overhead of fault detection [9]. Gelenbe assumed multiple nodes, the existence of a testing algorithm that periodically determined which nodes were operational, and an external source of requests. Requests sent to faulty nodes are routed to operational ones. Thus, the mean input rate to a node is a function of the job source rate, the number of faulty nodes, and the precision of fault detection. An expression for the optimum checkpoint interval and testing interval involving an unknown mean input rate were obtained for symmetric systems in which all nodes have identical configurations and load. Then, the optimum solution was determined by iteratively applying this expression and the flow conservation law.

Our models consider a related situation but with the following differences:

1) Jobs are generated internally as the result of other jobs rather than coming from an external Poisson source.
2) Synchronous checkpointing is employed rather than an asynchronous checkpointing algorithm.

3) Fault detection is not explicitly modeled.

Our motivation is driven by a desire to model applications in a scientific computation environment rather than a transaction-oriented one. These differences lead to different models, solution techniques, and results.

We begin by considering the single-node case and then extend this to multiple nodes. A node can be viewed as being in one of three states: A) available, C) checkpointing, or R) recovering. Recovery involves restoring the latest state and repeating work lost since the latest checkpoint. For mathematical tractability and simplicity, we assume Markovian state occupancy times and that failures only occur when a node is in the available state. This is reasonable when checkpoint and recovery times are small compared to the time that the node is available. For the parameter definitions in Table I, the state transition-rate diagram for this Markov process is shown in Figure 1. We can solve for the availability (steady-state probability of being in the available state) π_A using standard CTMC (continuous-time Markov chain) techniques.

$$\pi_A = \left[1 + \frac{\alpha}{\beta} + \frac{\phi}{\delta}\right]^{-1} \tag{1}$$

Note that the mean recovery rate δ is a function of the mean checkpoint rate α since the amount of work to be repeated after a failure is related to the time between checkpoints.

Parameter	Description
ϕ^{-1}	mean failure time
α^{-1}	mean time between checkpoints
β^{-1}	mean time to perform a checkpoint
δ^{-1}	mean recovery time

Table I. Model Parameters.

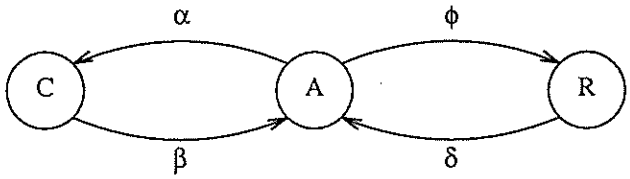

Figure 1. Single-Node State Transition Rate Diagram.

We assume that the mean recovery time is a fraction of the busy time between checkpoints plus a restore time. Suppose we observe the node in the time interval [0,T]. The node spends T_A seconds in the available state. While in the available state, the node can be idle (e.g., waiting for I/O) or busy processing jobs. The node performs n_C checkpoints and is busy processing jobs for T_B ($T_B \leq T_A$) seconds. We would like to determine the quantity T_B/n_C (the time between checkpoints in which the node is busy) since some fraction of this computation time would be lost from a failure.

Suppose that we know the fault-free utilization U which is the fraction of time spent processing jobs in a system that never fails and has no checkpoint/recovery overheads. We assume that the faulty system looks like the fault-free system when it is in the available state. So, the node is busy processing jobs for

$$T_B = UT_A \tag{2}$$

seconds in the interval [0,T]. By definition, the mean checkpoint rate α is equal to

$$\alpha = \frac{n_C}{T_A} \tag{3}$$

Using the above equations, the intercheckpoint busy time we desire can now be written as

$$\frac{T_B}{n_C} = \frac{UT_A}{\alpha T_A} = \frac{U}{\alpha} \tag{4}$$

We assume that the recovery time is a fraction k_δ of this quantity plus a restore time d:

$$\delta^{-1} = \frac{k_\delta U}{\alpha} + d \tag{5}$$

Now, the availability (the probability of the node being available) can be written as:

$$\pi_A = \left[+ \frac{\alpha}{\beta} + \phi \left[\frac{k_\delta U}{\alpha} + d \right] \right]^{-1} \tag{6}$$

Setting the derivative of π_A with respect to α to 0 and solving for α leads us to the optimum checkpoint rate:

$$\alpha^* = \sqrt{\phi \beta k_\delta U} \tag{7}$$

This corresponds to the result found in the literature [2,3]. The optimum checkpoint rate α^* behaves as expected. It should increase as the failure rate ϕ increases and decrease as the checkpointing time β^{-1} (rate β) increases (decreases). The checkpointing rate should increase as the fault-free utilization U increases; i.e., the loading of the node increases.

At first, it may seem strange that the optimum checkpoint rate does not depend upon the recovery parameter d which captures the repair and reload time. Although variations in d do not change α, they do affect the probability of being in the available state π_A. Since the checkpoint rate α is defined relative to the time spent in the available state ($\alpha = n_C/T_A$), the checkpoint rate defined over all time ($\alpha\pi_A$) is a function of this parameter d.

