Aspectos de Diseño de los Sistemas Distribuidos

Diego Alberto Rincón Yáñez MSc
drincony@poligran.edu.co
INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
Facultad de Ingeniería

Agenda

Conceptos Generales

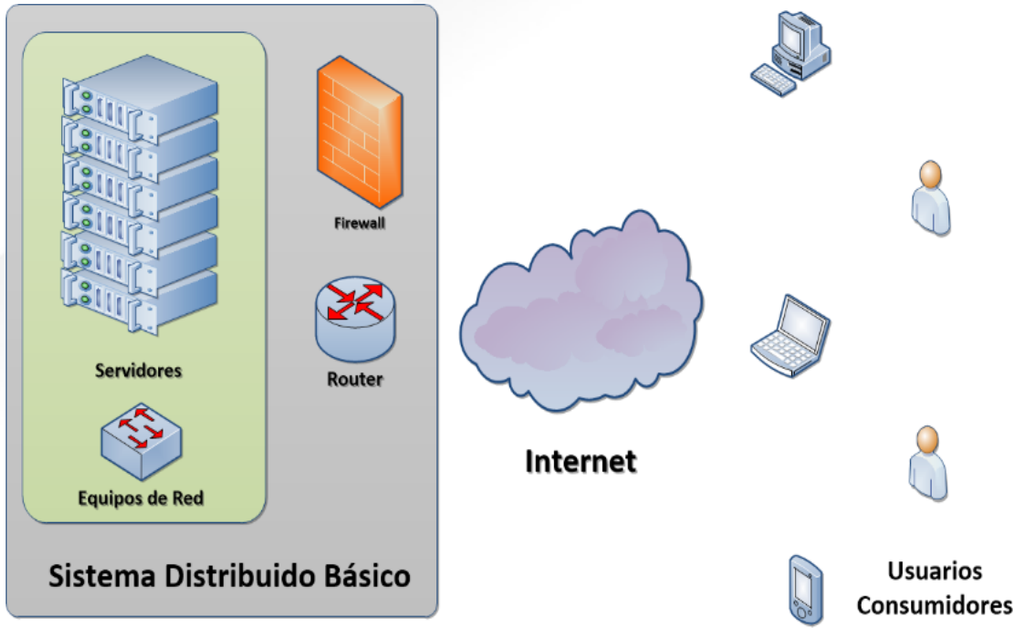
- ¿Que es?
- Composición
- Ventajas/Desventajas
- Conceptos Hardware/Software

Aspectos de Diseño

- Flexibilidad
- Confiabilidad
- Desempeño
- Escalabilidad
- Transparencia

¿Qué es un Sistema Distribuido?

Conjunto de máquinas de cómputo conectadas a una **red de computadores**, con el **fin de prestar un servicio** a un usuario final

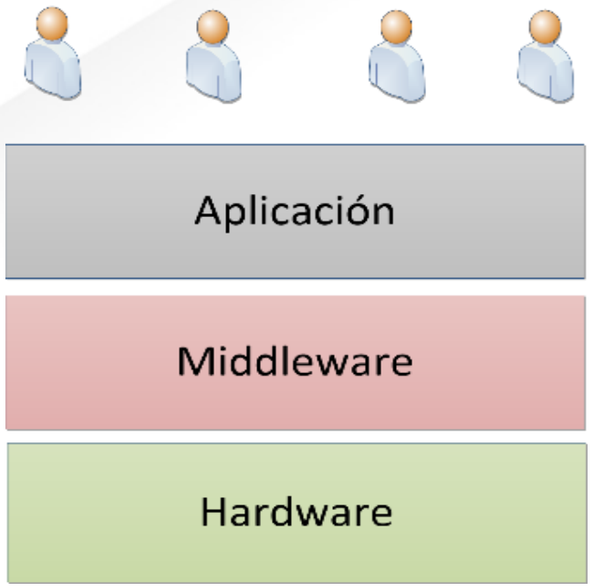


Composición

Equipos de Cómputo y hardware adicional

Servidores (Procesamiento y Almacenamiento)

