How to write a Masters thesis proposal

DAT315 The Computer Scientist in Society



This course

 Write "a" Masters thesis proposal

 Get feedback from course assistants

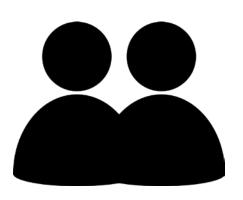
Judged on writing

Your real Masters thesis

No feedback

 Judged by your examiner

Same deadline!





Operating

R. Stockton Gaines

Time, Clocks, and the Ordering of Events in a Distributed System

Leslie Lamport Massachusetts Computer Associates, Inc.

The concept of one event happening before another in a distributed system is examined, and is shown to define a partial ordering of the events. A distributed algorithm is given for synchronizing a system of logical clocks which can be used to totally order the events. The use of the total ordering is illustrated with a method for solving synchronization problems. The algorithm is then specialized for synchronizing physical clocks, and a bound is derived on how far out of synchrony the clocks can become.

Key Words and Phrases: distributed systems, computer networks, clock synchronization, multiprocess systems

CR Categories: 4.32, 5.29

Introduction

The concept of time is fundamental to our way of thinking. It is derived from the more basic concept of the order in which events occur. We say that something happened at 3:15 if it occurred after our clock read 3:15 and before it read 3:16. The concept of the temporal ordering of events pervades our thinking about systems. For example, in an airline reservation system we specify that a request for a reservation should be granted if it is made before the flight is filled. However, we will see that this concept must be carefully reexamined when considering events in a distributed system.

General permission to make fair use in teaching or research of all or part of this material is granted to individual readers and to nonprofit libraries acting for them provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery. To otherwise reprint a figure, table, other substantial excerpt, or the entire work requires specific permission as does republication, or systematic or multiple reproduc-

This work was supported by the Advanced Research Projects Agency of the Department of Defense and Rome Air Development Center. It was monitored by Rome Air Development Center under contract number F 30602-76-C-0094.

Author's address: Computer Science Laboratory, SRI Interna-tional, 333 Ravenswood Ave., Menlo Park CA 94025. © 1978 ACM 0001-0782/78/0700-0558 \$00.75

the ACM

A distributed system consists of a collection of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. A network of interconnected computers, such as the ARPA net, is a distributed system. A single computer can also be viewed as a distributed system in which the central control unit, the memory units, and the input-output channels are separate processes. A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process.

We will concern ourselves primarily with systems of spatially separated computers. However, many of our remarks will apply more generally. In particular, a multiprocessing system on a single computer involves problems similar to those of a distributed system because of the unpredictable order in which certain events can occur.

In a distributed system, it is sometimes impossible to say that one of two events occurred first. The relation "happened before" is therefore only a partial ordering of the events in the system. We have found that problems often arise because people are not fully aware of this fact

In this paper, we discuss the partial ordering defined by the "happened before" relation, and give a distributed algorithm for extending it to a consistent total ordering of all the events. This algorithm can provide a useful mechanism for implementing a distributed system. We illustrate its use with a simple method for solving synchronization problems. Unexpected, anomalous behavior can occur if the ordering obtained by this algorithm differs from that perceived by the user. This can be avoided by introducing real, physical clocks. We describe a simple method for synchronizing these clocks, and derive an upper bound on how far out of synchrony they

The Partial Ordering

Most people would probably say that an event a happened before an event b if a happened at an earlier time than b. They might justify this definition in terms of physical theories of time. However, if a system is to meet a specification correctly, then that specification must be given in terms of events observable within the system. If the specification is in terms of physical time, then the system must contain real clocks. Even if it does contain real clocks, there is still the problem that such clocks are not perfectly accurate and do not keep precise physical time. We will therefore define the "happened before" relation without using physical clocks.

We begin by defining our system more precisely. We assume that the system is composed of a collection of processes. Each process consists of a sequence of events. Depending upon the application, the execution of a subprogram on a computer could be one event, or the execution of a single machine instruction could be one

R. Stockton Gaines

Time, Clocks, and the Ordering of Events in a Distributed System

Leslie Lamport Massachusetts Computer Associates, Inc.