The multi-node case offers us the opportunity to choose between several recovery methods. We consider two synchronous checkpoint recovery methods: one with load redistribution and one without load redistribution. Typically, if the down time after a failure is short, it is reasonable to wait for the failed node to recover before continuing the computation. However, if the failed node will be unavailable for a significant amount of time, it may make sense to redistribute the load among the remaining nodes, repeat the lost work, and then continue.

We make the following assumptions about each node:

1) State occupancy times are Markovian.
2) Failures form a Poisson process.
3) Failures only occur in the available state.
4) The fault-free, relative speed-up curve is known.

The first assumption is for mathematical tractability. The second assumption is a typical reliability assumption. The third assumption is reasonable when the checkpoint and recovery times are small compared to the time spent in the available state. And the fourth assumption is reasonable for many scientific computations. Relative speed-up in this context is defined to be the ratio of the multi-node computation rate to the single-node rate. For example, for certain scientific applications, after certain number of processors have been utilized, the parallelism inherent in the solution algorithm is exhausted. This results in a curve of the form shown in Figure 2.

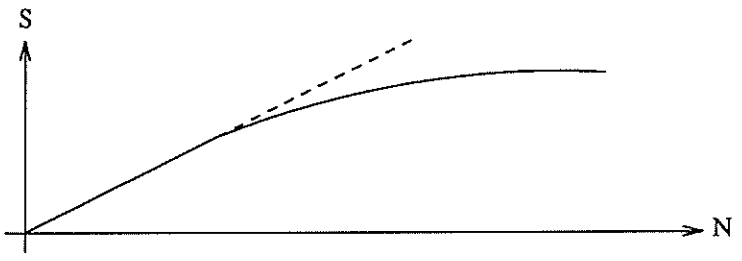

Figure 2. Speed-Up as a Function of Number of Nodes.

Numerical solution techniques can be employed to find the optimum checkpoint rate. For illustrative purposes in this paper, we assume that the nodes are symmetric. Equivalently, the interfailure time, inter-checkpoint time, and checkpoint duration at any node are stochastically identical to those on any other node. Although we make this last assumption for analytic tractability, the basic development would remain the same even without this last assumption, although the resulting equations would be more complex.

3. Model I — Synchronous Checkpointing Without Load Redistribution

Begin by considering the simplest case of synchronous checkpointing without load redistribution. The disadvantages of this approach are the high synchronization cost and low availability under even moderate failure rates in systems with many nodes. In this approach, all nodes checkpoint at approximately the same time. A two-phase commit protocol might be used to synchronize the start of a checkpointing phase. Once the checkpointing is done the system is available for normal computation. If any node fails, the whole system must go through a recovery phase. After recovery, the system is again operational with N nodes.

The system looks like a single-node running N times as fast when operational, but also failing N times as often. The system can be in one of three states: A) all nodes are available, C) all nodes are checkpointing, or R) all nodes are recovering. The availability π_A of this system can be obtained from the single-node case by replacing the mean failure rate ϕ by $N\phi$, the mean system failure rate. Furthermore, the single-node, fault-free utilization is now state-dependent and equal to U(N) when N nodes are operational. Using Equation 6, the availability is now given by

$$\pi_A = \left[1 + \frac{\alpha}{\beta} + N\phi\left[\frac{k_\delta U(N)}{\alpha} + d\right]\right]^{-1}$$

The objective is to maximize the computational rate of the system. We assume that each node computes by reading messages, computing results based on its input messages, and then perhaps sending its results to another node. Consider for the moment a single node in the time interval [0,T] which spends on average $\mu^{-1} = T_B/n_J$ seconds servicing each job where T_B is the total job processing time and n_J is the total number of jobs processed. Then the computation rate is just the mean output rate $\lambda_1 = n_J/T$. From our operational definitions in the single-node case, λ_1 can be written as:

$$\lambda_1 = \frac{n_J}{T} = \frac{n_J}{T_B}\left[\frac{T_B}{T_A}\right]\left[\frac{T_A}{T}\right] = \mu U(N)\pi_A \tag{8}$$

where the fault-free utilization U(N) is defined to be equal to T_B/T_A. Since we have assumed homogeneous node behavior, the computation rate is just N times the single-node output rate or

$$r = N\lambda_1 = N\mu U(N)\pi_A \tag{9}$$

We take the same approach in optimizing the computation rate here as we did in optimizing the availability in the single-node case since r is a linear function of π_A. We take the derivative of r with respect to α and set the derivative equal to 0 to determine the value of α which maximizes the computation rate. The optimum checkpoint rate is given by

$$\alpha^* = \sqrt{N\phi\beta k_\delta U(N)} \tag{10}$$

If load balancing is fairly uniform and the nodes exhibit homogeneous behavior, the fault-free utilization is related to the fault-free, relative speed-up S(N) by

$$S(N) = NU(N) \tag{11}$$

That is, in the fault-free environment, a maximum speed-up of N is achieved when all processors are at 100% utilization and between 0 and N otherwise*. Then, the optimum checkpoint rate can also be written as

$$\alpha^* = \sqrt{S(N)\phi\beta k_\delta} \tag{12}$$

If the fault-free system has linear speed-up (S(N) = cN), the optimum checkpoint rate is approximately $\sqrt{N}$ times higher than the single-node case because of the increase in the probability of system failure. However, the speed-up S(N) typically becomes less than linear with increasing number of nodes because of increases in synchronization and communication costs.

