Sistemas Distribuidos

Leandro R. León

Cemisid

Cecalcula

Postgrado de Computación

Informaciones generales

Vuestro Profesor: Leandro R. León

email: lrleon@cemisid.ing.ula.ve

Horario de clases: Miércoles y Jueves de 4pm a 6pm

Consultas

- martes y viernes de 10:30am a 11:30am
- En el Cemisid
- Otras consultas: previa cita

¿Qué es un sistema distribuido?

La definición más aceptada:

"A distributed system is a system where I can't get my work done because a computer has failed that I've never heard of"

-- Leslie Lamport

Breve introducción

- ¿Qué es la computación distribuida?

  Interacción cooperativa entre dos o más computadoras interconectadas de alguna manera para solucionar un problema específico

⇒Tratar con dos problemas principales :

- Cómo se comunican las computadoras (redes)
- **Cómo interactúan, cooperativamente, las computadoras (sistemas distribuidos)**

Contexto actual

- Computación es parte de la vida
- Hardware
  - Computadores personales de alto rendimiento (RISC, mucha memoria, grandes caches, grandes discos)
  - Estaciones de trabajo de alto rendimiento
  - Computadores paralelos son acequibles
- Redes
  - De alto rendimiento
  - Heterogéneas
  - Ubicuas (Internet)
- Algunos sistemas distribuidos
  - WEB
  - Reservaciones aéreas
- Indispensable la computación distribuida

La ley de Moore y el futuro

- Aumento densidad lógica de circuito integrado $2^{(t - 1962)}$ "2
- 37 años de verificación: 100X rendimiento/década
- Límites cuantitativos:
  - Memoria un electrón temperatura ambiente
  - Tamaño de celda de memoria: 7 nm^2
  - 1 $cm^2 \Leftrightarrow$ 250 GB
  - Para el 2018: 8MB x 10^4 = 80 GB
- Cambios cualitativos
  - 100 procesadores por chip en el 2010
  - Lógica reconfigurable
  - Nano procesadores de 100 Kb
  - RAM on-chip
  - Ancho de banda en redes: 50 GB/sec
  - Latencia trasatlántica 100 ms $\rightarrow$ 25 ms

Algunos retos ingeniería sist. distr.

Construir sistemas:

- Correctos: que cumplan sus especificaciones
- Escalables: al aumento de entrada y al aumento de nodos
- Evolutivos
- De alto rendimiento: máximo uso de recursos al mínimo costo
- Paralelismo/Distribución
- Componentes reusables
- Dignos de confianza (tolerantes a fallas – disponibles permanentemente)
- Facilidad de uso

Algunos retos programación distribuida

- Concurrencia
  - Programar chips de 100 procesadores
  - Programar redes de estaciones de trabajo como una estacón centralizada y un computador paralelo
- Máximo uso de procesos con mínimos recursos
  - Equilibrio de carga
  - Mínimo uso de ancho de banda red
- Computación ubicua: aprovechar todo computador (plancha, lavadora, etc)

Conclusión

- Entienda y acepte la ley de Moore
- Amplísima gama de sistemas y aplicaciones que aún no se han construido
- Distribuidos y paralelos
- Hay que saber cómo construirlos

⇒ **Aprenda a construir sistemas distribuidos**

Ámbito de los SD

- **Teoría fundamental**
  - Modelos de cálculo distribuido
  - Algoritmos
  - Correctitud
- **Confiabilidad**
- **Privacidad y seguridad**
- **Metodologías de diseño y herramientas (ingeniería del software distribuido)**
  - Especificación
  - Diseño
  - Implementación

Ámbito de los SD (cont.)

- **Distribución y compartimiento**
  - Replicación
  - Descomposición de recursos
  - Distribución de carga
  - Elección de recursos
- **Ambiente del usuario**
- **Sistemas operativos distribuidos**
- **Bases de datos distribuidas**
- **Redes**

Lo que trataremos en éste curso

- Fundaciones teóricas
  - Estado global y orden de eventos
  - Relojes físicos y lógicos
- Sistema operativos distribuidos
  - Protocolos de comunicación (IPC)
  - Exclusión mutua
  - Elección
  - Auto estabilización
  - Deadlocks
  - Arquitecturas de sistemas operativos distribuidos
  - Algoritmos que implantan memoria compartida distribuida
- Metodologías de diseño y herramientas
  - Diseño y estructuración de sistemas
  - RPC
  - Objetos distribuidos
  - Memoria compartida distribuida
  - Especificación y verificación automática de sistemas concurrentes

Objetivos del curso

- Muy modestos
- Entender la terminología
- Identificar oportunidades de distribución
- Comprender los fundamentos teóricos
- Conocer los mecanismos básicos para construir SD
  - Métodos de estructuración
    - Cliente/servidor
    - RPC
    - Sistemas distribuidos orientados a Objetos
    - Modelo memoria compartida
- ¿Ser capaz de construir un SD? (depende de Ud.)

