CS 533	**Name:** ____________________
Spring 2024	
Homework 4	**NetID:** ____________________
Assigned: 4/4/2024	
Due: 4/30/2024	**UIN:** ____________________

This homework contains 5 pages (including this cover page) and 4 questions.

Please clearly print or typeset your answers.

Note: up to 5 points (out of 50) may be deducted for illegible submissions

Problem Breakdown

Question	Points	Score
1	14	
2	9	
3	13	
4	14	
Total:	50	

1. (14 points) Software Reliability

   (a) (2 points) Your lab partner claims that on-the-fly bug detection schemes are overkill when it comes to debugging parallel programs. If you know the program inputs at the time a bug occurred, you can feed them into a static tester to perform diagnosis, he states. Is he right? (Consider different bug types to evaluate his statement).

   (b) (12 points) The following questions relate to paper 9d [1]

      i. In this system the main memory is protected by parity. Why is it still necessary to create checkpoints? Why is it necessary to perform two step checkpointing?

      ii. Why does the system have to maintain multiple checkpoints?

      iii. What types of applications incur the most overhead on ReVive? What types of applications have the least amount of overhead?

      iv. Why does distributed memory mirroring have less performance overheads than using distributed parity?

      v. After phase 3 of the recovery from an error is complete, what are the ways performance can still be hurt (assuming that no more errors occur)?

      vi. Explain why if no logging bits are included in the cacheline recovery, logs have to be processed in reverse order?

      vii. Assume the checkpointing frequency is once per 120ms. The average error detection and error recovery latencies are 60ms and 500ms respectively. What is the availability of the system if the mean time between errors is 1 hour?

[1] M. Prvulovic et al. "ReVive: Cost-Effective Architectural Support for Rollback Recovery in Shared-Memory Multiprocesor," ISCA, 2002

2. (9 points) Operating System/Database
   (a) (2 points) What sort of architectural support help or hurt the following:
      1. OLTP
      2. DSS
   (b) (7 points) Refer to paper 10e [2] when answering the following questions:
      i. What are the three application characteristics that determine the effectiveness of cache affinity? Explain the rationale for each.
      ii. Give the details of the affinity scheduling algorithm the paper describes. What is the complex version of their algorithm?
      iii. Why is their affinity scheduler no better than increasing the time quantum for scientific applications?
      iv. What are some drawbacks the authors mention about their scheduler?
      v. What is the difference between time quantum and effective time slice? How does the scheduler usually handle this discrepancy?

[2] J. Torrellas et al. "Evaluating the Performance of Cache-Affinity Scheduling in Shared-Memory Multiprocessors", JPDC, 1995

3. (13 points) Interconnection Networks/MPI

   (a) (2 points) Which interconection architectures are possible in a direct network if each router has

      1. 1 input channel and 1 output channel?
      2. 2 input channels and 2 output channels?

   (b) (2 points) Discuss the main sources of overhead in emulating message passing on a shared memory machine with a write-invalidate cache coherence protocol.

   (c) (2 points) Cannon's algorithm is message-passing based matrix multiplication algorithm for 2D torus systems. Is it more efficient than shared-memory based algorithms? Please explain.

   (d) (7 points) Several architectures have been proposed which embed the fundamentals of message passing directly in hardware. One of the more notable designs is the J-machine [3]

      i. Why is tagged memory helpful in the J-Machine?
      ii. What are futures? How are they supported in the J-Machine?
      iii. Describe the network used in this system. How is deadlock prevented?
      iv. How does the J-Machine try to elminate the usual overhead associated with message-passing?
      v. What does it mean to be a *message-driven* architecture?

[3] William J. Dally. "The J-Machine: System Support for Actors," 1988

4. (14 points) Dataflow Architectures

   (a) (7 points) The following questions relate to paper 12a [4]

      i. What are the two fundamental issues with using Von Neumann machines for parallel processing? Explain the ways dataflow machines solve these problems.

      ii. How can deadlock occur on the hybrid machine?

      iii. Why is the hybrid machine considered a mixture of a Dataflow and Von Neumann Architecture?

      iv. Describe the author's approach for preventing deadlock with using scheduing quanta.

      v. How does the hybrid architecture enable fast context switching?

   (b) (5 points) The following questions relate to paper 12b [5]

      i. How does one program a dataflow machine?

      ii. If a code is reentrant (loops, recursion), what are ways to ensure correct execution? What are the overheads associated with each technique?

      iii. What are ways the author suggests to limit the amount of tags needed?

      iv. Describe the matching unit and fetching unit operations in the Manchester Datflow Machine.

   (c) (2 points) Google's TPU [6] uses systolic dataflow for efficient matrix multiplication. Explain how its systolic execution unit reduces energy.

[4] Robert A. Iannucci "Toward A Dataflow/Von Neumann Hybrid Architecture" ISCA, 1998.

[5] Arthur H. Veen "Dataflow Machine Architecture", ACM Computing Surveys, 1986

[6] Jouppi et al. "In-Datacenter Performance Analysis of a Tensor Processing Unit", 2017