Consider for the moment the linear speed-up case. The mean computation rate at the optimum checkpoint rate can be written as

$$r = \frac{Nc\mu}{1 + \sqrt{N\phi k_\delta c/\beta} + N\phi d} \tag{13}$$

From this equation, we see that as the number of nodes N is increased, the mean computation rate increases by a factor of N but also decreases by a factor in the denominator that is related to the overhead associated with checkpoint and recovery. Careful control of these overheads will be necessary to achieve a

* We ignore for now the possibility for super-linear speed-up.

performance level that justifies the use of large processor populations.

4. Model II — Synchronous Checkpointing With Load Redistribution

In synchronous checkpointing without load redistribution, a large recovery time due to hard errors which can not be resolved by quick resets forces the entire system to be unproductive for a large time period. In this situation, it would make more sense to work around the faulty node and redistribute the load onto operational nodes. But typically, the cost of load redistribution is significant. If the faulty node becomes operational immediately after the load is redistributed, it would have been better to wait for the faulty node to be repaired and not redistribute the load. This section quantifies the overheads that justify a load redistribution and determines the optimum checkpoint intervals. Furthermore, the checkpoint interval is allowed to be dependent on the number of operational nodes.

For homogeneous nodes, the system state can be defined in terms of the vector $S = (n_A,n_C,n_R,n_F)$ where n_A is the number of nodes in the available state, n_C is the number of nodes in the checkpointing state, n_R is the number of nodes in the load redistribution state corresponding to downsizing, and n_F is the number of failed nodes. The nodes in the failed state must be repaired before going through a load redistribution (upsizing) phase. The state transition rate diagram is shown in Figure 3. Consider the available state (N–k,0,0,k) in the middle column in Figure 3 in which N–k nodes are operational and k nodes are being repaired. The state (0,N–k,0,k) (right column) represents N–k nodes checkpointing while k nodes are being repaired. Checkpoints are performed at a rate of α_{N-k} and returned to the available state at a rate of β. Since there are N–k operational nodes each failing at a rate of ϕ, a failure causes a redistribution of the load down to N–k–1 operational nodes at a rate of $(N-k)\phi$ to the state (0,0,N–k–1,k+1) (left column). If a node is repaired, the load is redistributed among the N–k+1 operational nodes, and some lost work is repeated. We assume that this occurs at a rate of γ_{N-k}.

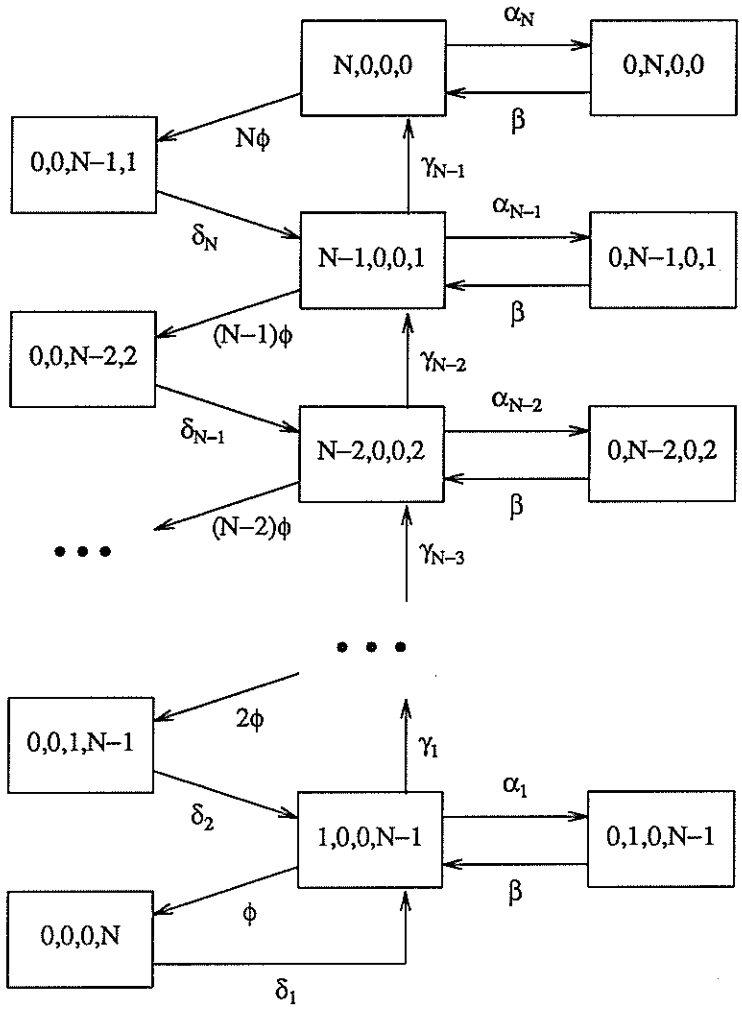

Figure 3. State Transition Rate Diagram
(Synchronous Checkpointing With Load Redistribution)