Equipos de Red (Switches, routers, firewalls)



Composición

Aplicación que prestará el servicio

- Correo Electrónico
- CRM - Customer Relationship Management
- ERP - Enterprise Resource Planning
- Portales Corporativos
- Pasarelas de Pagos
- Tiendas Virtuales

Composición

Middleware y software transversal

- Sistemas Operativos de tipo servidor
- Middleware
  - Manejador de Transacciones
  - Manejador de Sesiones
  - Bus de Servicios
  - Manejadores de conexiones a datos
- Servicios Transversales
  - Autenticación
  - Seguridad
  - Relojes

Ejemplos

Buscadores: Google, Yahoo, bing, Duck Duck Go, etc

Redes Sociales: Facebook, twitter, instagram, etc.

Comercio: amazon, mercadolibre, aliexpress, etc.

Finanzas: Bolsas de Valores, Bancos, Bitcoin, etc.

Educación: Coursera, EdX, MIT, Stanford, Politecnico Grancolombiano, etc.

Ciencias: LHC, SETI@Home, ATLAS; LIGO, Hubble.

Entretenimiento: Netflix, Steam, Youtube, PlayStation Network, etc.

Comparación

¿***Sistemas distribuidos con respecto a los centralizados/servidores***

Economía: Relación precio/rendimiento.

Desempeño: Se pueden unir gran cantidad de CPU y tener una capacidad de computo imposible de obtener en un mainframe.

Distribución inherente: Uso de máquinas separadas en distancia.

Comparación

¿Sistemas distribuidos con respecto a los centralizados/servidores

Confiabilidad: Si una máquina se descompone, el sistema puede sobrevivir como un todo.

Disponibilidad: Distribución de la carga.

Crecimiento por incrementos: Se puede añadir poder de cómputo en pequeños incrementos.

Comparación

¿***Sistemas distribuidos con respecto a los centralizados/servidores***

Complejidad:

Nivel de Abstracción

Los algoritmos son totalmente distintos a la forma natural de proponer soluciones.

Redes: Dependencia directa de la comunicación.

Seguridad: requerimientos de diseño versus sistemas seguros.

Comparación

Aunque todos los sistemas distribuidos constan de varias CPU, existen diversas formas de organizar el hardware.

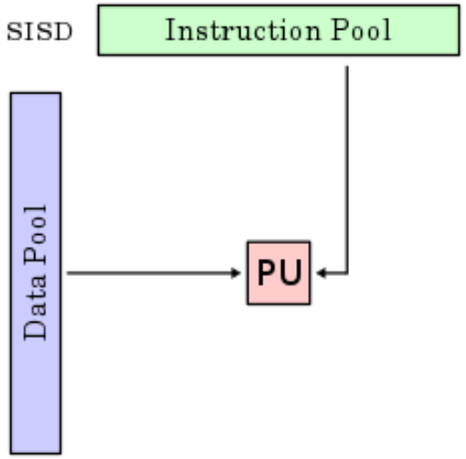
Taxonomía más citada Flynn(1972).

Basado en los números de flujos de instrucciones y de datos.

Conceptos de Hardware

SISD(Single Instruction, Single Data): Un flujo de instrucciones y un flujo de datos.

Ejemplo: Todas los computadores tradicionales de un procesador

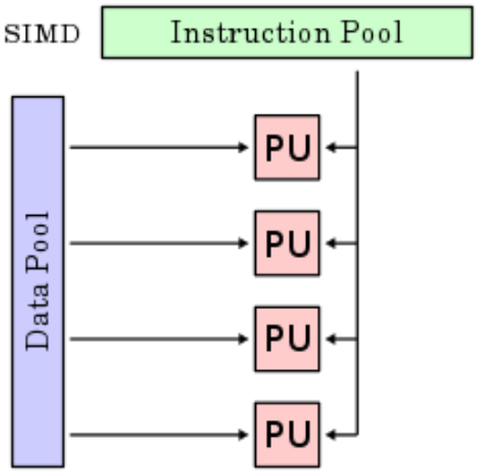


Conceptos de Hardware

SIMD(Single Instruction, Multiple Data): Un flujo de instrucciones y varios flujos de datos.

