

"Sistemas de Software Distribuido. Aplicaciones"

De Giusti A.¹, Pesado P.¹, Bertone R.¹, Boracchia M.¹, Thomas P.¹, Madoz C.¹

*LIDI - Laboratorio de Investigación y Desarrollo en Informática².
Facultad de Informática. Universidad Nacional de La Plata.*

Objetivo

Realizar investigación y desarrollo en temas relacionados con Sistemas de Software Distribuido, particularmente los relacionados con los aspectos de Ingeniería de Software que se orientan al desarrollo e implementación de proyectos de sistemas distribuidos, incluyendo especialmente los temas de métricas y calidad de software.

Estas tareas abarcan aspectos propios de la arquitectura, comunicaciones, estructuración de bases de datos, así como especificación, validación y desarrollo de aplicaciones de software en ambientes distribuidos. Asimismo se considera especialmente la investigación de la Ingeniería de Software de los sistemas distribuidos de tiempo real, considerando las extensiones necesarias en las metodologías y herramientas clásicas con el objetivo de considerar las restricciones de tiempo.

Introducción

Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan los recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada aunque esta pueda estar implementada por varias máquinas en distintas ubicaciones.

El desarrollo de sistemas distribuidos es una necesidad a partir de la utilización de redes de computadoras y de computadores personales de alta performance.

Las ventajas del procesamiento distribuido son:

- Mejora de la disponibilidad: la operación es factible en una configuración reducida cuando algunos nodos están temporalmente no disponibles. No hay un único punto de falla.
- Configuración más flexible: una aplicación puede configurarse de distintas maneras, seleccionando el número apropiado de nodos para una instancia dada.
- Control y administración más localizada: un subsistema distribuido, ejecutando en su propio nodo, puede diseñarse para ser autónomo, de modo que puede ejecutar en relativa independencia de otros subsistemas en otros nodos.
- Expansión incremental del sistema: si existe sobrecarga, el sistema puede expandirse agregando más nodos.
- Costo reducido: con frecuencia una solución distribuida es más barata que una centralizada.
- Balance de carga: en algunas aplicaciones la carga total del sistema puede ser compartida entre varios nodos.
- Mejora en el tiempo de respuesta: los usuarios locales en nodos locales pueden obtener respuestas más rápidas a sus requerimientos.

¹ Docentes-Investigadores LIDI – Facultad de Informática UNLP - E-Mail: ppesado@lidi.info.unlp.edu.ar

² LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.
TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

En particular un sistema distribuido de tiempo real debe interactuar con el mundo real, en puntos físicamente distantes y no necesariamente fijos, en períodos de tiempo que vienen determinados por el contexto o las restricciones de la especificación (en muchos casos a partir de una activación asincrónica).

La evolución tecnológica en el tratamiento de señales (locales o remotas) y en los sistemas de comunicaciones ha impulsado enormemente esta área temática, sobre todo en los aspectos de ingeniería de software tales como planificación, desarrollo y verificación de software para Sistemas Distribuidos. Es de fundamental importancia contar con herramientas CASE que asistan a lo largo de todo el ciclo de vida del proyecto.

La necesidad de supervisar el proceso de desarrollo del software torna imprescindible la utilización de métricas. La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua. También se puede utilizar en el proyecto para ayudar en la estimación, el control de calidad, la evaluación de productividad y el control de proyectos. Finalmente el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de productos de trabajos técnicos y para ayudar en la toma de decisiones tácticas a medida que el proyecto evoluciona. En muchos casos es necesario adaptar o ampliar las métricas existentes para poder utilizarlas en sistemas distribuidos, en particular de tiempo real.

Los conceptos de calidad que deben estar presentes en los procesos de desarrollo de software han llevado a crear modelos standarizados que permiten encuadrar a las organizaciones y a los proyectos. En particular existen dos normas ampliamente aceptadas que surgen de dos comunidades diferentes (CMM e ISO 900X).

La mejora continua en los procesos y productos de software es lo que se conoce como calidad total, que en gran parte es conseguida a través de la aplicación y evaluación de las métricas, y la realimentación del proceso productivo con estos resultados.

Desde el punto de vista de las aplicaciones se considerarán en particular las Bases de Datos Distribuidas y aspectos de Tecnología de Software aplicada a educación.

El modelo distribuido de datos hace posible la integración de BD heterogéneas proveyendo una independencia global del administrador de bases de datos (DBMS) respecto del esquema conceptual. Además, es posible implementar una integración tal que reúna varios modelos de datos, representando cada uno de ellos características propias de organizaciones diferentes, asociadas para un trabajo conjunto.

Este modelo de distribución normalmente tiene un soporte multiprocesador tipo MIMD sobre diferentes esquemas de comunicaciones. En tal arquitectura se reproducen los problemas de distribución óptima de datos y procesos, de migración de datos y procesos y de tolerancia a fallas.

En particular interesa estudiar el comportamiento (en tiempo de respuesta y confiabilidad) de grandes bases de datos sobre arquitecturas distribuidas de redes LAN y WAN. Esto significa problemas tales como: Tasa de pérdida de datos; Tiempo máximo necesario para recuperación de información; Complejidad y eficiencia de los algoritmos de recuperación; Tiempo de utilización de recursos del sistema; Incidencia del porcentaje de replicación en el tiempo de respuesta.

En este marco, una línea de investigación actual consiste en desarrollar ambientes de experimentación sobre redes de estaciones de trabajo (NOW) para estudiar grandes Bases de Datos distribuidas que deban ser accedidas en tiempo real (por ello puede requerirse tener un porcentaje de la BD on-line y otro off-line).