The mean rates ϕ and β have the same interpretation as in the single-node case. The mean rate α_k, k=1,...,N, is the multiprocessor equivalent to α and are the mean checkpoint rates when there are k operational nodes. The two rates δ_k, k=1,...,N, and γ_k, k=1,...,N−1, are the recovery rates associated with downsizing (omitting a failed node) and upsizing (including a recovered node) respectively. Both are functions of α_k and have a form similar to the mean recovery time expression in Model I.

$$\gamma_k^{-1} = \frac{k_\gamma U(k)}{\alpha_k} + g_k, \qquad k=1,...,N \tag{14}$$

$$\delta_k^{-1} = \frac{k_\delta U(k)}{\alpha_k} + d_k, \qquad k=1,...,N \tag{15}$$

The parameters d_k, k=1,...,N, represent the mean time to redistribute the load to k–1 nodes, and restore the node states. The parameters g_k, k=1,...,N, represent the mean time to repair one of the faulty nodes, redistribute the load to k+1 nodes, and restore the node states. Note that these parameters are state dependent. The remainder of the expressions represent the recomputation time required to return the node to the point at which it failed. Appendix I shows how the state probabilities can be computed using local balance equations and probability conservation. All state probabilities are expressed in terms of $\pi_{N,0,0,0}$.

The mean computation rate is the sum of the mean computation rates in each available state (k,0,0,N–k). From symmetry,

$$r = \sum_{k=1}^{N} k\lambda_k = \sum_{k=1}^{N} k(U(k)\pi_{k,0,0,N-k}\mu)$$

where λ_k is the mean computation rate when k nodes are available, and U(k) is the fault-free, single-node utilization when k nodes are operational. Appendix I shows that this computation rate can be written as

$$r = \mu\pi_{N,0,0,0}\sum_{k=1}^{N} U(k)\frac{N!}{(k-1)!}\frac{\phi^{N-k}}{\Gamma_k} \tag{16}$$

where

$$\Gamma_k = \begin{cases} \gamma_k \cdots \gamma_{N-1}, & k=1,...,N-1 \\ 1, & k=N \end{cases} \tag{17}$$

The optimum checkpoint interval when there are N operational nodes α_N can be shown to be identical to that of a system of N nodes without load redistribution after failure; that is,

$$\alpha_N^* = \sqrt{N\phi\beta k_\delta U(N)} = \sqrt{S(N)\phi\beta k_\delta} \tag{18}$$

However, the other optimum checkpoint intervals α_k, k=1,...,N–1, must in general be solved numerically. All Γ_i, i=1,...,k, are functions of α_k, and even the case N=2 is difficult to find a closed form expression for α_1.

Optimum checkpoint rates can be found symbolically for ranges of γ_k such that the recomputation time component (the one involving α_i) is small compared to the repair, load redistribution, and checkpoint restoration time components (g_k); that is, when there may be several checkpoints before the node is repaired. The optimum checkpoint rates α_k in this case turn out to be identical to the situation when there are k nodes and no load redistribution; that is,

$$\alpha_k^* = \sqrt{k\phi\beta k_\delta U(k)} = \sqrt{S(k)\phi\beta k_\delta}, \qquad k=1,...,N-1 \tag{19}$$

Numerical solutions verify this result. Upon preliminary examination of optimum checkpoint rates, this expression seems to be a lower bound on the optimum rates even when the assumption of small recomputation to recovery time does not hold.

5. An Example

This section presents an example of how the models can be used to choose the proper recovery algorithm. Earlier we conjectured that load redistribution could be beneficial if the cost of redistribution were justified in an increase in availability (and therefore output rate). There should be some range of failure rate and redistribution cost in which one algorithm is preferred over the other. Table II shows the parameter values of four models. In cases Ia, and Ib, the load is not redistributed after a failure. In cases IIa, and IIb, the load is redistributed after a failure. The difference between the *a* and *b* cases is that the *b* case has a higher repair time than the *a* case. All other parameters not shown in the table are assumed to be equal to 1. For example, we have assumed that the fault-free utilization, and the constants k_δ and k_γ are 1. Note that in model I the parameter d_k plays almost the same role that g_k plays in model II since the repair time in model I is included in d_k whereas it is included in g_k in model II. For simplicity, we have assumed in this example that the load redistribution time is negligible compared to the repair time in model II. If the redistribution time were significant compared to the repair time, g_k in model II would be larger than d_k in the corresponding model I.

Parameter	Model			
	Ia	IIa	Ib	IIb
N	4	4	4	4
β	1	1	1	1
g_k	na	1,000	na	10,000
d_k	1,000	10	10,000	10
μ	1	1	1	1
α	Optimum			
ϕ	Varied			

Table II. Model Parameters.