The concept of one event happening before another in a distributed system is examined, and is shown to define a partial ordering of the events. A distributed algorithm is given for synchronizing a system of logical clocks which can be used to totally order the events. The use of the total ordering is illustrated with a method for solving synchronization problems. The algorithm is then specialized for synchronizing physical clocks, and a bound is derived on how far out of synchrony the clocks can become

Key Words and Phrases: distributed systems computer networks, clock synchronization, multiprocess systems.

CR Categories: 4.32, 5.29

Introduction

The concept of time is fundamental to our way of thinking. It is derived from the more basic concept of the order in which events occur. We say that something happened at 3.15 if it occurred after our clock read 3.15 and before it read. ordering of some

A distributed system consists of a collection of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. A network of interconnected computers, such as the ARPA net, is a distributed system. A single computer can also be viewed as a distributed system in which the central control unit, the memory units, and the input-output channels are separate processes. A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process.

We will concern ourselves primarily with systems of spatially separated computers. However, many of our remarks will apply more generally. In particular, a multiprocessing system on a single computer involves problems similar to those of a distributed system because of the unpredictable order in which certain events can

In a distributed system, it is sometimes impossible to say that one of two events occurred first. The relation "happened before" is therefore only a partial ordering of the events in the system. We have found that problems often arise because people are not fully aware of this fact

In this paper, we discuss the partial ordering defined by the "happened before" relation, and give a distributed algorithm for extending it to a consistent total ordering of all the events. This algorithm can provide a useful mechanism for implementing a distributed system. We illustrate its use with a simple method for solving synchronization problems. Unexpected, anomalous behavior can occur if the ordering obtained by this algorithm differs from that perceived by the user. This can be avoided by introducing real, physical clocks. We describe a simple method for synchronizing these clocks, and derive an upper bound on how far out of synchrony they

results

What is the problem to be solved?

• Is it interesting?

• Is it important?

• Is it feasible?

- What is the question to be answered?
 - We should learn something from the project.
- There should be a clearly defined research question to be answered.

What do we mean by improvement?

What's Evosuite?

"Is it possible to improve Evosuite's test case generation by taking contracts into account?"

What are contracts in this setting?

• Is it interesting?

"Is it possible to build a web site for <...> AB?"



The answer to the question should not be obvious



• Is it important?

Company?
Open-source project?

- Is there a "customer"?
 - Why do they want it?
 - What do they plan to do with it?
- What will a solution make possible?

• Is it feasible?

"I plan to prove that P=NP.
Success will be the death knell
of non-quantum cryptography."

