Un Curso de Principios de Computación Distribuida

(Versión Preliminar)

Ricardo Marcelín*   Sergio Rajsbaum†

Mayo 28, 1996

Resumen

En este trabajo presentamos el curso *Principios de Computación Distribuida* dirigido a alumnos avanzados de licenciatura y de maestría de la UNAM, en el semestre enero-junio de 1996. Describimos el temario, la bibliografía y las tareas, así como las experiencias con los alumnos. Discutimos la filosofía general del curso, que sigue un enfoque específico de computación.

1 Introducción

En este trabajo presentamos el curso *Principios de Computación Distribuida* dirigido a alumnos avanzados de licenciatura y de maestría de la UNAM, en el semestre enero-junio de 1996. Describimos el temario, la bibliografía y las tareas, así como las experiencias con los alumnos. Discutimos la filosofía general del curso, que sigue un enfoque específico de computación.

La computación distribuida ha ido tomando más y más importancia, con el impresionante desarrollo de las telecomunicaciones y conforme los avances tecnológicos han hecho posible la construcción de computadoras que caben en un escritorio, pero con procesadores muy poderosos y grandes capacidades de memoria y disco. Millones de usuarios dependen de sistemas distribuidos diariamente para hacer transacciones bancarias, reservaciones de vuelos, telefonía, enviar correos electrónicos, obtener información de todo tipo y realizar operaciones de compraventa.

*Depto. de Ing. Eléctrica, UAM-Iztapalapa. Email: `calu@xanum.uam.mx`.

†Instituto de Matemáticas, UNAM, Ciudad Universitaria, D.F. 04510, México. Email: `rajsbaum@servidor.unam.mx`. Apoyo de DGAPA y Conacyt.

Entendemos a la computación distribuida en un sentido muy amplio, incluyendo a cualquier sistema en el que múltiples agentes autónomos, cada uno con capacidades de cómputo individual, se comunican entre si y afectan mutuamente su comportamiento. Los agentes, usualmente llamados *procesadores, procesos* o nodos, pueden ser desde computadoras completas hasta autómatas celulares con capacidad de cómputo y memoria muy limitados. Se pueden comunicar mediante mensajes o memoria compartida.

Se pueden resaltar las siguientes peculiaridades del tema:

1. Es un área de cambios constantes, debido al vertiginoso avance de tecnologías, tanto de hardware como de software; de nuevas ideas, modelos conceptuales y herraminetas matemáticas.
2. La complejidad de un sistema distribuido que abarca gran cantidad de problemas de muy diversos tipos.

Sin embargo, existe ya un cuerpo de principios fundamentales, de ideas básicas subyacentes que permiten entender muchos de los aspectos de un sistema distribuido. El objetivo de este curso es introducir al alumno a modelos, técnicas y conceptos relevantes, independientemente de las tecnologías del momento. Desarrollar en el alumno una capacidad de análisis, creatividad y razonamiento que lo ayuden a atacar problemas de tipo *distribuido.*

El temario del curso trata de identificar aspectos escenciales de lo que es un problema distribuido. Intenta ser una presentación coherente, *continua*, buscando que los temas presentados parezcan suceder unos a otros de manera natural. Más que aspirar a un alcance en cantidad de los temas presentados, es decir, de *información*, el curso intenta desarrollar en el alumno *habilidades.*

Finalmente, se intentó elegir temas que se entendieran ya lo suficientemente bien, para describirlos mediante modelos matématicos elegantes. Se omitieron algunos resultados o problemas importantes, cuyas técnicas parecen ser algo particulares para el problema.

Resumiendo, las ideas claves de la filosofía del curso son: principios fundamentales, continuidad en una visión coherente, desarrollo de habilidades creativas y de análisis, modelos matématicos elegantes. Son una consecuencia de esta, los siguientes puntos.

- Las tareas tienen un papel fundamental. Constan de problemas que requieren de un análisis serio por parte del alumno, de una dedicación y esfuerzo especiales. No hubo en esta ocación ejercicios de programación o familiarización con software.
- El curso se dirige al alumno que aspira a una carrera profesional de desarrollo tecnológico o investigación, y que ya tiene una madurez sustancial en computación.