Figure 4 shows the effect of the the mean failure rate ϕ on the mean output rate r for the two cases Ia and IIa. Figure 5 shows the effect of the the mean failure rate ϕ on the mean output rate r for the two cases Ib and IIb. All sets of curves (Ia-IIa and Ib-IIb) have crossover points: $\phi \approx 0.004$ and $\phi \approx 0.0002$. The cross-

over points indicate that load distribution is justified in case a and b only when the failure rate is greater than 0.004 and 0.0002 respectively. All curves show that the performance degrades with increasing failure rate. The output rate curves associated with model I (no load redistribution) start at a higher value than the ones for model II, but rapidly decrease until the output rate is less than the output rate for model II. Curves for Ib and IIb are lower than the ones for Ia and IIa since the mean repair times (g_k) are higher for these two cases than those for Ia and IIa. If the checkpoint/recovery overheads and failures were negligible, the maximum output rate would be 4 since each of the four nodes would be outputting at a rate of $\mu^{-1}=1$. The overheads assumed in this example have reduced the output rate to substantially less 4 for large failure rates.

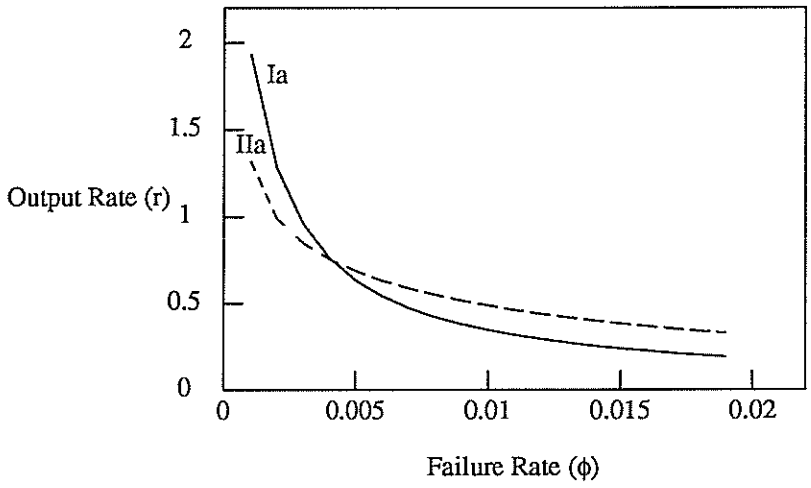

Figure 4. Effect of Failure Rate on Output Rate (Models Ia and IIa).

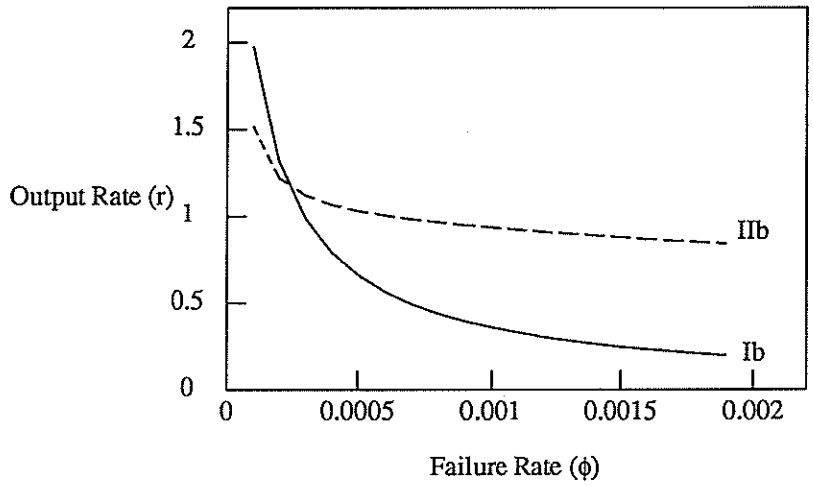

Figure 5. Effect of Failure Rate on Output Rate (Models IIb and Ib).

6. Conclusions and Further Research

We have analyzed two checkpoint/recovery strategies for non-transaction-oriented systems. In the case of synchronous checkpointing without load redistribution, the system appears as a system that can potentially compute at N times the rate of a single-node but with N times the failure rate. The optimum checkpoint rate was determined analytically and has a familiar form. In the case of synchronous checkpointing with load redistribution, state-dependent checkpoint intervals are allowed. The optimum checkpoint rates can be obtained numerically. When many checkpoints are done during the repair period, an approximate symbolic expression yields good solutions. In this case, the optimum checkpoint rate when there are k operational nodes is identical in form to the optimum solution of a k node system with synchronous checkpointing without load redistribution.

An extension to our models is to combine the two models into a single model in which soft and hard failures can occur. A soft failure is one in which the repair time is negligible (e.g., software/hardware reset) while a hard failure requires a long repair time. The response to a hard failure might be to redistribute the load whereas the response to a soft failure might be to wait for recovery without load redistribution.

Although the material presented here assumes a symmetric system in which all nodes act identically (in the stochastic sense), it is a straightforward extension to handle the heterogeneous case. However, a related issue that can also be addressed in our framework is the distribution of tasks in a heterogeneous

system.

We are continuing to improve our numerical solution techniques. The initial approach was to use a multidimensional bisection algorithm with multiple starting points. We are now exploring the use of simulated annealing to find the optimum checkpoint rates and the optimum number of nodes using more general speed-up curves.

