Course design: Distributed Systems

Maarten van Steen

This document is written primarily for participants in the 2021 version of course "Distributed Systems," as well as anyone interested in the course setup. It aims at being fully transparent with respect to the design of the course, and what is expected from participating students. By sharing this document, I hope to make clear not only what the goals of the course are, but that each student understands why specific course elements have been chosen and what their role is in achieving those goals.

Background and motivation

Distributed computer systems are all around us: they appear as sensor networks; cloud computers; control networks in airplanes and automotives; smart-home installations, to name but just a few. The field of distributed systems is well established and in the last 30 years the body of knowledge has become considerable. Unfortunately, this body of knowledge has too often not led to simple designs that can be easily understood and assessed for their technical merits. On the contrary, we often witness new developments that easily catch the attention of the less experienced causing new waves of often unfounded excitement. Examples include public peer-to-peer systems, cloud systems, and more recently blockchains.

In this course, we ask ourselves whether these excitements are justified. The complexity of distributed systems often makes it difficult, even for the expert, to quickly come with the correct answer. Moreover, there is often no correct answer, as it all depends on the context in which a solution is deployed. The question then becomes how to get closer to investigating claims that accompany new developments. Are public peer-to-peer systems secure? Can we just migrate all our data and computations to cloud systems? Can blockchains really rule out the use of trusted third parties?

By looking at a number of specific cases, the course aims to improve critical thinking about distributed systems such that new developments can be better and more quickly assessed for their merits. Transferring what we already know about the field is less useful as it may be soon obsolete, especially when looking at what some consider important today. Preparing ourselves to still being able to pick up what pops up tomorrow is what matters more. This course is all about developing the cognitive skills to be prepared for the unknown in distributed systems. This brings us to the **intended course outcome**:

> *After the course the student is able to provide a scientifically sound and justified advice on a given (generally complex) distributed systems concept taking into account key concepts in distributed systems theory.*

Intended learning outcomes [ILO]

Within a period of ten weeks, you (the student) will be able to...

1. analyze the technical merits of specific developments within the field of distributed computer systems,
2. substantiate your analyses using existing and scientific documented knowledge,
3. further develop an attitude in which being able to clearly explain matters is geared to optimize the quality of feedback
4. clearly write up those analyses in a brief essay,
5. assess the substantiated analyses of others.

These five outcomes form essential elements of *critical thinking*, in this case for a specific subfield of computer science. Quoting from The Foundation for Critical Thinking:[1]

A well-cultivated critical thinker:

- *Raises vital questions and problems, formulating them clearly and precisely*
- *Gathers and assesses relevant information, using abstract ideas to interpret it effectively*
- *Comes to well-reasoned conclusions and solutions, testing them against relevant criteria and standards*
- *Thinks openmindedly within alternative systems of thought, recognizing and assessing, as needs be, their assumptions, implications, and practical consequences*
- *Communicates effectively with others in figuring out solutions to complex problems*

Critically thinking about distributed systems can be quite tough, as there may be many technicalities to discover that really do not matter for the essence. In this course, we make a first attempt to let you learn how to dig deeper without getting lost by those technicalities. Important aspects of critical thinking are *openmindness* and *effective communication*. This is reflected in ILO 3: you need to explain matters so clearly that the receiver is placed in a position to formulate relevant questions. Those questions, in turn, are needed to improve your own understanding of distributed systems.

Background on intended learning outcomes

The course offers a number of case studies of distributed systems that will need to be assessed for their technical merits by a team of 3-5 students. The assessment itself is eventually documented in the form of an essay, along with annotated presentations.

Bloom distinguishes the following levels of increasing cognitive development[2]:

1. **Remembering**: Recognizing or recalling knowledge from memory. Remembering is when memory is used to produce or retrieve definitions, facts, or lists, or to recite previously learned information.
2. **Understanding**: Constructing meaning from different types of functions be they written or graphic messages or activities like interpreting, exemplifying, classifying, summarizing, inferring, comparing, or explaining.

[1] https://www.criticalthinking.org/

[2] L.W. Anderson et al. (editors): "A Taxonomy for Learning, Teaching, and Assessing - A Revision of Bloom's Educational Objectives", Addision-Wesley (2001). Material is cited from https://thesecondprinciple.com.