Ejemplo: Sumando todos los elementos de 64 vectores independientes.

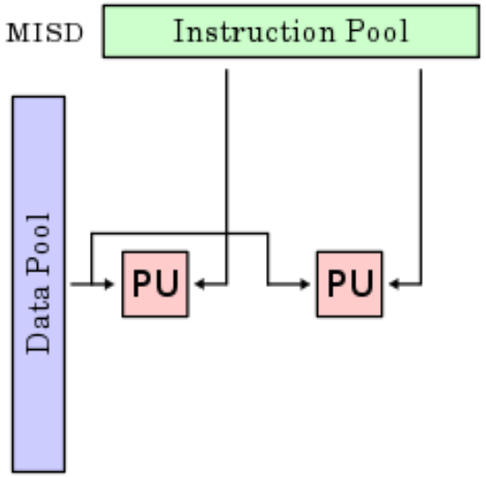
Útil: Cómputos que repiten los mismos cálculos en varios conjuntos de datos



Conceptos de Hardware

MISD(Multiple Instruction, Simple Data): Múltiples flujos de instrucciones y un flujo de datos.

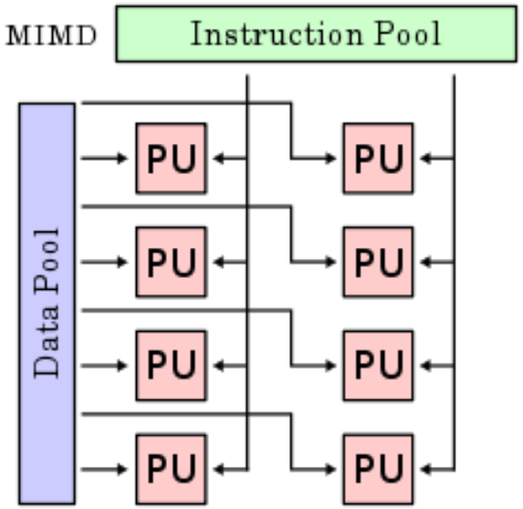
Ejemplo: Ninguna de los computadores conocidas se ajusta a este modelo.



Conceptos de Hardware

MIMD (Multiple Instruction, Multiple Data): Varios flujos de instrucciones y varios flujos de datos.

Computadores independientes interconectadas, todos los sistemas distribuidos hacen parte de esta categoría.



Conceptos de Software

La forma de pensar de los usuarios de un sistema queda determinada por el software del sistema operativo.

Tipos:

Débilmente acoplados: Maquinas y usuarios sean independientes entre sí en lo fundamental, pero interactúen en cierto grado cuando sea necesario.

Fuertemente acoplados: Multiprocesador dedicado a la ejecución del ajedrez en paralelo. Cada CPU evalúa un tablero de ajedrez y reporta resultados.

Conceptos de Software

Sistemas operativos de redes

Software débilmente acoplado y hardware débilmente acoplado.

Red de estaciones de trabajo interconectados por una LAN (rlogin, rcp).

Cada máquina tiene gran autonomía y existen pocos requisitos a lo largo del sistema (sistema operativo de red)

Conceptos de Software

Sistemas realmente distribuidos

Software fuertemente acoplado y hardware débilmente acoplado.

Crear la ilusión que toda la red de computadores es un sistema de tiempo compartido.

Características:

- Mecanismo de comunicación global entre procesos.
- Administración la misma en todas partes.
- Sistema de archivos debe tener la misma apariencia en todas partes.

