How to Write Papers

Dr. Xiao Qin

Auburn University
http://www.eng.auburn.edu/~xqin
xqin@auburn.edu

These slides are adapted from notes by Dr. Nitin Vaidya (UIUC)

How to write a paper

- When you have truly exceptional results.
  - Probably doesn’t matter how you write, people will read it anyway
- Most papers are not that exceptional
- Good writing makes significant difference
- Better to say little clearly, than saying too much unclearly

Readability

- If the paper is not readable, author has not given writing sufficient thought
- Two kinds of referees
  - If I cannot understand the paper, it is the writer's fault
  - If I cannot understand the paper, I cannot reject it
- Don't take chances. Write the paper well.
- Badly written papers typically do not get read

Do not irritate the reader

- No one is impressed anymore by Greek symbols
- Define notation before use

system at an aggregate rate of $\lambda = \sum_{i=1}^{m} \lambda_i$. Let $\boxed{p_{ij}}$ be the probability that the tasks of the ith class are dispatched to node j, where $1 \leq j \leq n$. Then, the aggregate task arrival rate of the jth node is expressed as

$$\Lambda_j = \sum_{i=1}^{m} p_{ij} \lambda_i. \tag{1}$$

- If you use much notation, make it easy to find
  - summarize most notation in one place

node. The function of LID is to detect whether or not the candidate node is overloaded. If it is overloaded, the current task class will be assigned to the node with the lightest load. Otherwise, the task class will be allocated to the candidate node. A detailed description of the SSAC strategy can be found in Section 3. To illustrate how APL works, we give an example (see Fig. 2).

In Fig. 2, we assume that there are eight nodes in a system. The first row shows the availability levels that the eight nodes exhibit. The second row displays the expected finish times for task class i on the nodes. The third row is a node list sorted by the node's availability level in a nondecreasing order. We suppose that the task's availability requirement is 0.85. Therefore, the first four nodes (3, 8, 4, 1) will be put into set N_i as all of them can fully satisfy the task's availability requirement. SSAC will eventually choose node 8 (the black node) as the candidate node because it can minimize the expected response time of the task class.

2.2 Modeling Multiclass Tasks with Availability Requirements

For future reference, we summarize the notation that is used throughout this paper in Table 1. There are m classes of tasks submitted to a heterogeneous system by users. Tasks are independent of one another. Each class of tasks requires a common availability specified by a user. Values of availability levels are normalized in the range from 0 to 1.0. For example, users may set the availability levels of critical task classes to 1.0, which means that critical tasks should be assigned to a node that ensures that the task can be successfully completed.

TABLE 1
Definitions of Notation

Notation	Definition
n	Number of nodes in a heterogeneous system. ($1 \bullet n < \infty$)
m	Number of task classes submitted to the system. ($1 \bullet m < \infty$)
λ_i	Arrival rate of tasks in the ith class.
Λ_j	Aggregate task arrival rate of the jth node. (see Eq. 1)
p_{ij}	Probability that tasks of the ith class are dispatched to node j.
μ_{ij}	Service rate of tasks in class i allocated on node j.
ρ_i	Service utilization of class i. (see Eq. 2)
ϕ_j	Service utilization for all tasks allocated to node j. (see Eq. 3)
ϕ	Summation of the service utilizations of all nodes. (see Eq. 4)
TN_j	Average response time of node j. (see Eq. 5)
TC_i	Expected response time of class i tasks. (see Eq. 8)
T	Mean response time of jobs averaged over all the classes. (see Eq. 9)
ξ_j	Availability of node j. ($0 \le \xi_j \le 1$)
a_i	Availability requirement of class i. ($0 \le a_i \le 1$)
δ_j	Availability shortage of node j. (see Eq. 11)
d_{ij}	Availability shortage factor of class i on node j. (see Eq. 12)
AC_{ij}	Availability cost of class i on node j. (see Eq. 14)
θ_j	Unavailable rate of node j. (see Eq. 15)
AC_i	Availability cost of class i. (see Eq. 16)
A_i	Availability of class i. (see Eq. 17)
A	Availability exhibited by the system. (see Eq. 18)

Do not irritate the reader

- Avoid Using Too Many Acronyms
  - AUTMA ?!
  - To save space?
  - Guarantee Ratio or GR

- You may know the acronyms well.
  Do not assume that the reader does (or cares to)

How to write a theory paper

- Unreadability is not the same as formalism
- Reader should be able to understand contributions without reading all details
- If some proofs are not too important, relegate them to an appendix
  - Proofs are not as worthy as new proof techniques

How to write a theory paper

- If some proofs are not too important, relegate them to an appendix

Lemma 2.1. $R_{S_a} = s[1 + A_{S_a}]$, *where*

$$A_{S_a} = A_{S_K S_K} + \frac{K-1}{K} A_{S_K S_{K-1}} + \cdots + \frac{2}{K} A_{S_K S_2} + \frac{1}{K} A_{S_K S_1}.$$

Here, A_{S_a} refers to the mean number of jobs in S_a seen just prior to arrival at S_a and $A_{S_K S_i}$ represents the mean number of jobs in S_i seen by a job just prior to arrival at S_K.

Proof. Given in Appendix B. □

APPENDIX B ← Before references

PROOF FOR LEMMA 2.1

Proof. Since S_a is a nonparallel subsystem with mean service time s, its response time is given by $R_{S_a} = s[1 + A_{S_a}]$ where A_{S_a} represents the mean number of jobs

[Ref] Response time analysis of parallel computer and storage systems-IEEE TPDS2001

How to write a systems paper

- Provide sufficient information to allow people to reproduce your results
  - people may want to reproduce exciting results
  - do not assume this won't happen to your paper
  - besides, referees expect the information
- Do not provide wrong information
- Sometimes hard to provide all details in available space
  - may be forced to omit some information
  - judge what is most essential to the experiments
  - cite a tech report for more information
  - Provide source code

Discuss related work

- Explain how your work relates to state of the art
  - Summarize pros and cons of existing approaches

- Discuss relevant past work by other people too
- Remember, they may be reviewing your paper.
  - Avoid: The scheme presented by Qin performs terribly
  - Prefer: The scheme by Qin does not perform as well in scenario X as it does in scenario Y

- Avoid offending people, unless you must

Tell them your shortcomings

- If your ideas do not work well in some interesting scenarios,
  - Tell readers
  - Explain why the ideas do not work well
  - May point out how to improve

- People appreciate a balanced presentation

How to write weak results

- If results are not that great, come up with better ones
- Do not hide weak results behind bad writing
  - Be sure to explain why results are weaker than you expected
- If you must publish: write well, but may have to go to second-best conference
  - Only a few conferences in any area are worth publishing in
  - Too many papers in poor conferences bad for your reputation
  - Just because a conference is “IEEE” or “ACM” or “International” does not mean it is any good
- If results not good enough for a decent conference, rethink your problem/solution

Miscellaneous

- Read some well-written papers
  - award-winning papers from conferences
  - organization
- Avoid long sentences
- If you have nothing to say, say nothing
  - don’t feel obliged to fill up space with useless text
  - if you must fill all available space, use more line spacing, greater margins, bigger font, bigger figures

Technical reports

- Useful to get early feedback from other researchers
- Puts a timestamp on your work
- Can include more information / results than might fit in a paper

Questions

Please ask at any time!