Programa tentativo del curso

1 Caracterización de los SD
2 Diseño de sistemas distribuidos
3 Redes e inter-redes
4 IPC
5 RPC
6 SD orientados a objetos (SDO)
7 Sistemas operativos distribuidos
8 Tiempo y coordinación
9 Memoria compartida distribuida
10 Especificación y verificación automática de sistemas concurrentes

Conocimientos de base

Para aprehender éste curso, Ud. requiere conocer

- **Nociones de algoritmos y estructuras de datos**
- **Nociones de redes**
- **Sistemas operativos**
- **Programación de sistemas**
  - Algo de programación paralela y distribuida
  - Algo de programación OO
  - C++
  - Ingeniería básica del software

Bibliografía

- **Distributed Systems. Coulouris, Dollimore, Kindberg. ISBN 0-201-62433-8 (este es nuestro libro texto)**
- Distributed Operating Systems. Tanenbaum. ISBN 0-13-143934-0
- Distributed Operating Systems. Sinha. ISBN 0-7803-1119-1
- Distributed Systems. Mullender. ISBN 0-201-62427-3
- Distributed Operating Systems & Algorithms. Chow - Johnson. ISBN 0-201-49838-3
- Distributed System Design. Wu. ISBN 0-8493-3178-1
- Distributed Operating Systems. Galli. ISBN 0-13-079843-6

Más bibliografía

- **Algoritmos distribuidos:**
  - Libros de Raynal:
  - Andrews ISBN 0-8053-0086-4
  - Lynch. ISBN 1-55860-348-4
  - Tel. ISBN 0-521-47069-2
  - Barbosa. ISBN 0-262-02412-8
- Design & validation of Computer Protocols Holzmann ISBN 0-13-539925-4 (está en web)
- Computer Networks
  - Peterson - Davie ISBN 1-55860-368-9
  - Tanenbaum ISBN 0-13-349945-6
- Transacciones
  - Gray - Reuter ISBN 1-55860-190-2 (Biblia)
  - Lynch et al ISBN 1-55860-104-X (teórico)

Actas de congresos

- ACM Symposium on Operating Systems Principles (SOSP)
- ACM Symposium on Principles of Distributed Computing
- ACM on Parallel Algorithms and Architectures
- IEEE Conference Distributed Computing Systems
- USENIX Symposium Operating Systems Design & Implementation (OSDI)
- ACM Architectural Support for Programing Languages & Operating Systems (ASPLOS)
- ACM SigComm conference

Publicaciones periódicas

- Distributed Computing
- Distributed Systems Engineering
- ACM Transactions on Computer Systems
- IEEE Transactions on Parallel and Distributed Systems
- Journal of Parallel and Distributed Computing
- ACM Transactions of Database Systems
- ACM Transactions of Programming Languages & Systems
- ACM SIGOPS Operating Systems Review

Metodología de trabajo

- Página web: ***www.cemisid.ing.ula.ve/~lrleon/sd/sd.htm***
  - sección ***"noticias"*** presenta información importante de última hora
  - Temas de seminarios y artículos a leer serán colocados en página web
  - sección ***"lecturas, programa y transparencias":***
    - Fecha, tópico, referencias y transparencias
    - **Ud. debe leer las referencias antes de clase**
    - Transparencias colocadas a lo sumo 7 días después de clase
- Clases magistrales
- Ejercicios libro texto asignados jueves, discutidos martes
- Lista email del curso: ***sd@cemisid.ing.ula.ve***
  - Para inscribirse, envíe email a *lrleon@cemisid.ing.ula.ve*
  - Divulgación de información; discusión

Evaluación

- Una revisión bibliográfica (15 %)
- Un seminario sobre un artículo reciente (15 %)
- Intervenciones, tareas y ejercicios en clase (15 %)
- Un desarrollo de software (40%)
- Dos exámenes/trabajo (20% cada uno)

- Temas de revisión bibliográfica y artículos de los seminarios serán colocados en página web
- Primero que llega, primero que tiene el tema
- Para el pregrado, la nota definitiva se multiplica por 1.5
- Para el postgrado, la nota definitiva es definitiva

Fechas importantes

- Revisión bibliográfica:
  - **jueves 9 de mayo**
  - Equipos de 2 personas
- Seminarios
  - A partir del **1ro de junio**
  - Horario fuera del curso a definir
- Proyecto de programación:
  - Equipos de 2 personas
  - No se haga ilusiones si Ud. no ha comenzado 30 días antes de la fecha de entrega
  - **Lunes 9 de septiembre**
  - En caso de retardo: Nota - $2^{(\#\text{ de días de retardo})}$

Tarea para el miércoles 13/3

Leer los siguientes artículos:

- Butler Lampson, *"Hints for Computer System Design"* Proceedings of the Ninth Symposium on Operating Systems Principles, Bretton Woods, NH, October 1983.
  También en ACM TOCS

- J. H. Saltzer, D. P. Reed, and D. D. Clark, *"End-To-End Arguments in System Design,"* ACM Transactions on Computer Systems, 2(4), November 1984

- Están en página web