Conceptos de Software

Sistemas de multiprocesador con tiempo compartido

Software y hardware fuertemente acoplados.

Ejemplos: Multiprocesadores que operan con un sistema de tiempo compartido (linux)

Característica:

Existencia de una cola de ejecución (Estructura de datos contenida en la memoria compartida)

Aspectos de Diseño de los Sistemas Distribuidos

Generalidades

Aspectos de Diseño

Lineamientos

- Capa Física
- Capa Lógica

Primer paso

- ¿De cuántos componentes físicos y lógicos está compuesto el sistema?
- ¿Cómo están los componentes físicos interactuando entre sí?
- ¿Cuál es el nivel de servicio de mi aplicación?
- ¿Qué tan preparado estoy para los desastres?
- ¿Cuantos usuarios soporta mi sistema?

Generalidades

Aspectos de Diseño

- **Flexibilidad**
  - Heterogeneidad
  - Apertura
  - Capacidad de evolución
- **Confiabilidad**
  - Disponibilidad
  - Tolerancia a Fallos
- **Desempeño**
  - Modelos de programación distribuida
  - Ley de Moore/Amdahl
- **Escalabilidad**
  - Escalabilidad Vertical
  - Escalabilidad Horizontal
- **Transparencia**
  - Relacionados Aspectos de Diseño
    - Concurrencia
    - Replicación
    - Fallos
    - Prestaciones
    - Migración
    - Escalado
  - Exclusivos
    - Localización
    - Acceso

Flexibilidad

Heterogeneidad: En el curso anterior de sistemas operativos se realizó el estudio de la arquitectura Von-Neuman(Godfrey & Hendry, 1993). Esta arquitectura nos ha permitido estandarizar lo que es un computador, y gracias a esto, existen muchas opciones de computadores en el mercado, ya sean de características grandes o pequeñas, especificaciones grandes o pequeñas

Apertura: Significa que el sistema no sea cerrado, que pueda ser capaz de intercambiar información y servicios con otros sistemas diferentes, como es el caso de varios proveedores de identidad como lo son Facebook y Google. Lo anterior sucede es debido al momento de una aplicación externa conectarse con estos sistemas, esa aplicación conoce ciertos protocolos orientados a (en este caso) la autenticación, tal como lo nombramos en el escenario 1 en el apartado de los protocolos.

Flexibilidad

Capacidad de evolución: Tal y como lo indica el ciclo de vida del desarrollo del software(Pressman, 2005), el sistema debe estar en constante evolución y crecimiento, el sistema distribuido deberá tener la capacidad de aceptar nuevas funciones o cambios de sus funciones a través del tiempo, esto permite que no se convierta en un sistema estático, sino que sea un ser vivo dentro de la organización, el cual está abierto a nuevos procesos e intercambios con nuevas funcionalidades y sistemas externos. Cabe aclarar que este tipo de evolución está orientada a la capacidad de soportar nuevas funcionalidades, la habilidad de crecer en cuanto a cantidad de usuarios se explorará la Escalabilidad.

Confiabilidad

Disponibilidad: tiempo de operación continuo sin fallos que puede tener un sistema, este tiempo está medido en porcentaje (%) el cual es va desde 0% que sería total indisponibilidad en el año a 100% total disponibilidad del servicio de forma ininterrumpida en el año

$$\%D = \frac{Tt - \sum_1^f Tf}{Tt}$$

%D : Porcentaje de Disponibilidad

Tt: Tiempo total del Servicio (generalmente 1 año – 31.536.000 segs)

f : Número total de fallos en el tiempo total del servicio

Tf: Tiempo individual en segundos de los fallos ocurridos

Confiabilidad

Disponibilidad

- **TIER I:** Ofrece el nivel más básico de protección y ninguna redundancia para la infraestructura de misión crítica de la organización.
- **TIER II:** Necesita cumplir con las medidas anteriores y adicionalmente capacidad redundante en los sistemas críticos y proveer aislamiento para dichos componentes
- **TIER III:** Cumplir con las medidas del TIER II y adicionalmente, todos los equipos tienen que tener redundancia a nivel eléctrico, debe tener mecanismos de generación de energía o fuente de energía alterna.
- **TIER IV**: Cumplir con las medidas del TIER III y adicionalmente, múltiples y distintos caminos eléctricos, de servicio y red, cambio automático para para momentos de falla a los sistemas de respaldo, enfriamiento continúo para todos los equipos en el sitio.