La Tecnología de Software aplicada en Educación, es una disciplina en pleno crecimiento, no sólo por el interés y múltiples aplicaciones que ha suscitado en las escuelas, universidades y centros de entrenamiento empresarial, sino porque ha permitido incorporar el tratamiento de la información al proceso educativo.

Esta incorporación de la informática al quehacer educativo involucra a las diferentes modalidades del proceso de enseñanza / aprendizaje - entrenamiento, esto es a distancia, semi presencial o presencial. El desarrollo de ambientes y herramientas con este objetivo (en particular los dedicados WEB based learning) es un tema muy importante en la Ingeniería de Software de Sistemas Distribuidos.

Es por esta razón que los recursos tecnológicos actuales se han constituido en temas de investigación y aplicación fundamentales en los desarrollos multidisciplinarios de Ciencias de la Educación y Ciencia Informática. Por lo tanto, la evolución del mundo educativo en cuanto a métodos y recursos tecnológicos requiere que los usuarios de dicha tecnología (docentes y alumnos) renueven y actualicen sus conocimientos de productos de software educativos y estén preparados para administrar esos nuevos productos. En este contexto la definición y utilización de métricas precisas para la evaluación de calidad de productos de software educativo (en particular en Educación no presencial) es una línea de I/D presente en el proyecto.

Líneas de Estudio/Investigación y Desarrollo:

- Procesamiento distribuido. Arquitectura, comunicaciones y software. Lenguajes y ambientes para procesamiento distribuido.
- Ingeniería de Software de sistemas distribuidos. Extensiones para el tratamiento de sistemas distribuidos de tiempo real.
- Metodologías de especificación, validación y desarrollo de sistemas de software distribuido. Herramientas CASE orientadas a sistemas de software distribuido.
- Métricas para la evaluación de software. Métricas para la estimación y seguimiento de proyectos. Normas de calidad en software. ISO 900X y CMM.
- Herramientas de integración y mantenimiento de proyectos distribuidos.
- Reingeniería de sistemas complejos que migran por down-sizing a esquemas cliente-servidor distribuidos.
- Bases de Datos distribuidas. Replicación. Consistencia. Migración de datos. Tolerancia a fallas.
- Tecnología de software aplicada en Educación. Educación a distancia. Educación basada en WEB. Métricas de calidad para software educativo.

Bibliografía

ACM eLearn , ACM Journal of Educational Resources in Computing, ACM Transactions on Computer Systems ,ACM Transactions on Computer-Human Interaction, ACM Transactions on Database Systems ,ACM Transactions on Information Systems, ACM Transactions on Software Engineering and Methodology.

Andrews, "concurrent Programming", Benjamin/Cummings, 1991.

Arthur L. "Improving software quality" , Wiley 1993

Beizer B. "Software testing techniques" Internation Thompson Computer Press 1990

Bell, David; Grimson, Jane, "Distributed Database Systems",. Addison Wesley. 1992

Bobak, A, "Distributed and Multi-Databse Systems", Artech House, 1996

Buretta, M "Data Replication", John Wiley and Sons, 1997

Burleson, D, "Managing Distributed Databases, Building Bridges between Database Island", John Willey 1995.

Communications of the ACM

Coulouris G., Dollimore J., Kindberg T. "Distributed Systems Concepts and Design", Addison Wesley 1994

Date, C.J., "Introducción a los sistemas de Bases deDatos". Addison Wesley 1994.

De Marco."Controlling Software Projects". Yourdon,1998

Frakes, W, Baeza Yates, R "Information Retrieval. Data Structures & Algorithms", Prentice Hall 1992

Ellison K. "Developing real time embedded software" . Wiley 1994.

Hatley D., Pirbhai I., "Strategies for Real-Time System Specification", Dorset House, 1988.

Humphrey W.."Managing the software process". Addison Wesley, 1989.

Huns M, Singh, M Readings in Agents", Morgan Kaufmann Publishers, 1997.

IEEE Computer, IEEE Concurrency, IEEE Realiability and Manteniability Symposium, IEEE Sigact News, IEEE Software Engineering, IEEE Transactions on Parallel and Distributed Processing

Jajodia S., Kerschberg L. " Advanced transaction models and architectures" Kluwer Academic Pub. 1997

Larson, J., "Database Directions. From relational to distributed, multimedia, and OO database Systems". Prentice Hall. 1995

Leben, M, "Client Server Databases Enterprise Computing", Prentice Hall, 1997.

Loucopoulos, P, Vassilio, K, "System Requirements Engineering" McGraw-Hill Book Company

Marques, P. "Metodología para la elaboración del software educativo en Software Educativo. Guía de uso y Metodología de diseño" , Barcelona 1995.

Moore J. "Software engineering standards" IEEE 2000

Özsu, T, Valduriez, P,"Principels of distributed database systems", Prentice Hall, 1999

Pfleeger S. "Ingeniería de Software. Teoría y Práctica". Addison Wesley 2000.

Pressman, R;"Ingeniería de Software, Un enfoque práctico." Cuarta edición. Mc Graw Hill. 1998

Robinson, W, Volkov, S, "Conflict-Oriented Requirements Restructuring" Departament of Computer Information Systems. Georgia State University.Atlanta. USA.

Seepold, R, Martinez Madrid, N, "Virtual Components Design and Reuse", Kluwer Academic Publishers - 2000.

Shumate K., "Software specification and design for real-time systems", Wiley 1992.

Silbershatz; F., "Fundamentos de las Bases de Datos".. Mc Graw Hill. 1998.

Wolf, W, "Computers As Components: Principles of Embedded Computing System Design", Morgan Kaufmann - 2000.