V.1.3, 10th-September-2019

Study Guides on Computer Networks and Distributed Systems

Solutions to Selected Study Guide Exercises about the Introduction to Distributed Systems

Context

The Study Guides offered in the CN and DS courses aim to help students in their preparation of these subjects; they include context explanations, technology uses, comments about certain textbook sections and emphasize significant structures and concepts. Study Guides include a Questionnaire related to the topics included in each chapter and their Solutions are published shortly after the Questionnaire publication date.

The Powerpoint presentations that we use in lectures convey concepts that are to depict on the board and usually summarize the essential concepts and articulate them as well. Grasping the concepts and structures conveyed by the presentation slides entails the student's scanning the relevant textbook sections and their class notes, without which, the presentations will result awkward and too contrived.

The nature of your work in this introductory chapter consists, among other things, in reading it and skimming the chapters that expand the relevant concepts, for example, in pg. 2 in this study guide, the *no global clock* characteristic of DS should lead you to skim chapter 14 of the Distributed Systems textbook (Dollimore *et al.*) where you will find the aspects of DS related to time. Particularly, when a concept might sound contrived, it is very advisable to deepen your study about it a little by using other sections of the book until you attain a reasonable grasp of it, that is, one that allows you proceed with the ensuing sections. Examples usually result very helpful.

The terminology of Distributed Systems (DS) used in this chapter is very broad, one of the objectives of this chapter is to be able to understand the meaning and the significance of each concept in context, for example, *concurrency* in DS is normally understood as *parallelism* since the programs that work on shared data structures usually run on physically separated computers (Networked computers).

Reviewing the course on CN as a basis for this course

The main objective a of a Distributed System (DS) consists of *resource sharing*. That is stated in section 1.4 of the DS textbook, where we are offered a few of the most used concepts in DS: service, server, client, process. For review and for further study, if necessary, we advise you to explore also the textbook by Peterson and Davie (Computer Networks, Ed.5) which presents these important concepts in chapter 1. Some of the DS challenges explained in section 1.5 are known from our course on Computer Networks, specifically we have already studied the error detecting and correcting codes, the concept of scalability (Scalable connectivity in CN), Quality of Service, but, others will be brand new: heterogeneity, transparency and concurrency.

Internet is the ground upon which Distributed Systems are built and operated, then, a firm grasp of Computer Networks is a must for safely undertaking the study of Distributed Systems. Of particular importance is the concept of Network Architecture since, as said above, distributed systems assume a fully functional Internet. In fact, the style of thinking of Computer Networks is also applied in Distributed Systems, though greatly expanded. What does then that style of thinking consist of? Maybe it has a name of its own? Is that style of thinking exclusive of Computer Networks? Why is it important that I become aware of that would-be style of thinking?

The style of thinking we refer to is one based on thinking with abstractions, which is nothing new for you, except for the fact that in Computer Network Architecture and in Distributed Systems Architecture, the application of it is particularly broad. As we explained in the initial Lecture and Lab sessions, Distributed Systems are comprised of Computer Systems interconnected by means of Internet which enables the applications running in them to coordinate, collaborate and share resources. If Computer Systems are complex, imagine the complexity emerging when they are set to communicate, and the possibilities opened by that complexity. Complexity can be leveraged when properly managed, this is exactly what has happened in the Internet after the emergence of Google, Amazon WS and Spotify to name a few.

The two Network Architectures that we studied in CN constitute the basis for DS. The functions of a computer network is organized into four layers in the Internet Architecture (IA) and into seven in the case of the OSI reference model. In most situations we will use the IA whereas in particular developments of DS we will refer to the OSI reference model. You can refer to the "Weekly Questionnaires and Solutions" section in http://paloalto.unileon.es/ds where you will find reviews about network architectures. Also, we recommend that you scan slides from 19 through 23 where you will find a valuable summary of these precise concepts: architecture, layer, service, service interface, peer-to-peer interface (protocol) and protocol stack. We suggest that you respond to the following questions about network architectures relevant to distributed systems as you conduct your review of CN:

(The review questions about CN are to be solved by students, any questions that might arise in the solving of those questions can be sent to foces.informatica.unileon@gmail.com or commented in a tutor teaching session)

1. Explain the difference between a protocol stack and a network architecture.

2. What's the technical name of the *service interface* to the UDP protocol? What do you think a *service interface* comprised of?

3. What's ssh, is it an application or maybe is it a protocol, or maybe both?

4. List the services provided by the UDP and the TCP protocols.

5. Which layer does the ICMP protocol belong to?

6. In this question we are listing a series of functionalities that we would wish a network to offer, your work consists of mapping each functionality to the layer in the OSI architecture that is responsible for implementing it, if any and including a brief explanation about your choice:

   a. Load balancing

   b. Main memory error that causes a host Operating System to crash

   c. A bus error in an application that causes an application implementing an Application-layer protocol to crash

   d. Transmission errors

   e. The physical clock of a host that is not properly synchronized with a quality clock source

   f. An application server that cannot cope with the current service demand

   g. Packets lost due to Internet congestion

   h. Sending the packets onto the next hop that belongs to the least-cost path

Questionnaire about the first lecture on Distributed Systems

1. Which DS transparencies apply to the TCP protocol?

   Amongst the most representative transparencies about TCP we could list the *failure transparency* since TCP hides omission faults, *i.e.,* if some segment is lost, TCP will enact its Timeout and Retransmission mechanisms to recover from the fault without the C/S applications becoming aware of that fact whatsoever. Other transparencies applying to IP might equally be applied to TCP such as the access transparency. By contrast, the location transparency cannot apply to TCP since the establishment of a new TCP connection needs a specific destination address.