Nivel del Data Center	Porcentaje de Disponibilidad	Tiempo de Indisponibilidad
Tier I	99.671%	28.82 horas
Tier II	99.741%	22.68 horas
Tier III	99.982%	1.57 horas
Tier IV	99.995%	52.56 minutos

Confiabilidad

Tolerancia a Fallos

- **Configuración de Capacidad (N):** Es la configuración básica necesaria para la prestación del servicio, sin ningún componente adicional, es la capacidad mínima necesaria para mantener el servicio, en requerimiento de cualquier mantenimiento es necesario crear una interrupción de servicio. Generalmente se aplican en cuartos de datos artesanales o datacenters tipo TIER I.
- **Redundancia Aislada (N+1):** Esta es una de las configuraciones mínimas para la redundancia, este nivel requiere que exista otro componente, adicional al mínimo, en stand-by, el cual en caso de fallos el componente en stand-by entra a operar como servicio principal, bajando la cantidad de tiempo de indisponibilidad considerablemente, este tipo de configuración se conoce normalmente como esquema Activo-Pasivo. Generalmente se aplican en cuartos de datos artesanales o datacenters tipo TIER II.

Nombre de Redundancia	Ecuación de Redundancia
Configuración de Capacidad	N
Redundancia Aislada	N + 1
Redundancia en paralelo	N + 1
Redundancia Sistema * Sistema	2 N
Doble redundancia en paralelo	2 (N + 1)
Doble redundancia distribuida	2 (N + 1)

Confiabilidad

Tolerancia a Fallos

- **Redundancia en paralelo (N+1):** Esta configuración se conoce como esquema Activo-Activo, de modo de que ambos componentes se encuentran prestando el servicio de forma sincronizada, lo que significa que existe capacidad adicional prestando el servicio, al momento de fallo, toda la carga pasaría al componente que continua activo, garantizando que el servicio se prestaría a pesar del fallo. Generalmente se aplican en cuartos de datos artesanales o datacenters tipo TIER II y III.
- **Redundancia Sistema * Sistema (2N):** Este esquema está relacionado con la duplicación de los componentes necesarios para la prestación del servicio, así como los esquemas anteriores que agregaban recursos adicionales este los duplica, previniendo garantizando que en caso de fallas los usuarios podrán tener el servicio con toda capacidad, adicionalmente cabe aclarar que este esquema permite que los recursos aparte de estar redundantes, estén distribuidos geográficamente, lo que aumenta la probabilidad de continuidad en los niveles de servicio. Generalmente se aplican en cuartos de datos artesanales o datacenters tipo TIER IV mientras exista la posibilidad de tener dos proveedores eléctricos independientes y redundantes para cada unidad.
- **Doble redundancia en paralelo (2(N+1)):** Este esquema de servicio es de los más confiables existentes, los cuales combina los puntos tres y cuatro (3 y 4), garantizando el esquema de servicio mínimo, más el componente adicional del +1, teniendo doble redundancia de componentes primarios y adicionalmente manejando de forma paralela un esquema de servicio activo-activo.
- **Redundante distribuida (2(N+1)):** Para finalizar, este esquema tiene en cuenta los esquemas presentados en los puntos dos y cuatro (2 y 4), con el gin de conservar la capacidad de servicio adicional del n+1, pero con los componentes replicados en una ubicación geográfica totalmente diferente.