3. Applying: Carrying out or using a procedure through executing, or implementing. *Applying* relates to or refers to situations where learned material is used through products like models, presentations, interviews or simulations.
4. Analyzing: Breaking materials or concepts into parts, determining how the parts relate to one another or how they interrelate, or how the parts relate to an overall structure or purpose. Mental actions included in this function are *differentiating, organizing, and attributing,* as well as *being able to distinguish between* the components or parts. When one is analyzing, he/she can illustrate this mental function by creating spreadsheets, surveys, charts, or diagrams, or graphic representations.
5. Evaluating: Making judgments based on criteria and standards through checking and critiquing. Critiques, recommendations, and reports are some of the products that can be created to demonstrate the processes of evaluation. *Evaluating* comes before creating as it is often a necessary part of the precursory behavior before one creates something.
6. Creating: Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing. Creating requires users to put parts together in a new way, or synthesize parts into something new and different creating a new form or product. This process is the most difficult mental function.

Although presented as a hierarchy, it is unclear to what extent one can speak of a linear dependency between these layers.

For this course, we focus first on layers 1 and 2, in which you are required to study existing material by yourself.

Next to layers 1 and 2, there is a strong focus on layers 4 and 5, yet particularly on 6. You are required to investigate a specific problem in distributed systems, assess proposed solutions, and come to a conclusion that addresses an initial question about that distributed system. By focusing on layers 4-6, the underlying idea is that you can more easily go through that process again when confronted with another problem in distributed systems. In other words, rather than learning what distributed systems are all about, you are being trained to critically think about distributed systems.

Place in the master curriculum "Computer Science"

The course is typical for the master program (now track) "Internet Science & Technology," which addresses questions such as whether the Internet can suddenly stop, or what the effects are of increasing demands on cloud services. Questions that are easily phrased yet difficult to answer. Any answer requires a thorough knowledge of what's going under the hood of networked computer systems, but even more an understanding of how all the various elements relate to each other.

The curriculum offers four mandatory courses:

- Mobile & Wireless Networks
- Internet of Things
- Internet Security
- Performance Evaluation

of which "Internet of Things" and "Internet Security" strongly link to the general topic of Distributed Systems. Of the electives, notably "Web Services," "Cloud Networking," and "Ubiquitous Computing" are relevant. What "Distributed Systems" offers in addition to these courses is a very strong emphasis on explicitly training critical thinking about networked computer systems. The course can be taken independently of each of the electives, although "Cloud Networking" forms notably good supplementary material. At the same time, it can be expected that students from, for example, "Embedded Systems" will participate.

By and large, the course is designed to accommodate a mixed group of students having different background knowledge. The course aims to offer enough challenging problems, which come from different corners of distributed systems, that such a diversity should not be a problem. The main emphasis remains on developing skills of Bloom's layers 4-6, which is largely independent of specific subjects. Also important: the course centers around enhancing skills and competencies concerning *critical thinking*: Is decentralization of computing really such a good idea? Can an open peer-to-peer system ever be secured? Are blockchains scalable? Are Tor networks the solution to anonymous browsing? Many of such deep-technical questions can be thought of. In virtually all cases a single answer is not available. A critical, substantiated assessment, however, can always be given. It does require developing a good understanding of the system at hand.

The course itself

Rationale

Distributed systems are often experienced as being complex. This is partly caused by the fact that so many aspects cannot be decently considered in isolation. For example, there is generally no single place where all security is dealt with. Likewise, communication between processes is everywhere, and different everywhere. Sometimes communication needs to be secured, sometimes not.

Teaching distributed systems is often equally complex, certainly when the teacher is not an expert in the field. This complexity can be overcome by providing views on distributed systems, an approach that Andrew Tanenbaum and I developed over a decade ago, leading to one of the few, often-used reference textbooks on distributed systems.[3] In addition, it helps a lot if students gain experience in actually programming (existing) components of distributed systems. Some universities offer lab-only courses on distributed systems in which students get to use the latest software. It is unclear how effective such lab-only courses are without courses that concentrate more on general principles and theory.

However, what is really missing in many, if not virtually all courses, is the development of cognitive competences for analyzing distributed systems. In this time and age, it is less important to learn facts about distributed systems except for those that are by now considered to belong to the core of the subject. Much more important is the development of skills by which new elements can be readily assessed for their technical merits.