2018

January					February						March									
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6	28	29	30	31	1	2	3	25	26	27	28	1	2	3
7	8	9	10	11	12	13	4	5	6	7	8	9	10	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24	18	19	20	21	22	23	24
28	29	30	31	1	2	3	25	26	27	28	1	2	3	25	26	27	28	29	30	31
4	5	6	7	8	9	10	4	5	6	7	8	9	10	1	2	3	4	5	6	7
		-	۱pr	il .						May	,						Jun	е		
Sun	Mon	Tue	Wod	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	29	30	1	2	3	4	5	27	28	29	30	31	1	2
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23
29	30	1	2	3	4	5	27	28	29	30	31	1	2	24	25	26	27	28	29	30
6	7	8	9	10	11	12	3	4	5	6	7	8	9	1	2	3	4	5	6	7
July					August						September									
			July	,					A	ugu	st				5	Sep	ten	ıbe	r	
Sun	Mon		July	Thu	Fri	Sat	Sun	Mon	Al Tue	ugu Wed	st Thu	Fri	Sat	Sun	Mon	Sep	ten Wed	nbe Thu	r Fri	Sat
Sun	Mon 2				Fri	Sat	Sun 29	Mon 30		•		Fri	Sat	Sun 26						Sat
		Tue	Wed	Thu					Tue	Wed	Thu				Mon	Tue	Wed	Thu	Fri	
1	2	Tue 3	Wed 4	Thu 5	6	7	29	30	Tue 31	Wed 1	Thu 2	3	4	26	Mon 27	Tue 28	Wed 29	Thu 30	Fri 31	1
1 8	2 9	3 10	Wed 4 11	Thu 5 12	6 13	7 14	29 5	30 6	Tue 31 7	1 8	Thu 2 9	3 10	4 11	26 2	Mon 27 3	Tue 28 4	Wed 29 5	Thu 30 6	Fri 31 7	1 8
1 8 15	2 9 16	3 10 17	Wed 4 11 18	5 12 19	6 13 20	7 14 21	29 5 12	30 6 13	Tue 31 7 14	Wed 1 8 15	Thu 2 9 16	3 10 17	4 11 18	26 2 9	Mon 27 3 10	Tue 28 4 11	Wed 29 5 12	30 6 13	Fri 31 7 14	1 8 15
1 8 15 22	2 9 16 23	Tue 3 10 17 24	Wed 4 11 18 25	5 12 19 26	6 13 20 27	7 14 21 28	29 5 12 19	30 6 13 20	Tue 31 7 14 21	1 8 15 22	2 9 16 23	3 10 17 24	4 11 18 25	26 2 9 16	Mon 27 3 10 17	Tue 28 4 11 18	Wed 29 5 12 19	Thu 30 6 13 20	Fri 31 7 14 21	1 8 15 22
1 8 15 22 29	9 16 23 30	3 10 17 24 31	Wed 4 11 18 25 1	5 12 19 26 2	6 13 20 27 3	7 14 21 28 4	29 5 12 19 26	30 6 13 20 27	Tue 31 7 14 21 28	1 8 15 22 29	2 9 16 23 30	3 10 17 24 31	4 11 18 25 1	26 2 9 16 23	Mon 27 3 10 17 24	Tue 28 4 11 18 25	Wed 29 5 12 19 26	Thu 30 6 13 20 27	Fri 31 7 14 21 28	1 8 15 22 29
1 8 15 22 29	9 16 23 30	Tue 3 10 17 24 31 7	Wed 4 11 18 25 1	5 12 19 26 2 9	6 13 20 27 3	7 14 21 28 4	29 5 12 19 26	30 6 13 20 27 3	7 14 21 28 4	1 8 15 22 29	9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1	26 2 9 16 23	Mon 27 3 10 17 24 1	Tue 28 4 11 18 25 2	Wed 29 5 12 19 26	Thu 30 6 13 20 27 4	7 14 21 28 5	1 8 15 22 29
1 8 15 22 29	9 16 23 30	Tue 3 10 17 24 31 7	Wed 4 11 18 25 1 8	5 12 19 26 2 9	6 13 20 27 3	7 14 21 28 4	29 5 12 19 26	30 6 13 20 27 3	7 14 21 28 4	1 8 15 22 29 5	9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1	26 2 9 16 23	Mon 27 3 10 17 24 1	Tue 28 4 11 18 25 2	Wed 29 5 12 19 26 3	Thu 30 6 13 20 27 4	7 14 21 28 5	1 8 15 22 29
1 8 15 22 29 5	9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1 8	5 12 19 26 2 9	6 13 20 27 3 10	7 14 21 28 4 11	29 5 12 19 26 2	30 6 13 20 27 3	7 14 21 28 4	1 8 15 22 29 5	9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1 8	26 9 16 23 30	Mon 27 3 10 17 24 1	11 18 25 2	wed 29 5 12 19 26 3	30 6 13 20 27 4	Fri 31 7 14 21 28 5	1 8 15 22 29 6
