The sort paper is due in two parts – the preliminary paper is due in Assignment 9, and the final paper is due in Assignment 14. See Assignment 9 for the setup details for LaTeX and RStudio. This document is the specification for the paper content.

1. *Purpose of the paper*
   The purpose of the paper is:

   - To take performance data on the insertion sort, selection sort, heap sort, merge sort, and quick sort as a function of the number of values sorted.
   - To do a least squares curve fit on the data with an n^2 model and an $n \lg n$ model.
   - To state the theoretical Θ increase in execution time as a function of the number of values sorted.
   - To report whether or not the data confirms the theory.
   - To recommend the best of the five sort algorithms analyzed.
   - To answer any other interesting questions about the data that may arise from your analysis.

2. *Structure of the paper*
   The paper must be typeset in LaTeX, use single-column format with 1.5 line spacing, and be organized into the following sections.

   Abstract

   In one short paragraph, describe the purpose of your paper and your conclusions in general.

   1. Introduction

   Explain the course assignment. Explain in the last paragraph what each following section describes.

   2. Method

   Describe the performance metric issues.

   2.1 Sort algorithms

   Describe the characteristics of each sort.

   2.2 Data collection

   Describe the OO design pattern used to take the data. Describe the computer runs that took the data.

   2.3 Analysis

   Define mathematically the RSE and explain how it is used to determine the asymptotic run time. Explain how to determine the best sort.

   3. Results

   Include the following subsections with appropriate plots and/or tables.

   3.1 Raw data

   Show the raw data in table form and graphically and discuss any interesting overall features of it.

   3.2 Insertion sort

   Analyze the insertion sort.

   3.3 Selection sort

   Analyze the selection sort.

3.4 Heap sort

Analyze the heap sort.

3.5 Merge sort

Analyze the merge sort.

3.6 Quick sort

Analyze the quick sort.

3.6 Sort comparisons

Analyze which sort is best.

4. Conclusions

One paragraph of the conclusions from your experiment. This section is a summary of your results section with specific conclusions. It differs from the abstract, which summarizes the conclusions in general.

3. *Grading rubric*
The final paper will count for 100 points toward your homework score. It will be graded according to the following rubric.

- 5 points: Form, LaTeX layout.
- 15 points: Grammar, punctuation, style.
- 5 points: Abstract.
- 5 points: Introduction.
- 20 points: Method.
- 40 points: Results.
- 5 points: Conclusion.
- 5 points: References.

The most important part of the evaluation is the quality and completeness of the Results section. This paper is open-ended, and it is your decision how to analyze the data based on the performance metrics that are available. There will probably be some unexpected results from your investigation. You should explain your results based on your understanding of the methods.

Following are some style guidelines to which your paper *must* adhere. Included are some excerpts from this sample research paper to give you an idea of the organization and style of English to use in your paper.

Origin 2000 Design Enhancements for Communication Intensive Applications

Gheith A. Abandah
Department of Electrical Engineering
University of Jordan
Amman - Jordan
abandah@fet.ju.edu.jo

Edward S. Davidson
Advanced Computer Architecture Laboratory
University of Michigan
1301 Beal Avenue, Ann Arbor, MI 48109
davidson@eecs.umich.edu

4. *Abstract and Conclusions*

The Abstract is a one-paragraph *general* summary of the content of the paper. Its purpose is for a potential reader to decide whether she would be interested to read the complete paper. The Conclusions section is also a summary of the content of the paper, but it can be more than one paragraph and includes the *specific* results of the paper. Following are the Abstract and Conclusions of the sample research paper. Notice how much more detail is included in the Conclusions section.

Abstract

The SGI Origin 2000 is designed to support a wide range of applications and has low local and remote memory latencies. However, it often has a high ratio of remote to local misses. In this paper, we evaluate the Origin 2000 performance with communication intensive applications. We use detailed execution-driven simulation of six shared-memory applications. This paper evaluates a base, Origin 2000-like system and three derived systems that incorporate techniques to reduce the communication cost by lowering the ratio of remote misses. We show that the performance of these applications is generally improved when the local bus is used in the snoopy mode, the number of processors per node is increased, the processors use the Illinois cache coherence protocol, or when adding a snoopy cache to retain remote data within each node. Illinois protocol and the interconnect cache reduce the average remote miss ratio by 16% and 21%, respectively.