Se enfatiza que en un solo semestre, fue posible incluir solamente una fracción de los principios de computación distribuida importantes; los restantes deberían cubrirse en otros curso(s) , que además podría(n) tener un carácter mas avanzado.

Así mismo, el curso se debe complementar con otro(s) que describan los sistemas distribuidos existentes, los problemas de desarrollo de software e implantación, aspectos relacionados con lenguajes, seguridad y criptología.

2 El Area y Material Existente.

Hace ya bastante tiempo que existen libros y se imparten cursos relacionados a sistemas distribuidos, con un enfoque que estudia los aspectos de *arquitectura,* por ejemplo, [24, 25, 23]. Es decir, la organización en modulos de acuerdo a su función. Muy comunes son los cursos con este enfoque de redes de computadoras, sistemas operativos y bases de datos distribuidas. En los últimos diez años, se ha desarrollado el enfoque que sigue este curso. Un enfoque que intenta construir una teoría, un enfoque mas *algorítmico,* que estudia la manera de resolver problemas específicos, las capacidades y limitaciones fundamentales de los sistemas distribuidos.

Aunque en mucho menor cantidad que en el enfoque arquitectural y de sistemas, ya se imparten cursos de principios de computación distribuida en decenas de universidades en el mundo. Existen dos congresos anuales: en norteamérica, Principles of Distributed Computing (PODC) cuyas memorias son publicadas por la ACM, y en Europa, Workshop on Distributed Algorithms (WDAG) cuyas memorias son publicadas por Springer Verlag, en la serie *Lecture Notes in Computer Science.* Y otros congresos dedicados a aspectos más específicos, como Conference on Concurrency Theory (CONCUR), Parallel Algorithms and Architectures (SPAA), Real-Time Systems Symposium (RTSS); o mas generales, como Symposium on the Theory of Computing (STOC) y Foundations of Computer Science (FOCS). Revistas como *Distributed Computing, SIAM Journal on Computing, Information Processing Letters, Information and Computation* y *Journal of the ACM* incluyen artículos de esta área regularmente. Recientemente se han publicado algunos libros: [17, 21, 26] y artículos panorámicos [14, 2].

Nuestro enfoque es similar, y debe mucho al curso de Hagit Attiya en el Technion, Israel, descrito en sus notas [1], al de Nancy Lynch en MIT, descrito en su reciente libro [17], a cursos de Adrian Segall y Shmuel Zaks en el Technion, y a los libros de Michel Raynal [21], y de Gerard Tel [26].

Sin embargo, ninguno de los libros anteriores parece adecuarse directamente a las circunstancias de la UNAM. Los libros [1, 17], son demasiado avanzados o extensos. De un nivel más adecuado son [26, 21], pero [21] es menos actualizado. Todos incluyen material para al menos dos semestres (quizás a exepción de [21]). Además, no incluyen el mismo material, lo cual parece indicar la falta de acuerdo total de cuales son los resultados fundamentales. La elección de los temas del curso y la formación de los estudiantes, hizo que fuera conveniente basarse en distintas ocaciones en distintos libros, y también en artículos, como se describe en la siguiente sección.

3 Objetivos y Filosofía del Curso.

En esta sección describimos los objetivos generales de aprendizaje que se espera logren los alumnos, y la filosofía general que nos guio en el diseño del temario y tareas (descritos en secciones subsiguientes).

Podemos resumir los objetivos que se discuten a continuación utilizando la clasificación de Bloom . En el área *cognocitiva,* comprender algunos de los problemas y técnicas importantes de la computación distribuida. En el área *afectiva,* desarrollar una actitud científica y crítica; del gusto por entender a fondo. En el área *psicomotriz,* desarrollar habilidades y destrezas que permiten analizar y resolver problemas complejos.

3.1 Objetivos Principales.

Consideramos de primera importancia formar alumnos que puedan resolver problemas complejos de sistemas distribuidos, así como alumnos con capacidad crítica y creadora. Creemos que la manera de lograr esto es: (a) logrando un entendimiento a fondo, y (b) aprendiendo a resolver problemas difíciles; enfocándose en material escencial, en principios fundamentales subyacentes no solo a las tecnologías del momento, sino a cualquier otra que aparezca en el futuro. Esto es especialmente importante en vista de las dos siguientes peculiaridades de esta área.