1 8 15 22 29 5	2 9 16 23 30 6	Tue 3 10 17 24 31 7	Wed 4 11 18 25 1 8	5 12 19 26 2 9	6 13 20 27 3 10	7 14 21 28 4 11	29 5 12 19 26 2	30 6 13 20 27 3	Tue 31 7 14 21 28 4	Wed 1 8 15 22 29 5	2 9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1 8	26 9 16 23 30	Mon 27 3 10 17 24 1	Tue 28 4 11 18 25 2	Wed 29 5 12 19 26 3 Wed Wed	Thu 30 6 13 20 27 4 bei	Fri 31 7 14 21 28 5	1 8 15 22 29 6
1 8 15 22 29 5 sun 30	2 9 16 23 30 6	Tue 3 10 17 24 31 7	Wed 4 11 18 25 1 8 Wed 3	12 19 26 2 9 Per	6 13 20 27 3 10	7 14 21 28 4 11 sat 6	29 5 12 19 26 2 sun 28	30 6 13 20 27 3 Mon 29	Tue 31 7 14 21 28 4 Vol.	Wed 1 8 15 22 29 5	9 16 23 30 6	3 10 17 24 31 7	4 11 18 25 1 8 sat 3	26 2 9 16 23 30 Sun 25	Mon 27 3 10 17 24 1 Mon 26	Tue 28 4 11 18 25 2 Dec Tue 27	Wed 29 5 12 19 26 3 Wed 28	Thu 30 6 13 20 27 4 bel Thu 29	Fri 31 7 14 21 28 5	1 8 15 22 29 6
1 8 15 22 29 5 sun 30 7	2 9 16 23 30 6 Mon 1 8	Tue 3 10 17 24 31 7 Oct Tue 2 9	Wed 4 11 18 25 1 8 Wed 3 10	5 12 19 26 2 9 Thu 4 11	6 13 20 27 3 10 Fri 5 12	7 14 21 28 4 11 sat 6 13	29 5 12 19 26 2 sun 28 4	30 6 13 20 27 3 Mon 29 5	Tue 31 7 14 21 28 4 VOV Tue 30 6	Wed 1 8 15 22 29 5 Vem Wed 31 7	2 9 16 23 30 6 Thu 1 8	3 10 17 24 31 7	4 11 18 25 1 8 Sat 3 10	26 2 9 16 23 30 Sun 25 2	Mon 27 3 10 17 24 1 Mon 26 3	Tue 28 4 11 18 25 2 Dec Tue 27 4	Wed 29 5 12 19 26 3 Wed 28 5	Thu 30 6 13 20 27 4 ber Thu 29 6	7 14 21 28 5 Fri 30 7	1 8 15 22 29 6 sai 1 8
1 8 15 22 29 5 sun 30 7 14	2 9 16 23 30 6 Mon 1 8 15	Tue 3 10 17 24 31 7 Oct Tue 2 9 16	Wed 4 11 18 25 1 8 Wed 3 10 17	Thu 5 12 19 26 2 9 Der Thu 4 11 18	6 13 20 27 3 10 Fri 5 12 19	7 14 21 28 4 11 Sat 6 13 20	29 5 12 19 26 2 sun 28 4 11	30 6 13 20 27 3 Mon 29 5 12	Tue 31 7 14 21 28 4 Wov Tue 30 6 13	Wed 1 8 15 22 29 5 5 /em Wed 31 7 14	9 16 23 30 6 Thu 1 8 15	3 10 17 24 31 7 Fri 2 9 16	4 11 18 25 1 8 sat 3 10 17	26 2 9 16 23 30 Sun 25 2	Mon 27 3 10 17 24 1 Mon 26 3 10	Tue 28 4 11 18 25 2 Dec 27 4 11	Wed 29 5 12 19 26 3 Wed 28 5 12	Thu 30 6 13 20 27 4 bel 7 Thu 29 6 13	5 Fri 30 7 14 14 14 15 14 15 16 16 16 16 16 16 16 16 16 16 16 16 16	1 8 15 22 29 6 Sall 1 8 15
1 8 15 22 29 5 sun 30 7 14 21	2 9 16 23 30 6 Mon 1 8 15 22	Tue 3 10 17 24 31 7 Occ 2 9 16 23	Wed 4 11 18 25 1 8 etok Wed 3 10 17 24	12 19 26 2 9 Der Thu 4 11 18 25	6 13 20 27 3 10 Fri 5 12 19 26	7 14 21 28 4 11 8at 6 13 20 27	29 5 12 19 26 2 sun 28 4 11 18	30 6 13 20 27 3 Mon 29 5 12 19	Tue 31 7 14 21 28 4 Nov Tue 30 6 13 20	Wed 1 8 15 22 29 5 Vem Wed 31 7 14 21	Thu 2 9 16 23 30 6 Thu 1 8 15 22	3 10 17 24 31 7 Fri 2 9 16 23	4 11 18 25 1 8 Sat 3 10 17 24	26 2 9 16 23 30 Sun 25 2 9 16	Mon 27 3 10 17 24 1 Mon 26 3 10 17	Tue 28 4 11 18 25 2 Dec 27 4 11 18	Wed 29 5 12 19 26 3 Wed 28 5 12 19	Thu 30 6 13 20 27 4 bell Thu 29 6 13 20	Fri 31 7 14 21 28 5 7 Fri 30 7 14 21	1 8 15 22 29 6 Sall 1 8 15 22