7. Conclusions

In this paper, we have studied the communication cost in CC-NUMA systems and evaluated some techniques that have the potential for reducing it. System **D**, which relies purely on the directory for ensuring cache coherence, favors sequential programs in multiprogrammed parallel environments over shared-memory applications with high communication percentages. For our six applications, the three snoopy systems handle communication better than system **D**. System **I**, which allows processors to supply clean cached lines, and system **IC**, which uses an interconnect cache for reducing the communication cost of remote line accesses, both have superior performance.

System **IC** handles local producer-consumer communication better than system **I**. Moreover, unlike system **I**, system **IC** is compatible with the cache coherence protocols supported by modern processors. These two systems, however, do have increased system cost: system **I** because it requires higher processor cache bandwidth, and system **IC** because it requires the addition of an interconnect cache in each node.

Although system **IC** increases the cost of the node, it has less traffic and contention than system **D** and permits more processors per node. When the node becomes larger, the overall system cost decreases as fewer network links, routers, and coherence controllers are needed. The cost reduction with larger nodes may exceed the IC cost.

The IC can be implemented in the local memory to reduce its cost. In this case, a dedicated section of memory is used to hold the data, but the cache coherence controller still stores the IC tags in order to keep up with the demand for snooping. Consequently, the IC can continue to participate in the bus coherence protocol without causing more delay than a typical processor cache. As we assume that the IC supplies lines with a 190 nsec latency, getting an IC data line from the local memory would not take much more time.

Finally, we have noticed that the effectiveness of system **IC** is best with applications that have high communication costs. Furthermore, it reduces the execution time of all six case study applications which have been carefully designed to achieve low communication rates (the execution time reduction with problem size I is larger than with size II). Thus, we expect that other less-tuned shared-memory applications would have at least this much benefit. This technique reduces the communication cost and consequently lowers the NUMA factor of DSM systems which is a step toward enabling CC-NUMA systems to efficiently support more applications with less tuning effort.

5. *Introduction*
Here is the Introduction section of the sample research paper. The last paragraph of your introduction should refer to the following sections by number, as in this paper.

1. Introduction

The SGI Origin 2000 is designed to achieve low remote latency [11]. However, it often has a high ratio of remote to local misses, consequently its average latency and internode traffic are high [2]. The main causes of the Origin 2000's frequent remote misses are:

1. Each node contains only two processors, thus the shared data is often spread over a large number of nodes.
2. The processor does not snoop the requests of the other local processor, thus remote traffic is generated even when one processor can satisfy its neighbor's request.
3. No interconnect cache is used for caching remote data that is referenced by the local processors, thus remote traffic is generated whenever a processor requests a remote line, even when this line has recently been requested by the other local processor.

However, there are no magic solutions for these weaknesses. Although increasing the number of processors per node would increase the possibility of finding shared data in the local memory, the local bus will be more heavily utilized and contention may increase the memory latency; allowing processors to snoop the requests of other processors decreases remote traffic, but the snooping overhead may affect the processor performance; and incorporating an interconnect cache, as in the Convex SPP1000, may increase the remote latency [6, 2].

In this paper, we evaluate four Origin 2000-like systems that address these design trade-offs using execution-driven simulation

of six applications. Section 2 reviews the communication cost in these systems; Section 3 expands on these design trade-offs and introduces the four evaluated systems; Section 4 describes the experimental setup and the applications used; Section 5 provides a preliminary evaluation to investigate the design spaces of cache size, number of processors, number of memory banks, and processor model; Section 6 evaluates the execution time and traffic of the four systems; and Section 7 presents the conclusions.