Primero, es esta un área de cambios constantes, debido al vertiginoso avance de tecnologías, tanto de hardware como de software: las redes de computadoras cambian constantemente, incorporando lineas de comunicación más rápidas cada vez, y de nuevos tipos, como fibra óptica y redes móviles de radio. Nuevas arquitecturas como ATM, sistemas cada vez más sofisticados de televisión interactiva, telefonía celular, comunicadores personales *pager* que pueden transmitir además de recibir. Multiprocesadores más poderosos, redes de estaciones de trabajo. Se generan constantemente nuevas ideas, y modelos conceptuales que cambian nuestra manera de ver el cómputo y manjeo de información distribuida; quizás el ejemplo más dramático es *web,* pero existen otros como las listas de correo y noticias electrónicas, nuevos lenguajes y paradigmas, que generan nuevos problemas facinantes y que cambian las suposiciones de problemas conocidos (por ejempo, la relación entre la velocidad de procesamiento y la de transmisión). Entre tanto, se descubren nuevas y más poderosas herramientas matemáticas para atacar los problemas, como: probabilidad, algebra, combinatórica, y topología.

Segundo, la enorme complejidad de un sistema distribuido, que se aprecia al detenerse a pensar en la gran diversidad de problemas que abarca, entre los cuales estan los siguientes. Aspectos de comunicación: códigos para detección de errores, protocolos de enlaces, problemas de ruteo, congestión, ATM, etc. Sincronización: exclusión mutua, relojes, abrazos mortales, etc. Manejo de procesos: administración de recursos, threads, tolerancia a fallas, tiempo-real, calendarización, etc. Seguridad y criptología. Uso compartido de memoria. Especificicación, modelado y verificación. Complejidad y análisis de algoritmos. Problemas que pueden observarse en sistemas de archivos, bases de datos y sistemas operativos distribuidos. Redes de interconexión: hipercubo, mariposa, *fat-tree,* CCC, etc.

3.2 Visión Coherente

En un solo semestre es imposible presentar todos los principios fundamentales del área, debido a la gran cantidad que existe, y a que algunos de ellos son demasiado avanzados para presentarse en un primer curso. Además, reiteramos que, siendo un área tan nueva, todavía no hay un consenso absoluto de cuales son los resultados mas importantes.

Se eligió el material de manera que hubiera alguna relación entre los distintos temas, presentando una visión general lo más coherente posible.

3.3 Principios Matemáticos

Intentamos elegir material que se pudiera modelar matematicamente, con fundamentos sólidos. Suponemos que esto es un indicativo de que tan fundamental es el material; que material de este tipo tiene más posibilidades de ser de utilidad para el alumno en el uso y diseño de mayor variedad de tecnologías. Y finalmente, que contribuye a la formación y madurez académica general del alumno, así como a su cultura matemática, que es de utilidad en ciencias de la computación en general.

Incluimos material que requiere de conocimientos de teoría de gráficas y combinatórica, como: árboles, busquedas DFS, conexidad, distancias más cortas, topología combinatórica.

4 El Temario.

El curso está dividido en tres partes. En la primera se presentan algoritmos y técnicas básicas de desarrollo y análisis de algoritmos distribuidos. El paradigma subyacente es el de *difusión de información* en una red. Se enfatizó el modelo de envío de mensajes, y la complejidad de comunicación (cotas superiores, y muy poco de inferiores, sobre el número de mensajes enviados por un algoritmo.

En la segunda parte se estudiaron problemas de tiempo y sincronización. Aqui las ideas pasan a ser más abstractas y generales, después de haber estudiado diversos algoritmos concretos en la primera parte. Sin embargo, el paradigma de difusión de información sigue siendo fundamental; el enfoque de complejidad es sustituido en esta parte por el de análisis. Se describe su utilidad para comprender algoritmos de detección de estado global, de sincronización de relojes, y de causalidad. Aunque el modelo sigue siendo el de envío de mensajes, en esta parte ya no es escencial. La mayor parte de las ideas se aplican igualmente al modelo de memoria compartida. De tal manera que esta segunda parte sirve de transición entre las otras dos partes del curso.