Other failures may cause deterioration in system performance. For example, our models do not include the possibility for portions of the communication network to fail. As these components fail, messages will need to be rerouted requiring a greater delay in starting some processes. One way to model this situation is to use a performability approach [10]. In this approach, our model would have to be extended by adding one or more state components representing the reliability state of communication components and then determining an expression for the affects on the computation rate for failed components.

We are in the process of modeling asynchronous checkpointing strategies in which nodes can checkpoint without coordinating with other nodes. Our intent here is to explicitly model the overheads such as message logging and multiple checkpoint rollbacks in these protocols [11,12,13].

The models presented in this paper characterize the system architecture explicitly and the task assignment indirectly through the fault-free speed-up curve. Another approach is to model the computational load by a task graph explicitly and the system architecture indirectly as affects on arc and/or node costs. We are also developing models using this alternative approach and trying to match the model parameters to real workloads (e.g., logic simulation) and use speed-up curves derived from experimental data. Our hope is that results from both approaches can be used iteratively to produce results which identify the fundamental parameters in determining task allocation in a faulty computation environment.

REFERENCES

1. J.S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE Trans. on Comm.* 36(6) pp. 734-743 (June 1988).

2. John W. Young, "A First Order Approximation to the Optimum Checkpoint Interval," *Communications of the ACM* 17(9) pp. 530-531 (Sept. 1974).

3. K. Mani Chandy, James C. Browne, Charles W. Dissly, and Werner R. Uhrig, "Analytic Models for Rollback and Recovery Strategies in Data Base Systems," *IEEE Transactions on Software Engineering* **SE-1**(1) pp. 100-110 (March 1975).

4. E. Gelenbe and D. Derochette, "Performance of Rollback Recovery Systems under Intermittent Failures," *Comm. ACM* **21**(6) pp. 493-499 (June 1978).

5. Erol Gelenbe, "On the Optimum Checkpoint Interval," *Journal of the ACM* **26**(2) pp. 259-270 (Apr. 1979).

6. Asser N. Tantawi and Manfred Ruschitzka, "Performance Analysis of Checkpointing Strategies," *ACM Transactions on Computer System* **2**(2) pp. 123-144 (May 1984).

7. Kang G. Shin, Tein-Hsiang Lin, and Yann-Hang Lee, "Optimal Checkpointing of Real-Time Tasks," *IEEE Transactions on Computers* **C-36**(11) pp. 1328-1341 (Nov. 1987).

8. Victor F. Nicola and Johannes M. Van Spanje, "Comparative Analysis of Different Models of Checkpointing and Recovery," *IEEE Trans. Software Engineering* **16**(8) pp. 807-821 (Aug. 1990).

9. Erol Gelenbe, David Finkel, and Satish K. Tripathi, "Availability of a Distributed Computer System with Failures," *Acta Informatica* **23** pp. 643-655 (1986).

10. R. M. Smith, Kishor S. Trivedi, and A. V. Ramesh, "Performability Analysis: Measures, an Algorithm, and a Case Study," *IEEE Trans. on Computers* **37**(4) pp. 406-417 (Apr. 1988).

11. Anita Borg, Wolfgang Blau, Wolfgang Graetsch, Ferdinand Hermann, and Wolfgang Oberle, "Fault Tolerance under UNIX," *ACM Trans. on Computer Systems* **7**(1) pp. 1-24 (Feb. 1989).

12. David B. Johnson, "Distributed System Fault Tolerance Using Optimistic Message Logging and Checkpointing," , Phd. Thesis, Department of Computer Science, Rice University (Dec. 1989).

13. Robert E. Strom, David F. Bacon, and Shaula A. Yemini, "Volatile Logging in N-Fault-Tolerant Distributed Systems," *Eighteenth Annual Intl. Symp. on Fault-Tolerant Computing: Digest of Papers*, pp. 44-49 (June 1988).

Appendix I

Optimum Synchronous Checkpointing With Load Redistribution

This appendix develops the equations for the state probabilities in Model II (synchronous checkpointing with load redistribution), and derives an expression for the optimum checkpoint rate while in state (N,0,0,0). It also discusses the procedure for finding the optimum checkpoint rate in the other states.

The Markov chain for Model II obeys local balance equations. Local balance equations can be written using three pairs of arcs for each row of states where row k corresponds to the states with k failed nodes:

1) the arcs coming into and going out of a recovery state (column 1);

2) the arc going out of the availabile state (N–k,0,0,k) to a recovery state (0,0,N–k–1,k+1) and the arc coming into the available state (N–k,0,0,k) from the available state (N–k–1,0,0,k+1); and

3) arcs going into and out of checkpoint states (column 3).