Confiabilidad

Tolerancia a Fallos

MTBF: Confiabilidad de un componente de software o de un sistema informático, esta métrica es el tiempo medio entre fallas o llamado en inglés Mean Time Between Failures. El MTBF normalmente está expresado en horas, que transcurren hasta que se produce un error en un componente y es preciso repararlo.

$$MTBF = \frac{Tt - \sum_1^f Tf}{f}$$

MTBF : Tiempo medio entre fallas (Men Time Between Failures)
Tt: Tiempo total del Servicio-generalmente 1 año – 31.536.000 segs
f : Número total de fallos en el tiempo total del servicio
Tf: Tiempo individual en segundos de los fallos ocurridos

Desempeño

El desempeño está relacionado con una rama de la computación, la cual se explorará más adelante de este curso (Computación de Alto Desempeño).

Diferentes métricas:

- Tiempo de respuesta
- Número de trabajos por hora (rendimiento)
- Uso del sistema

Tiempo de ejecución (Tp), sobrecarga paralelizada (To), speed-up (s), la eficiencia (E) y costo (PTp)

Desempeño

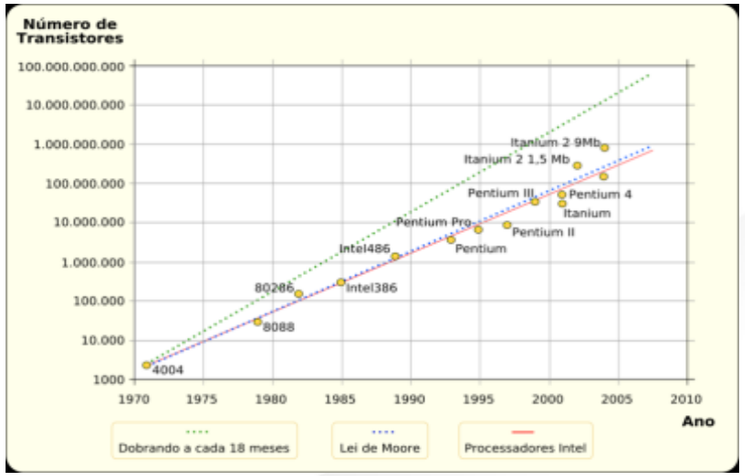
Modelos de programación distribuida

Comunicación de memoria compartida: Componentes concurrentes se comunicación alterando el contenido de memorias compartidas. Este estilo de programación concurrente generalmente requiere de la aplicación de algunas maneras de asegurado (e.g. mutexs, semáforos o monitores) para coordinar entre hilos (Ben-Ari, 2006).

Comunicación de Paso de Mensajes: Componentes concurrentes se comunican intercambiando mensajes que pueden ser cargados asíncronamente o que puede usarse un estilo “rendezvous” en cual la entidad que envía bloquea hasta que el mensaje es recibido (Quinn & J., 2004). Asincrónicamente el paso de mensajes puede ser de confianza o de desconfianza. Una amplia cantidad de teorías matemáticas para el entendimiento de y el análisis de sistemas de paso de mensajes están disponibles (Ben-Ari, 2006).

Desempeño

Ley de Amdahl y Ley de Moore



$$S = \frac{1}{(1-P) + \frac{P}{N}}$$

1: Unidad de Tiempo que toma el algoritmo
P: Parte modificada de Código (Paralelizada)
N: Cantidad de procesadores

Escalabilidad

Este aspecto de diseño hace referencia a la capacidad de un sistema de crecer según la cantidad usuarios concurrentes requiriendo sus servicios. Con el concepto de escalabilidad le permitimos al sistema crecer sin necesidad de:

- Afectar el servicio de los usuarios existentes.
- Incrementar la cantidad de usuarios del servicio.
- Garantizar las todas las funcionalidades del sistema.