En la tercera parte, se introduce el modelo de memoria compartida. Una razón de estudiar este modelo es que es un área de intensa investigación y de muchas aplicaciones. Otra razón es de que, para cierto tipo de problemas, es un modelo más apropiado. De hecho, es un modelo muy conveniente para estudiar problemas de tolerancia a fallas (benignas), así se aprovecha para introducir al alumno a este importante aspecto de los sistemas distribuidos (que no había sido tocado antes). En esta parte se comienzan presentando varios problemas y algoritmos que los resuelven, encaminándose hacia el problema fundamental de tolerancia a fallas, *consenso.* Se describe la estructura topológica del cómputo libre de espera, y a partir de esto se demuestra la imposibilidad de resolver el problema de consenso.

Temario

- Presentación del Curso
- Primera Parte
  1. Modelo de un sistema distribuido.
  2. Protocolos de enlaze.
  3. Difusión de información por ficha. Busqueda en profundidad (DFS) [4].
  4. Conexidad y certificados; más algoritmos de tipo ficha [8].
  5. Algoritmo de difusión sin respuesta (PI) y con respuesta (PIF) [22].
  6. Elección de líder, en anillos y en gráficas arbitrarias [1, 12].
  7. Arbol generador de peso mínimo [10].
- Segunda Parte
  1. Causalidad, relojes escalares y relojes vectoriales [13, 20].
  2. Detección de estados globales [7].
  3. Tiempo real y sincronización de relojes [18, 19].
- Tercera Parte
  1. Modelo de memoria compartida.
  2. Problemas de consenso solucionables (*consenso seguro* [5, 16]) y no solucionables.
  3. Problema de los conjuntos ordenados y el modelo de la foto atómica inmediata [5, 6].
  4. Imposibilidad de consenso y estructura topológica del cómputo libre de espera [3, 5, 9].

5 Tareas y Evaluación

Hubo cinco tareas: dos referentes a la primera parte (una de ellas dedicada al problema del árbol generador de peso mínimo, y la otra al resto), y una a cada una de las otras partes. Incluyeron ejercicios sencillos para ayudar a asimilar el material visto en clase, y más complicados para desarrollar habilidades de análisis y solución de problemas complejos.

Se evaluó al alumno tomando en cuenta su desempeño en las tareas y en la clase.

6 Conclusión.

Esta es la primera ocasión que el curso se imparte. El profesor titular fue Sergio Rajsbaum y el ayudante fue Ricardo Marcelín. Estuvieron inscritos en el curso tres alumnos de licenciatura, y cuatro de maestría. Las clases fueron los martes y jueves de 12:30 a 2:00PM, desde el 30 de enero hasta el 30 de mayo de 1996, sumando un total de 33 sesiones.

Siendo esta la primera vez que se imparte el curso, está sujeto a muchas mejoras. Sentimos que es posible distribuir mejor el tiempo dedicado a cada clase. Mejor balance entre los temas de envío de mensajes y de memoria compartida (que es ahora investigado tan intensamente como el de mensajes). Idear tareas que sean de mayor provecho para el alumno. Quizas requerir de un trabajo final o de experimentación con sistemas existentes (ya sea reales o simulados). Sin embargo, no es claro como lograr esto, sin sobrecargar demasiado a los alumnos, que ya así parecieron estar bastante ocupados.

En general los alumnos parecieron responder bien, aunque tuvieron dificultades: los de licenciatura, con mejor formación matemática; los de maestría con mejores bases de computación. Mi impresión es que un problema común a todos, son bases débiles de conceptos e ideas básicas de ciencias de la computación (e.g. complejidad y análisis de algoritmos; reducir un problema a otro), y falta de habilidad para resolver problemas difíciles.

Al final del curso se entregó a los alumnos un cuestionario de evaluación donde se les pedía su opinión con respecto a el material expuesto, la exposición, las tareas, la manera de calificar (no hubo exámenes), el apoyo al alumno (e.g. apoyo bibliográfico), y si estarían interesados en un segundo curso. El gusto por el material fue unánime, y excepto uno, todos estarían interesados en un segundo curso. Por otro lado, expresaron su dificultad con las matemáticas, y la falta de material de consulta.