2. Can you explain what the *end-to-end* argument is in networking and distributed systems? Do you think that TCP represents the essential idea in that argument? Consult the textbook by Peterson and Davie, chapter 1.

   In networks and distributed systems, the so called *end-to-end argument* deals about whether certain functionalities should be implemented in the network or in the network end systems. For example, over the years, network technology improvements have made it recommendable to implement packet retransmissions in the end hosts instead of in the intermediate router interfaces. TCP does represent this philosophy.

3. Can distributed systems be built upon *non-reliable* IP infrastructures?
   Distributed systems can be successfully deployed upon IP infrastructures because the network and the application architecture offset the lack of reliability inherent to the infrastructure. For example, where errors might take place, the protocols detect them and compensate for their effects by dropping the involved PDUs, thereby forcing the transport protocol to retransmit them.

4. Which one of the following is the main **goal** of Distributed Systems?
   ○ Concurrency
   ○ Lack of a global clock
   ○ Resource sharing
   ○ That failures be independent

5. The following definition of Distributed System is from Leslie Lamport's paper titled Time, Clocks and the Ordering of Events in a Distributed System: *"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 system."*

   a. According to the preceding definition, what is the most important characteristic of a Distributed System?
      "If the message transmission delay is not negligible…"

   b. What is the most uncertain aspect of a Distributed System included in this definition?
      The disparity between transmission delay and the time between events in the DS

   c. What are the essential three key words in the definition?
      1. Processes
      2. Messages
      3. Delay

6. How do the components of a DS coordinate their actions?

- ○ By sharing a logical clock
- ○ By sharing an atomic-precision clock
- ○ [✓ By exchanging messages]

7. The components of a DS are installed in:
   - ○ Isolated computers
   - ○ [✓ Networked computers in a LAN with TCP/IP]
   - ○ Networked computers in any type of internetwork

8. What are the differences between mobile and ubiquitous computing?

   · Mobile: Any host moving to a foreign network communicates over the Internet as though it were located in its home network.

   · Ubiquitous: Networks comprised of a large number of small and low-power devices generally communicating over wireless networks. The Internet of Things corresponds to this computing model.

9. You are asked to provide a TCP-based network service in your organization's network and the server location is to be found out by using UDP, devise a way to do that and explain it with detail.

   The server location could be discovered by having the client send a broadcast datagram containing a request for the location of the server.

10. What's the difference between an open system and a open-source system? Provide one example of each.

    We state a system is open if the access interfaces it offers are open, *i.e.,* the implementation may be kept private, but the access for using it is open. The system calls of the HP-UX system constitute an open interface for accessing the HP-UX kernel resources. The kernel internals need not be known in order for programmers to use its resources, though.

    The source code comprising an open-source system is public thereby granting everyone not only the right to use the system but to extend it and to study it. OpenBSD and GNU-Linux are open-source systems.

11. Choose three aspects of a DS that must be transparent to the application programmer and explain why it is convenient that they remain transparent.

    1. That the storage elements be replicated, so they offer an improved availability but that the replication technologies remain hidden so that programmers will not have to consider it.

    2. That when an application accesses a shared resource in the distributed system, the programmers won't have to implement their programs considering whether or not the resource can be accessed in concurrency with other applications.

    3. That when any failing network in the path from a client to a server won't hamper the communication by providing network redundancy and several capabilities of failure detection, etc.

12. Contrast the concepts of concurrency and parallelism. Discuss whether the Java language offers support for either.

    Concurrency refers to a computing system in which it is possible to have several *threads of execution* executing a single program. When those threads may execute that single program at exactly the same time, we state the computing system is multi-threaded and parallel. A Linux system executing on a single-core microprocessor is a concurrent system; the same Linux system, running on a 4-core i5 microprocessor is a parallel system.

13. Why is QoS important in the design of present-day Distributed Systems?

    QoS stands for Quality of Service. That distributed systems provide QoS is essential in the Internet of today since the services that are based on it have a real time character, for example, all the multimedia networking applications such as video conferencing.

14. List the most important transparencies that apply to the Mobile IP technology

    Access and location.

15. Solve the following exercises from DK (Coulouris, Dollimore, Kindberg, Blair)

    Ex 1.4:

    The PDA (Or a present-day smartphone) might submit a request to the broadcast address over the wireless network so that a listening server could provide the specific location of a server containing the relevant information.

    Ex 1.5:

    C/S server computing *vs.* Cloud Computing: Cloud computing is much more general than C/S. Cloud computing employs the C/S model continually, every time a client accesses any of the varied services provided in the Cloud.

    Ex 1.11:

    · The client process itself might undergo a memory access failure
    · The client's access network's router might fail
    · The server process, running in a internet host might undergo a denial of service attack

16. Build a simple table containing the distributed systems transparencies explained in the lecture along with a short explanation of each or a relevant example

    Complete this response on your own so that you practice with the concepts.

17. Find out the basic features of the Hadoop distributed technology and highlight its most important DS transparencies

    · When upgrading Hadoop we have the guarantee of API compatibility, data compatibility, and wire compatibility. This means Hadoop upgrades result transparent to the APIs used by the programmers, to the data formats and to the protocols used for accessing the data. Hadoop will resolve any incompatibilities in the upgrade process.

· Hadoop's file system (HDFS) clients programs are configured to offer *namenode failover* which results transparent to the file systems users.

· Hadoop offers the following transparencies *with different degrees*: Access, location, concurrency, transaction, failure, scaling and performance (In a limited way), etc.