6. *Assertion – evidence*
The writing style in the Results section should be to make a statement (assertion), then back it up by quoting data or referring to graphs (evidence). Do not make an assertion and then leave it to the reader to extract the evidence from a table or a plot. For example, the statement

"The data in Figure 4 shows that the assignment count for merge sort grows as $\Theta(n^2)$."

is not a good statement, because it leaves it to the reader to extract the evidence from the table. Instead, you should quote the RSE for the $\Theta(n^2)$ model, quote the RSE for the $\Theta(n \lg n)$ model, observe which one is smaller, and then draw your conclusion.

7. *Personal pronouns*
As much as possible, avoid personal pronouns "I" and "we". Sometimes their use is unavoidable, but they are unfortunately used too frequently in the literature. Eliminating personal pronouns usually makes the sentence more concise.

Here is a critique from the Conclusions section of the sample paper. The original sentence is:

"Finally, we have noticed that the effectiveness of system **IC** is best with applications that have high communication costs."

You could write it more concisely without personal pronouns like this:

"Finally, the effectiveness of system **IC** is best with applications that have high communication costs."

Here is another critique. The original sentence is:

"We use the Communication Contention Analysis Tool (CCAT) to evaluate the performance of these four systems using six applications."

An improvement is:

"The Communication Contention Analysis Tool (CCAT) evaluates the performance of these four systems using six applications."

8. *Present tense*
As much as possible, use present tense, not future tense. For example, do not write:

"Section 3 will explain how the data was gathered…"

Instead, write in the present tense like this:

"Section 3 explains how the data was gathered…".

9. *Active voice*
As much as possible, use active voice instead of passive voice. For example, instead of

"Tables of the raw data are shown in Section 4."

write

"Section 4 shows tables of the raw data."

Using passive voice unnecessarily is probably the most common stylistic mistake in the technical literature. Try hard to make all your sentences active instead of passive. Conversion of a sentence from the passive voice to the active voice usually decreases the number of words in the sentence and makes it crisper.

10. *Conciseness*
Be concise. For example, here is a statement from a student paper.

"The five sorting algorithms being discussed in this paper are Insert Sort, Select Sort, Heap Sort, Merge Sort, and Quick Sort."

A more concise statement is

"This paper discusses five sorting algorithms: Insert Sort, Select Sort, Heap Sort, Merge Sort, and Quick Sort."

11. *Amount versus number*
Students frequently use the word "amount" when the word "number" is more appropriate. For example, this statement from a student paper

"…the data for the amount of comparisons and assignments seems to stay the same …"

should be written

"…the data for the number of comparisons and assignments seems to stay the same …"

12. *Figures*
How you place your figures and tables in your paper is crucial. Follow these rules:

- Each table and figure must be numbered and have a caption. This format is provided in the sample paper template.
- A common mistake is to clump all the tables and figures together separate from the running text of the paper. Instead, you must intersperse your figures and tables throughout the text, so that each figure, as much as possible, is on the same page as the paragraph in which the first reference to it occurs.
- Every figure and table must be introduced for the first time by a descriptive statement in the running text. A standard way to introduce a figure is

  "Figure *x* shows…".
- When you introduce a figure you will invariably repeat some of the phrases in the caption of the figure. For example, Table 2 in the sample research paper has the caption,

  "Table 2. Signal occupancies of shared resources (nsec)."

  The first reference to this table is

  "…Table 2 shows the occupancies of the shared resources used with the four systems."

CCAT differentiates between two signal types: a short signal that does not carry data, e.g., a processor request; and a long signal that carries a 128-byte data line, e.g., a data response. For each signal type, Table 2 shows the occupancies of the shared resources used with the four systems. In the interconnection network, a packet carrying a short signal is 16 bytes, and one carrying a long signal is 144 bytes.

Table 2. Signal occupancies of shared resources (nsec).

Type	Short signal	Long signal
Processor request bus occupancy	10	170
Processor response bus occupancy	0	170
CCC recall bus occupancy	10	NA
CCC response bus occupancy	10	160
Memory bank access occupancy	100	100
Interconnection link occupancy	20	180