

Código-Materia: 09661 - Sistemas Distribuidos

Requisitos: 09732 - Redes de Computadores 1, 09728 - Programación en red

Programa-Semestre: Ingeniería Telemática - Semestre 9

Periodo académico: 2019-2

Intensidad Horaria: 4 Horas / Semana

Créditos: 3

Programa: Ingeniería de Sistemas	
Resultados de Aprendizaje (Outcomes)	<i>Curso Electivo</i>
Fuente de Valoración (Assessment Source)	<i>Curso Electivo</i>
Programa: Ingeniería Telemática	
Resultados de Aprendizaje (Outcomes)	Outcome K: Level Apply (T)
Fuente de Valoración (Assessment Source)	Outcome K-PI1 a PI2

Introducción

El curso de sistemas distribuidos está orientado a formar estudiantes con la capacidad de diseñar, implementar y desplegar arquitecturas distribuidas que den respuesta a las necesidades de procesamiento en los campos científico y tecnológico a niveles tanto académico como empresarial.

Un **sistema distribuido** es una colección de sistemas finales separados físicamente y conectados entre sí por una red de comunicaciones. En dicho sistema distribuido confluyen un conjunto de tecnologías que hacen posible cumplir con características para el manejo de altos niveles de concurrencia, tolerancia a fallos, escalabilidad, transparencia y seguridad.

Las aplicaciones distribuidas pueden ser clasificadas en dos grandes grupos. El primero lo constituyen las aplicaciones que requieren la ejecución de tareas de alto costo computacional por lotes sin intervención del usuario o agentes externos. El segundo está conformado por las aplicaciones que requieren de un alto nivel de conectividad con agentes externos que permita el intercambio masivo de información sin pérdidas (paradigma microservicios). Los tipos de aplicaciones descritos anteriormente, sus fundamentos, despliegue y monitoreo serán el objeto de estudio del curso sistemas distribuidos.

Objetivos

Objetivo General

- Diseñar y desplegar sistemas distribuidos considerando los atributos de calidad del servicio*.
*(tolerancia a fallas, niveles de concurrencia, alta disponibilidad, transparencia, seguridad y escalabilidad)

Objetivos Terminales

- Diseñar o rediseñar la arquitectura de un sistema distribuido que implemente un servicio, considerando las implicaciones técnicas asociadas con su escalabilidad, tolerancia a fallos y concurrencia, y las mejoras en el desempeño a través de la asignación o reasignación de los recursos y tareas.
- Desplegar el sistema distribuido que implementa un servicio, teniendo en cuenta las estrategias de administración de sus recursos consideradas en el diseño.
- Gestionar el servicio distribuido, aprovisionando automáticamente sus recursos.
- Efectuar un análisis de causa raíz de un error que resulta de la integración de componentes de un sistema distribuido, empleando la documentación oficial de las herramientas y logs de los sistemas operativos

Contenido

Este es el contenido propuesto para el desarrollo del curso:

Unidad 1. Introducción a los Sistemas Distribuidos

Temas: Introducción a Grid y Cloud Computing

- Explicar las necesidades de emplear una arquitectura distribuida.
- Diferenciar los conceptos de Grid y Cloud Computing
- Proponer el empleo de un nivel de servicio en Cloud Computing (SaaS, PaaS, IaaS) dependiendo la naturaleza del problema
- Identificar los principales vendedores de servicios en la nube y sus modelos de negocio

Unidad 2. Diseño de Sistemas Distribuidos

Temas: Modelo del Sistema, Modelo de Arquitectura, Modelo Fundamental

- Reconocer los tipos de entidades que se comunican en un sistema distribuido
- Comprender los paradigmas de comunicación que se emplean en un sistema distribuido
- Diferenciar los tipos de arquitectura que se presentan en un sistema distribuido
- Clasificar los tipos de fallas que surgen en un sistema distribuido a nivel de los procesos y canales de comunicación

Unidad 3. Comunicación inter-procesos

Temas: Protocolos por capas, RPC, Invocación Remota de Objetos, Comunicación por paso de mensajes, Comunicación orientada por flujos

- Plantear las ventajas y desventajas de emplear un paradigma de comunicación en un sistema distribuido para solucionar un problema específico

- Implementar mecanismos de comunicación de paso de mensajes empleando frameworks de mensajería
- Implementar aplicaciones que consuman o provean servicios web

Unidad 4. Procesos

Temas: Threads, Clientes, Servidores, Migración de código, Máquinas virtuales

- Comprender la importancia de la concurrencia en la implementación de servidores web y servidores de aplicación
- Identificar las ventajas y desventajas entre usar contenedores tipo LXC ó hipervisores de tipo 1 y 2.