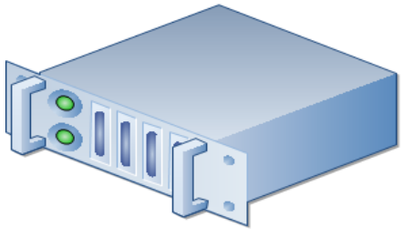
Generalmente se recomienda estudiar las ventajas y desventajas de cada una de las anteriores expuestas según

- Requerimientos de los Usuarios
- Tiempo de desarrollo del proyecto
- Infraestructura Disponible
- Esquema de servicio y operación
- Presupuesto

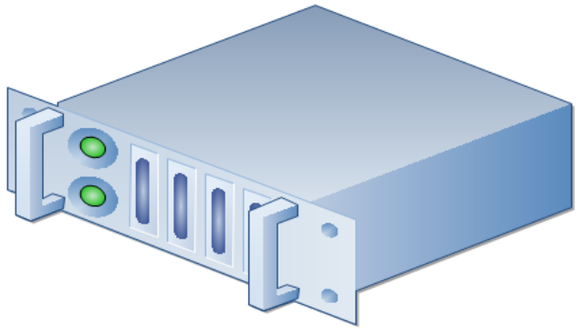
Escalabilidad

Escalabilidad Vertical

Este esquema de crecimiento está orientado al crecimiento de los componentes del sistema, estos componentes pueden ser el crecimiento de un servidor en capacidad, con el fin de que con las mismas unidades físicas se presten servicios a más usuarios.



Servidor de con Características Específicas

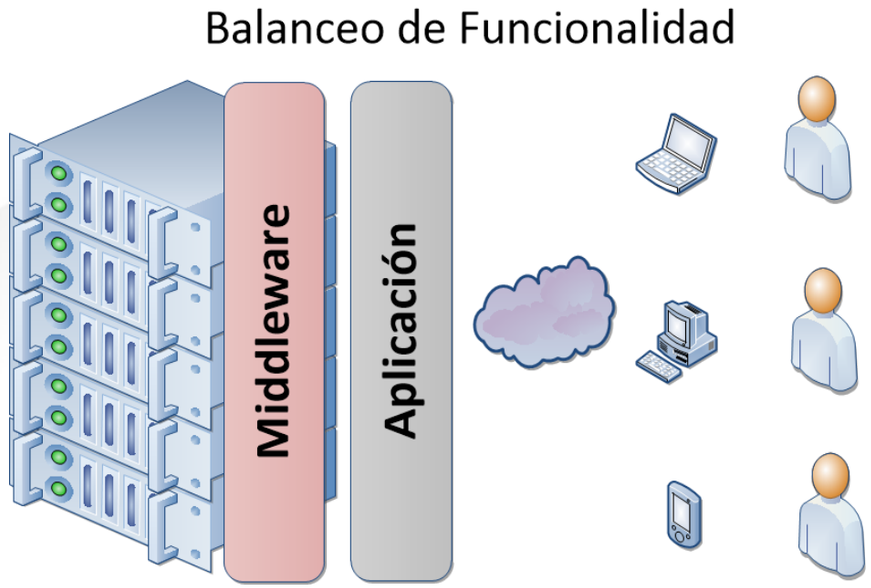


Servidor de con Características Aumentadas

Escalabilidad

Escalabilidad Horizontal

Este esquema también conocido como **balanceo de carga**, es necesario cuando los componentes de la aplicación están fuertemente acoplados, esto quiere decir que los módulos que prestan el servicio hacia los usuarios necesitan que la información esté disponible en tiempo de ejecución en la misma instancia de la aplicación