What has been done before?

- Show that you know the area
 - At the very least, identify key papers to study
- Show that there is previous work to build on
 - Feasibility!
- Identify the gap you will fill
 - Show what is new
 - Feasibility!

Problem and **previous work** are key parts of the thesis too!



What is the planned approach?

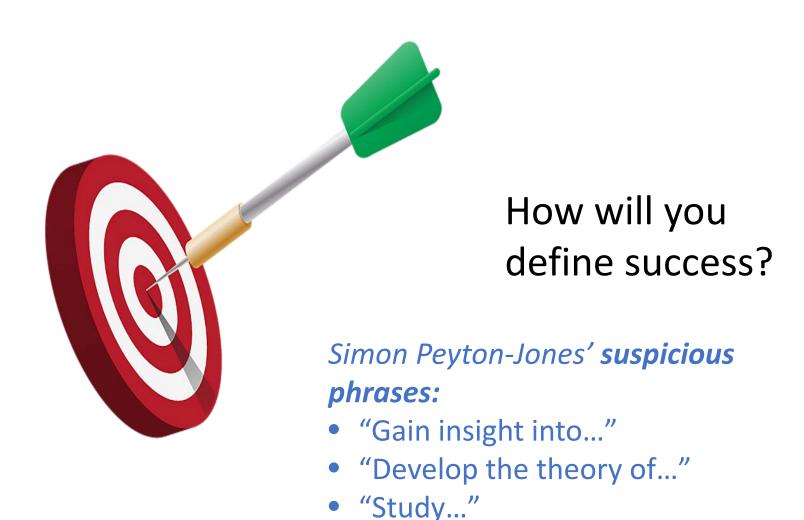


What is your idea?

What are the key challenges?

• Are there *initial results?*

How will you evaluate results?





What do we mean by improvement?

"Is it possible to improve Evosuite's test case generation by taking contracts into account?"



Why you?

Why are you the right person(s) for this project?

- What previous experience can you bring to bear?
 - Courses, obviously
 - Other relevant projects





What are the risks?

- Must you learn about an entirely new tool?
 - How long will that take?
- Are you relying on existing software?
 - What if it works less well than you expect?
- What if the problem is harder than you think?









What about IPR?



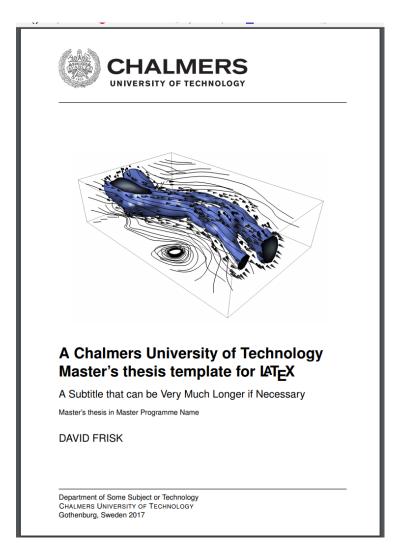


- e.g. source code
- Open source?

- Probably hope to use the results
- May give you private information



One more thing—the template





Contents

	st of Figures st of Tables Introduction 1.1 Section levels	Theory Methods Results Conclusion	
	1.2.1.1 Paragr .2.1.1.1.1	aph	1
2	Theory		3
3	Methods		5
4	Results		7
5	Conclusion		9
$\mathbf{B}_{\mathbf{i}}$	bliography		11



Deadlines

29 Nov

Submit first draft proposal

6 Dec

You get feedback from TAs

13 Dec

Deadline for complete proposal