Se afirma [15] que una experiencia de aprendizaje es real, solo si las actividades propuestas responden a los intereses de los destinatarios. Las actividades del sujeto adquieren sentido cuando lo que aprende es valorado. Solo un alumno expresó sentimientos en el sentido de que el material es demasiado teórico.

Esperamos que conforme el curso se vaya perfeccionando, logre cada vez mejor llegar a lo que es el aprendizaje. Se dice [11] que aprender es transformar la disposición analítica del individuo, su actitud crítica y reflexiva; desarrollar habilidades prácticas para comprender y transformar su entorno; apropiarse de recursos para enfrentar adecuadamente los problemas que la situación impone.

References

[1] Hagit Attiya. *Distributed Algorithms, Lecture Notes for Course 236357 (Spring 1993)*. Department of Computer Science, Technion, 1994.

[2] Hagit Attiya. *Theory of Distributed Computing*. Chapter 5 in The Handbook of Parallel and Distributed Computing, A. Zomaya (ed.), McGraw-Hill, December 1995.

[3] Hagit Attiya and Sergio Rajsbaum. A combinatorial topology framework for wait-free computability. In *Proceedings of the 10th International Workshop on Distributed Algorithms*, pages 321–343, 1996.

[4] Baruch Awerbuch. A new distributed depth-first-search algorithm. *Information Processing Letters*, 20:147–150, April 1985.

[5] E. Borowsky and E. Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, pages 91–100, May 1993.

[6] E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing*, pages 41–51, August 1993.

[7] K. M. Chandy and L. Lamport. Distributed snapshots: Determining global states in distributed systems. *ACM TOCS*, 3(1):63–75, 1985.

[8] Shimon Even, Gene Itkis, and Sergio Rajsbaum. On mixed connectivity certificates. In P. Spirakis, editor, *Lecture Notes in Computer Science*, number 979, pages 1–16. Springer-Verlag, 1995.

[9] M. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of distributed commit with one faulty process. *Journal of the ACM*, 32(2), April 1985.

[10] R. Gallager, P.A. Humblet, and P.M. Spira. A distributed algorithm for minimum weight spanning trees. *ACM TOPLAS*, January 1983.

[11] Juan Luis Hidalgo Guzmán. *Aprendizaje Operatorio*. Ed. Casa de la Cultura del Maestro Mexicano, 1992.

[12] D.S. Hirschberg and J.B. Sinclair. Decentralized extrema-finding in circular configurations of processes. *Communications of the ACM*, 23:627–628, 1980.

[13] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Comm. ACM*, 21(7):558–564, July 1978.

[14] Leslie Lamport and Nancy Lynch. Distributed computing: Models and methods. *In Jan Van Leeuewn, Ed, Handbook of Theoretical Computer Science*, B, 1990.

[15] Antoine Leon. *Pedagogía de Adulto*. Ed. Trillas, 1981.

[16] N. Lynch and S. Rajsbaum. On the borowsky-gafni simulation algorithm. In *Proceedings of the 4th Israel Symposium on of Theory of Computing and Systems*, June 1996.

[17] Nancy Lynch. *Distributed algorithms*. Morgan Kaufman Pub., 1996.

[18] David L. Mills. On the chronology and metrology of computer network timescales and their application to the Network Time Protocol. *ACM Computer Communications Review*, 21(5):8–17, October 1991.

[19] B. Patt-Shamir and S. Rajsbaum. A theory of clock synchronization. In *Proceedings of the 1994 ACM Symposium on Theory of Computing*, pages 810–819, May 1994.

[20] M. Raynal and M. Singhal. Logical time: Capturing causality in distributed systems. *IEEE Computer*, pages 49–56, February 1996.

[21] Michel Raynal. *Networks and distributed computation*. MIT Press, 1988.

[22] Adrian Segal. Distributed network protocols. *IEEE Transactions on Information Theory*, IT-29(1):23–35, January 1983.

[23] M. Sloman and J. Kramer. *Distributed Systems and Computer Networks*. Prentice-Hall, 1987.

[24] Andrew S. Tanenbaum. *Computer networks*. Prentice-Hall, 1988.

[25] Andrew S. Tanenbaum. *Distributed operating systems*. Prentice-Hall, 1995.

[26] Gerard Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 1994.