These arc pairs correspond to surfaces for states in columns 1, 2, and 3 in the state transition diagram respectively. We equate the flows across these pairs of arcs. Since the solution to these local balance equations also satisfy the global balance equations in which the flow into each state is equal to the flow out of each state. For the states in column 2,

$$\pi_{N-k,0,0,k} = \pi_{N-k+1,0,0,k-1}\frac{(N-k+1)\phi}{\gamma_{N-k}}$$

$$= \left[\pi_{N,0,0,0}\frac{N!}{(N-k)!}\frac{\phi^k}{\gamma_{N-k+1}\cdots\gamma_{N-1}}\right]\frac{1}{\gamma_{N-k}}$$

$$= \pi_{N,0,0,0}\frac{N!}{(N-k)!}\frac{\phi^k}{\Gamma_{N-k}}, \qquad k=1,\ldots,N-1 \tag{1}$$

where

$$\Gamma_k = \begin{cases}\gamma_k\cdots\gamma_{N-1}, & k=1,\ldots,N-1\\ 1, & k=N\end{cases} \tag{2}$$

For the states in column 3,

$$\pi_{0,N-k,0,k} = \pi_{N-k,0,0,k}\frac{\alpha_{N-k}}{\beta}$$

$$= \pi_{N,0,0,0}\frac{N!}{(N-k)!}\frac{\phi^k}{\Gamma_{N-k}}\frac{\alpha_{N-k}}{\beta}, \qquad k=0,...,N-1 \tag{3}$$

For the states in column 1,

$$\pi_{0,0,N-k,k} = \pi_{N-k+1,0,0,k-1}\frac{(N-k+1)\phi}{\delta_{N-k+1}}$$

$$= \left[\pi_{N,0,0,0}\frac{N!}{(N-k+1)!}\frac{\phi^{k-1}}{\Gamma_{N-k+1}}\right]\frac{(N-k+1)\phi}{\delta_{N-k+1}}$$

$$= \pi_{N,0,0,0}\frac{N!}{(N-k)!}\frac{\phi^k}{\Gamma_{N-k+1}\delta_{N-k+1}}, \qquad k=1,...,N \tag{4}$$

Using probability conservation, we solve for $\pi_{N,0,0,0}$.

$$\pi_{N,0,0,0} = \left[\sum_{k=0}^{N} P_k\right]^{-1} \tag{5}$$

where $P_0 = (\pi_{N,0,0,0}+\pi_{0,N,0,0})/\pi_{N,0,0,0}$, $P_N = \pi_{0,0,0,N}/\pi_{N,0,0,0}$, and $P_k = (\pi_{N-k,0,0,k} + \pi_{0,N-k,0,k} + \pi_{0,0,N-k,k})/\pi_{N,0,0,0}$, k=1,...,N–1. Note that the expressions represented by P_k, k=0,...,N, are just the probabilities of being in a state with k failed processors relative to $\pi_{N,0,0,0}$.

$$P_k = \frac{N!\phi^k}{(N-k)!}\left[\frac{1}{\Gamma_{N-k}}\left[1+\frac{\alpha_{N-k}}{\beta}\right]+\frac{1}{\Gamma_{N-k+1}\delta_{N-k+1}}\right], \qquad k=1,...,N-1 \tag{6}$$

$$P_0 = 1 + \frac{\alpha_N}{\beta} \tag{7}$$

$$P_N = N!\frac{\phi^N}{\Gamma_1\delta_1} \tag{8}$$

Note that the recovery rates are now state dependent. Recovery from the states in column 2 in which there were N–k non-failed nodes involves repeating jobs that were being checkpointed at a rate α_{N-k}.

$$\gamma_k^{-1} = \frac{k_\gamma U(k)}{\alpha_k} + g_k, k=1,...,N \tag{9}$$

The parameter g_k, k=1,...,N, is the mean time for repair, load redistribution and state restoration. Recovery from the states in column 1 in which there are N–k non-failed nodes involves repeating jobs that were being checkpointed at a rate α_{N-k+1} and a redistribution of the load. So,

$$\delta_k^{-1} = \frac{k_\delta U(k)}{\alpha_k} + d_k, \qquad k=1,...,N \tag{10}$$

The parameter d_k, k=1,...,N, is the analog to g_k. Note that the last set of checkpoint rates are obtained from the states (N–k+1,0,0,k–1).

The problem now is to find checkpoint rates α_k for k=1,...,N non-failed nodes which will optimize the output rate r. The mean output rate is

$$r = \sum_{k=1}^{N} k\lambda_k = \sum_{k=1}^{N} kU(k)\pi_{k,0,0,N-k}\mu$$

where λ_k is the output rate while in state (k,0,0,N–k), and $\lambda_k = U(k)\pi_{k,0,0,N-k}\mu$. Using the local balance equations to put probabilities in terms of $\pi_{N,0,0,0}$ and simplifying, the output rate can be written as

$$r = \sum_{k=1}^{N} kU(k)\mu\pi_{N,0,0,0}\frac{N!}{k!}\frac{\phi^{N-k}}{\Gamma_k} = \mu\pi_{N,0,0,0}\sum_{k=1}^{N} U(k)\frac{N!}{(k-1)!}\frac{\phi^{N-k}}{\Gamma_k}$$