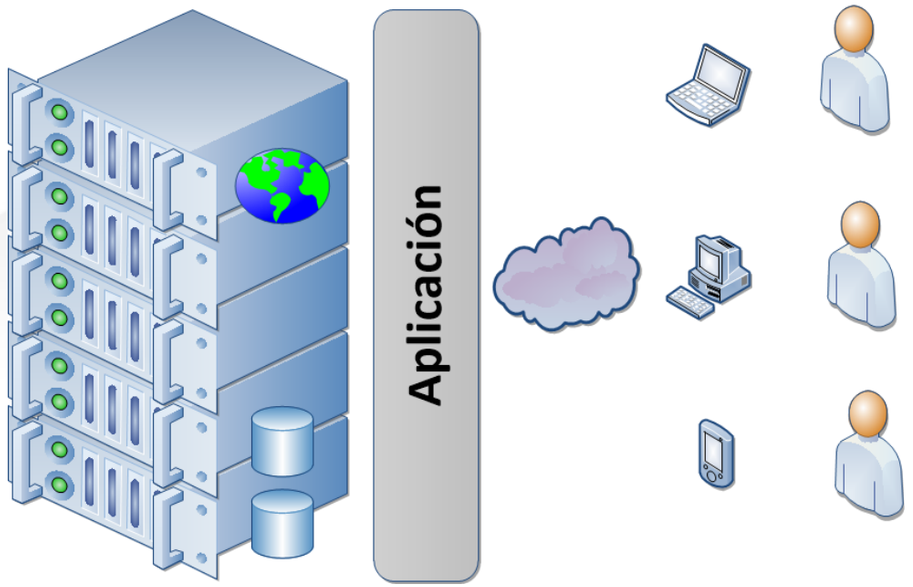


Escalabilidad

Escalabilidad Horizontal

Este tipo de replicación es posible cuando el software definido posee características o módulos débilmente acoplados, esto quiere decir que los componentes pueden ejecutarse en ambientes de memoria separados y que se comunican por medio de mensajes o protocolos específicos,

Replicación o Distribución de Componentes



Transparencia

Hace referencia a la capacidad que tiene el sistema distribuido, verse como un todo a pesar de los procesos que estén ejecutándose en el mismo. Así mismo el usuario debe percibir que esta interactuando con una sola entidad (sistema) y no con varias entidades más pequeñas que se complementan en funcionalidad entre ellas a pesar de que el sistema esté compuesto de varias partes, ya sean servidores, aplicativos, etc.

Relacionados con los Aspectos de Diseño

- **Transparencia de Concurrencia:** También conocida como paralelismo, posibilita al sistema que el usuario utilice varios recursos en el mismo instante de tiempo, producto del procesamiento de una tarea más grande o más compleja.
- **Transparencia de Replicación:** Autoriza el uso de recursos adicionales para completar la tarea en menor tiempo, es un efecto de la anterior, esto se explica mejor en el capítulo tres.
- **Transparencia de Fallos:** Si ocurre un fallo en el sistema, existe algún mecanismo de control que permite ya sea salvar la tarea del usuario y moverla a otro recurso activo o corregir el fallo y continuar con la tarea en el mismo recurso activo nuevamente.

Transparencia

- **Transparencia de Migración:** Habilita el movimiento de instancias de objetos, sesiones, etc., a otro recurso por la razón que sea, ya sea temas relacionados con el servicio o con fallos del mismo, este también es un efecto de la transparencia de fallos y de prestaciones.
- **Transparencia de Escalado:** Tolera el crecimiento o el decremento del tamaño del sistema según las necesidades del mismo, esto generalmente está asociado a la carga de los usuarios y a la cantidad concurrente de los mismos en el sistema. Este es uno de los principios del Cloud, llamado Elasticidad.

Exclusivos

- **Transparencia de Localización:** Permite que los procesos que el usuario ejecute puedan encontrar, información o recursos, sin importar en qué lugar del sistema se encuentren.
- **Transparencia de Acceso:** Permite el uso de la información o recursos previamente descubiertos, con el fin de que los usuarios perciban una única entidad o un sistema como un todo y no como un conjunto de partes.


INSTITUCIÓN UNIVERSITARIA
POLITÉCNICO
GRANCOLOMBIANO
¡GRACIAS!
POLI.EDU.CO
Poligran
MIEMBRO DE LA RED
ILUMNO