Unidad 5. Nombramiento

Temas: Entidades de nombramiento, Localización de entidades móviles, Removiendo entidades no referenciadas

- Explicar la importancia del nombramiento y el descubrimiento de recursos en un sistema distribuido.
- Entender los problemas relevantes en el diseño e implementación de un servicio de nombramiento, entre ellos: el espacio de nombres, el mecanismo de resolución, la división y replicación de la información entre servidores y la operación de la memoria cache

Unidad 6. Sincronización

Temas: Sincronización de reloj, Relojes lógicos, Estado global, Algoritmos de elección, Exclusión mutua, Transacciones distribuidas

- Proponer mecanismos de sincronización para un sistema distribuido, teniendo en cuenta las limitaciones de emplear relojes locales
- Comprender las bases que soportan la concurrencia en aplicaciones distribuidas

Unidad 7. Consistencia y replicación

Temas: Modelos de consistencia centrada en datos, Modelos de consistencia centrada en clientes, Protocolos de distribución, Protocolos de consistencia

- Comprender la importancia de las réplicas de los sistemas distribuidos como un mecanismo para alcanzar disponibilidad en un sistema distribuido
- Identificar los problemas que se generan al emplear replicas en un sistema distribuido
- Proponer mecanismos para mantener la consistencia de los datos en un entorno distribuido

Unidad 8. Tolerancia a fallos

Temas: Procesos, Comunicación confiable cliente/servidor, Comunicación confiable en grupo, Commit distribuido, Recuperación

- Comprender por qué son importantes en un sistema distribuido los mecanismos de tolerancia a fallos
- Proponer mecanismos para garantizar la tolerancia a fallos en un sistema distribuido
- Diseñar e implementar sistemas que puedan continuar su operación de forma transparente ante la falla de uno de sus componentes garantizando los principios de confiabilidad y disponibilidad

Unidad 9. Sistemas de ficheros distribuidos

Temas: Sistemas de ficheros distribuidos, Sistema de archivos distribuido NFS, Sistema de archivos distribuido AFS

- Identificar y explicar los componentes de la arquitectura de un sistema de archivos distribuido
- Comprender el funcionamiento de los sistemas de archivos distribuidos en las herramientas que proveen servicios de computación distribuida hoy en día

Talleres

Este es el listado de los talleres propuestos para el desarrollo del curso:

1. Microservicios
2. Infraestructura como código (Vagrant, Chef, Puppet, Ansible, Salt Stack)
3. Mensajería (RabbitMQ, Kafka)
4. Monitoreo (Stack ELK, Prometheus+Kibana)
5. Descubrimiento de Servicio (Consul, Etcd)
6. Contenedores virtuales (docker, rocket)
7. Orquestación de contenedores (docker swarm, k8s)
8. Contenedores HPC (MPI, Spark) - **Opcional**
9. Serverless (OpenWhisk) - **Opcional**

Formación Previa

El curso de sistemas distribuidos requiere la integración de conocimiento de diversos campos de ingeniería. Los conocimientos mínimos para la comprensión del curso comprenden:

- Redes y Comunicaciones
- Sistema Operativos
- Programación Orientada a Objetos
- Programación de Aplicaciones Cliente-Servidor

Metodología

El curso se desarrollará de forma teórico-práctica. En las clases teóricas se expondrán los fundamentos de los sistemas distribuidos y se discutirá sobre artículos relacionados con las tendencias en computación *Grid/Cloud*. En las clases prácticas se abordarán en un comienzo las herramientas básicas que dan soporte a la comunicación de sistemas finales en un entorno distribuido, hasta las herramientas que permiten el despliegue de aplicaciones.

Los parciales serán teóricos-prácticos. El proyecto final consistirá en la solución de un problema que se apoye sobre una infraestructura de cómputo distribuida, del cual se deberá entregar un informe empleando el formato IEEE con al menos 5 referencias de revistas electrónicas como IEEE, ACM o Springer que den soporte al trabajo realizado.

Evaluación

La forma de evaluación del curso es la siguiente:

- 3 Parciales

- Talleres
- 1 Proyecto final

El cálculo de la nota final se realizará teniendo en cuenta el siguiente algoritmo:

$\text{Nota promedio} = (\text{Promedio Parciales} * 60\%) + (\text{Promedio Talleres} * 40\%) +$

Si $\text{Nota promedio} \geq 3.0$ entonces

$\text{Nota definitiva} = (\text{Nota promedio} * 60\%) + (\text{Nota proyecto final} * 40\%)$

Sino

$\text{Nota definitiva} = \text{Nota promedio}$

Referencias

[1] Andrew S. Tanenbaum and Maarten Van Steen. Distributed Systems: Principles and Paradigms (2nd Edition). Prentice Hall PTR, Upper Saddle River, NJ, USA, 2006

[2] Chuck Lam. Hadoop in Action. Manning Publications, 2010

[3] George F. Coulouris and Jean Dollimore. Distributed Systems: Concepts and Design (5th Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2007.

[4] Kevin Jackson. OpenStack Cloud Computing Cookbook. Open Source. 2012

[5] Robert Daigneau. Service Design Patterns Fundamental Design Solutions for SOAP WSDL and RESTful Web Services. Addison Wesley Professional, 2011

[6] <http://www.github.com/ICESI>