Sistemas Operativos II

Resumen Segunda Parte

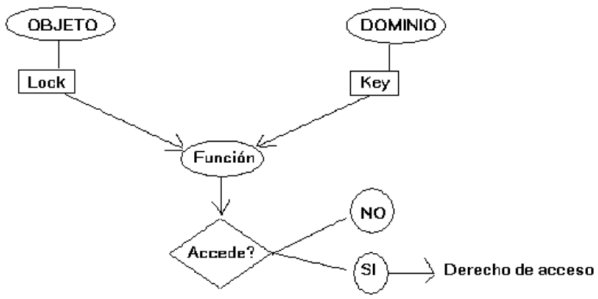
Emiliano Salvatori

Noviembre 2019

1. Clase nº 6

Mecanismo de Lock-Key

A cada objeto y cada domino se le asigna un número binario. Esos dos números se ingresan en una función booleana, entonces según el objeto que se tiene con el número binario, luego de la función se obtiene verdadero o falso por lo que de ahí se obtiene si se puede tener accesso al objeto o no.



Sistemas Confiables

¿Por qué los sistemas operativos no son absolutamente confiables dsde la seguridad?

- **Razón 1**: Porque los sistemas no son confiables, pero las personas lo usan igual, lo usuarios no quieren dejar de usarlos.
- **Razón 2**: para que un sistema sea confiable, debe ser simple. Como los usuarios quieren tener "características" en el software, por lo que el software se vuelve complejo y por ende el sistema se vuelven complejo, volviendose inseguros. La única forma conocida de construir un sistema seguro es mantenerlo simple. Las características son enemigas de la seguridad. Los usuarios quieren más características. Esto significa más complejidad, más código, más errores y más errores de seguridad.

Ejemplo

:

Cuando los servicios Web consistian en páginas de HTML pasivas, no imponía un problema grave de seguridad. Ahora que muchas páginas Web contienen programas (applets) que el usuario tiene que ejecutar para ver el contenido, aparece una fuga de seguridad tras otra. Tan pronto como se corrige una, otra toma su lugar.

Para construir un sistema seguro, hay que tener un modelo de seguridad en el núcleo del sistema operativo que sea lo bastante simple como para que los diseñadores puedan comprenderlo de verdad, y que resistan toda la presión de desviarse de este modelo para

Base de Cómputo Confiable

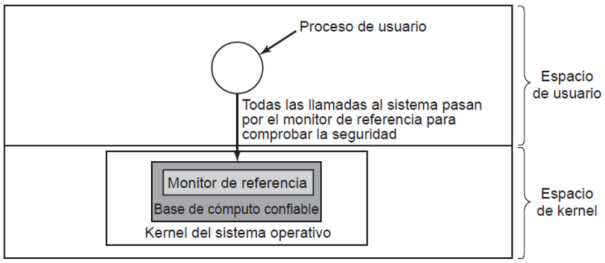
Aclaración: NO tiene nada que ver con las Bases de Datos Relacionales.

Modo usuario: Proceso usuario (que le hace llamadas al kernel para, por ejemplo, imprimir en pantalla) Cuando se hace una petición al kernel, baja el pedido y se encuentra con un programa que se llama **Monitor de la Base de Cómputo Confiable** que dentro tiene una Base de Cómputo Confiable.

Este analiza las llamadas y analiza desde el punto de vista de la Seguridad. Está dentro de una rutina del kernel. Este soft monitor, debe ser muy fácil y pequeño, para que sea lo más confiable posible (Es la misma idea del microkernel, que sea una rutina chica, simple, y óptima desde la perspectiva de la seguridad).

El monitor de referencia acepta todas las llamadas al sistema relacionadas con la seguridad (como abrir archivos), y decide si se deben procesar o no.

El monitor de referencia permite que todas las decisiones de seguridad se coloquen en un solo lugar, sin posibilidad de pasarlo por alto. La mayoría de los sistemas operativos están diseñados de esta forma, lo cual es una de las razones por las que son tan inseguros.



Uno de los objetivos actuales referentes a la búsqueda de seguridad consisten en reducir la base de cómputo confiable, de millones de líneas de código a unas cuantas decenas de miles.

Con MINIX 3, sólo se ejecutan aproximadamente 4000 líneas de código en el kernel. Todo lo demás se ejecuta como un conjunto de procesos de usuario.

Algunos de estos procesos, como el sistema de archivos y el administrador de procesos, forman parte de la base de cómputo confiable debido a que pueden comprometer con facilidad la seguridad del sistema.

Otras partes como el driver de impresora no forman parte de la base de cómputo confiable, por lo tanto pueden ser afectados por agentes externos, pero no comprometen al sistema.

Esteganografía

Estructrura de una imagen BMP: Considera una foto o dibujo como una cuadrilla o una matriz donde cada pixel en realidad tiene (por ejemplo si es de 24 bytes, porque cada 8 bytes represetan verde, rojo y azul, la mezcla que te da el color de cada pixel) Si se hace un zoom sobre el byte que hace de azul en la cuadrilla, se tiene un arreglo de ocho bytes,

Otro ejemplo, se toma una novela en txt, donde cada caracter es un byte, entonces si a ese byte se abre, se tienen 8 bits. Se quiere enviar por internet y ofuscarlo. Se puede tomar el bit menos significativo de esos bits del texto y se pueden meter en la imagen suplantando a cada bit de texto y se mete en el bit menos significativo de cada uno de los bits de la imagen. La imagen no cambia mucho, pero se tiene dentro de una imagen, un texto que se envía de forma encriptada, disfrazada.