A necessary (but not sufficient) condition for optimality of the checkpoint rates is that the partial derivatives of r with respect to the checkpoint rates be zero. We now take the derivative of the reward with respect to the N*th* checkpoint rate. Both $\pi_{N,0,0,0}$ and Γ_k, k=1,...,i, are functions of α_i. Using the product rule for derivatives,

$$\frac{d\,r}{d\,\alpha_N} = \sum_{k=1}^{N} \mu U(k)\frac{N!}{(k-1)!}\phi^{N-k}\left[\frac{\dot{\pi}_{N,0,0,0}}{\Gamma_k} + \pi_{N,0,0,0}\frac{d\,\Gamma_k^{-1}}{d\,\alpha_N}\right] \tag{11}$$

where $\dot{\pi}_{N,0,0,0}$ ($\pi_{N,0,0,0}$ dot) is the partial derivative of $\pi_{N,0,0,0}$ with respect to α_N. But since Γ_k is independent of α_N,

$$\frac{d\,r}{d\,\alpha_N} = \sum_{k=1}^{N} \mu U(k)\frac{N!}{(k-1)!}\frac{\phi^{N-k}}{\Gamma_k}\left[\dot{\pi}_{N,0,0,0}\right] = \sum_{k=1}^{N} \mu U(k)\frac{N!}{(k-1)!}\frac{\phi^{N-k}}{\Gamma_k}\left[\pi_{N,0,0,0}^2\sum_{m=0}^{N}\frac{d\,P_m}{d\,\alpha_N}\right] \tag{12}$$

where P_m are the terms in the output rate equation defined earlier. Note that the derivative of r with respect to α_N is zero if the parenthesized expression is zero. The only expressions P_m which are functions of α_N are P_0 and P_1 since Γ_k for all k are independent of α_n and the only term involving δ_N is P_1. The derivatives of P_m are:

$$\frac{d\,P_0}{d\,\alpha_N} = \frac{1}{\beta}$$

$$\frac{d P_1}{d \alpha_N} = -\frac{N!\phi}{(N-1)!} \frac{k_\delta U(N)\mu}{\alpha_N^2} = -\frac{N\phi k_\delta U(N)\mu}{\alpha_N^2}$$

$$\frac{d P_k}{d \alpha_N} = 0, \qquad k=2,...,N$$

There is only one non-negative root of the derivative of r with respect to α_N, and it is given by

$$\alpha_N^* = \sqrt{N\beta\phi\mu k_\delta U(N)} \tag{13}$$

indicating that if an optimum solution exists, the optimum checkpoint interval when all nodes are operational is identical to the interval in a system of N nodes when there is no load redistribution after a failure. These results agree with those determined by using the state transition rate diagrams for N=1,2,3 and solving the balance equations directly.

Finding the other optimum checkpoint rates α_k, k=1,...,N–1 is much more difficult. The derivatives with respect to the other checkpoint rates are:

$$\frac{d r}{d \alpha_k} = \sum_{k=1}^{N} \mu U(k) \frac{N!}{(k-1)!} \phi^{N-k} \left[\frac{\dot{\pi}_{N,0,0,0}}{\Gamma_k} + \pi_{N,0,0,0} \frac{d \Gamma_k^{-1}}{d \alpha_k} \right], \qquad k=1,...,N-1 \tag{14}$$

Unfortunately, all Γ_i, i=1,...,k, are functions of α_k. Even for the case N=2, the derivative is difficult.

However, optimum checkpoint rates can be found symbolically for ranges of γ_k such that the replay time component is small compared to the load redistribution and checkpoint restoration time components. We assume that approximately $\gamma_k = a_k$, dropping the term which is dependent on α_k. Then, the derivative of the reward is zero whenever the derivative of $\pi_{N,0,0,0}$ is zero; that is, we seek the roots of

$$0 = \sum_{m=0}^{N} \frac{d P_m}{d \alpha_k}, \qquad k=1,...,N-1 \tag{15}$$

The only terms involving α_k are those containing α_k and δ_k since Γ_k is now independent of any α_i. Examination of the expressions for P_m shows that P_m is a function of both α_{N-m} and δ_{N-m+1}, indicating that the terms for which $k = N-m$ and $k = N-m+1$ contain non-zero terms after taking the derivative with respect to α_k. Thus the terms P_m in which $m = N-k$ and $m = N-k+1$ will contain the terms of interest.

$$\sum_{m=0}^{N} \frac{d P_m}{d \alpha_k} = \frac{N!\phi^{N-k}}{(N-(N-k))!} \frac{1}{\Gamma_k \beta} - \frac{N!\phi^{N-k+1}}{(N-(N-k+1))!} \frac{k_\delta U(k)}{\Gamma_k \alpha_k^2}, \qquad k=1,...,N-1$$

which reduces to

$$= \frac{N!\phi^{N-k}}{(k-1)!\Gamma_k}\left[\frac{1}{k\beta} - \frac{\phi k_\delta U(k)}{\alpha_k^2}\right], \qquad k=1,\ldots,N-1$$

Setting the derivative to zero and solving for α_k yields the optimum checkpoint rate.

$$\alpha_k^* = \sqrt{k\beta\phi k_\delta U(k)}, \qquad k=1,\ldots,N-1 \tag{16}$$

We can interpret this result as saying that when nodes can be non-operational for periods of time much longer than the replay period, that the system looks like a system with k nodes without load redistribution and the optimum checkpoint interval is the same as found in Model I with the appropriate number of nodes. Unfortunately, the conditions allowing for the above approximation do not always hold.