For example, the introduction of blockchain has created an immense hype that is caused by the belief that such systems can support a myriad of applications now requiring a trusted third party but that such a party can be completely omitted if blockchains are used. At the same time, there is a lot of buzz as blockchains are often portrayed to be inherently scalable to thousands if not hundreds of thousands users. Unfortunately, both claims are very hard to make by looking at the methods and techniques underlying blockchains, and that it is questionable whether those techniques can be easily replaced by alternative ones that do support completely decentralized and scalable blockchains.

What this course aims for is bringing students to a level that they can critically assess such new developments.

Requirements

Improving competencies regarding assessing distributed systems requires knowledge of relevant, well-known methods and techniques. These underpinnings are all well documented, for example in textbooks. In general, the level at which the principles of distributed systems can be successfully taught is senior undergraduate, typically as part of a bachelor program in (technical) computer science. For this course, we assume students have knowledge of networked systems, typically by previously having taken a course on computer networks or an (introductory) course on distributed systems.

[3] M. van Steen, A. Tanenbaum, "Distributed Systems". 3rd edition, 2017. Previous editions were available through Pearson education, but after regaining the copyrights, we have made the book online available as well as in cheap print. Accessible through www.distributed-systems.net.

Course setup

The course covers a total of 9 weeks. During the first week, there will be lectures covering material from the book, yet in a condensed form. Each lecture will pay special attention to at least one, specific highlight of distributed systems. The remaining weeks you will be working in groups, digging into a specific topic of your choice.

Lectures

1. Introduction
2. Peer-to-peer computing
3. Edge computing
4. Virtualization
5. Pub-sub communication
6. Blockchains
7. Extra material (if needed)
8. Extra material (if needed)

Assignment

In groups of maximal three students, you will work on a topic of your choice. You will make a first attempt of critically assessing the merits of a specific (aspect of a) distributed system, first by studying the literature, and then writing down your findings in a brief essay and a presentation. More specifically:

After week 3:

- Each group will have studied relevant material from the literature and the book.
- Each group will have prepared an annotated group presentation.

After week 5:

- Each group member will individually work on a subtopic and prepare an annotated presentation to be presented to the other members. The main purpose of this step is to make sure that you understood your part of the topic your group is working on.

After week 7:

- Each group will have prepared an advice in the form of an essay
- Each group will have molded their advice in a presentation

After week 8:

- Each group will have studied the material from another group and prepared questions for an opposition

After week 9:

- Each group will have presented their material and defended it before their opponent group.
- Each student will act as a jury member for the defense/opposition, and will have read the relevant essays.

- Each jury member will have prepared at least two questions for the relevant essays.

Details will be provided on Canvas.

Essay

The essay addresses a *question* concerning an often controversial topic in distributed systems. The question can typically be seen as coming from the need for advice. You are asked to provide that advice.

An advice will need to be substantiated, requiring at least three components:

1. A clear description of the problem at hand.
   Simply posing the question is often not enough. A problem has a context from which it becomes clear that we are dealing with a relevant and interesting problem. That context needs to be clearly described, as well as the problem itself. A good descriptionn reflects that you have understood what the problem is all about.

2. Background information on what has already been studied by others.
   Many problems have already been addressed by others. With some luck, survey articles have been written that provide an overview of such work. Providing background information, which often consists of exploring a variety of solutions forms an essential part of the advice.

3. An analysis of the situation, which will eventually lead to an advice.
   Primarily based on background information, but also an analysis of the specific situation at hand, you will come to an advice concerning the stated problem. This analysis forms the core foundation of your advice.

The essay will typically encompass some 2500 words. Guidelines are available on Canvas.

Final grade

You will be required to attend the final plenary session, and to individually rank the presentations (including the discussions), and motivate your ranking. How this works will be There are thus four components:

- E: Essay (weight 50%) **Group grade**
- P: Presentation and interaction (weight 20%) **Group grade**
- Q: Individual questions, and motivation for those questions (weight 15%) **Individual grade**
- R: Individual ranking, and motivation for that ranking (weight 15%) **Individual grade**

The final grade is thus computed as 0,5*E + 0,20*P + 0,15*Q + 0,15*R.