Canales Encubiertos

Todas estas ideas sobre modelos formales y sistemas definitivamente seguros suenan bien, pero ¿en realidad funcionan? En una palabra: no.

Aun en un sistema que tenga un modelo de seguridad subyacente, que haya demostrado ser seguro y que esté implementado en forma correcta, puede haber fugas de seguridad.

A continuación analizaremos cómo puede haber fuga de información incluso cuando se haya demostrado con rigor que dicha fuga es matemáticamente imposible.

El modelo de Lampson se formuló originalmente en términos de un solo sistema de tiempo compartido, pero se pueden adaptar las mismas ideas a las LANs y otros entornos multiusuario.

En su forma más pura, implica tres procesos en cierta máquina protegida. El primer proceso (el cliente) desea que el segundo (el servidor) realice cierto trabajo.

El cliente y el servidor no confían completamente uno en el otro. Por ejemplo, el trabajo del servidor es ayudar a que los clientes llenen sus formularios fiscales. A los clientes les preocupa que el servidor registre en secreto sus datos financieros; por ejemplo, para mantener una lista secreta de cuánto gana cada quién, y después vender la lista. Al servidor le preocupa que los clientes traten de robar el valioso programa fiscal.

El tercer proceso es el colaborador, que está conspirando con el servidor para robar los datos confidenciales de los clientes.

El objeto de este ejercicio es diseñar un sistema en el que sea imposible que el proceso servidor filtre al proceso colaborador la información que ha recibido de manera legítima del proceso cliente. A este problema Lampson lo denominó **problema del confinamiento**

Desde el punto de vista del diseñador, el objetivo es encapsular o confinar el servidor de tal forma que no pueda pasar información al colaborador.

Mediante el uso de un esquema de matriz de protección, podemos garantizar que el servidor no se podrá comunicar con el colaborador mediante la escritura de un archivo al que el colaborador tiene acceso de lectura.

También podremos asegurar que el servidor no se podrá comunicar con el colaborador mediante el uso del mecanismo de comunicación entre procesos del sistema.

Por desgracia pueden existir canales de comunicación más sutiles. Por ejemplo, el servidor puede tratar de comunicar un flujo de bits binario de la siguiente manera:

- Para enviar un bit 1, realiza todos los cálculos que pueda durante un intervalo fijo.
- Para enviar un bit 0, permanece inactivo durante la misma cantidad de tiempo.

Explicación del profesor:

Se crea una rutina pequeña determinada Çolaborador". La cual se fija en la tabla de procesos y ver si el servidor está bloqueado o en proceso. El colaborador lo que hace es ver cuando está en proceso (haciendo cuentas) eso sería un 1, ahora bien, si se bloquea entonces sería un cero.

EL colaborador mide el tiempo del proceso por el PID y puede enviar ceros o unos dependiendo si está en proceso o bloquedo. Envía información encubierta.

La idea es que envía información encubierta dependiendo si el soft está haciendo cuentas (utilizando la ALU) o está bloqueado, enviando ceros y uno dependiendo de este estado.

Modelo Bell-la padulla

En el mundo militar, los documentos (objetos) pueden tener un nivel de seguridad, como no clasificado, confidencial, secreto y de alta confidencialidad.

A las personas también se les asignan estos niveles, dependiendo de los documentos que puedan ver.

Un general podría tener permiso para ver todos los documentos, mientras que a un teniente se le podría restringir a todos los documentos clasificados como confidenciales o con un nivel menor de seguridad.

Un proceso que se ejecute a beneficio de un usuario adquiere el nivel de seguridad del usuario. Como hay varios niveles de seguridad, a este esquema se le conoce como sistema de seguridad multinivel.

Básicamente se resume en lo siguiente:

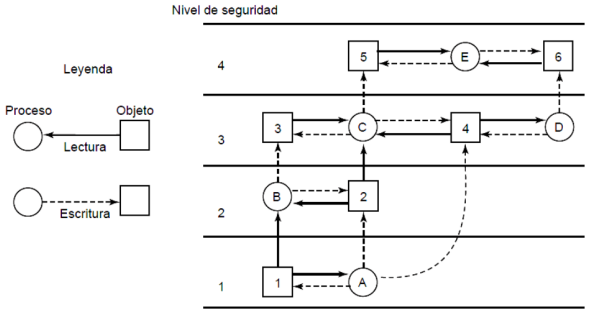
- **Regla 1**: Un archivo puede leer en su nivel o en uno inferior.
- **Regla 2**: Proceso puede escribir en su nivel o en uno superior

Esto se usó porque un proceso podía leer en un nivel inferior y reporta escribiendo un reporte hacia arriba. El que está en un nivel puede leer archivo en su nivel o inferior, pero NO superior.

En este modelo, los procesos leen y escriben objetos, pero no se comunican entre sí de una manera directa. El modelo Bell-La Padula se ilustra en modo gráfico en la figura siguiente:

Modelo Biba: ——— Funciona de la misma manera que el anterior, sólo que lo que hace lo contrario al modelo anterior. Regla 1: Un archivo puede leer en su nivel o en uno superior. . Regla 2: Proceso puede escribir en su nivel o en uno inferior

Según el profesor esto no funciona en organizaciones civiles, porque se pueden escribir de forma çorrupta"hacia arriba. Puede interrogar hacia abajo y escribir hacia arriba cualquier cosa maliciosa. Sólo funciona en organizaciones militares.



El modelo Bell-La Padula hace referencia a la estructura organizacional, pero en última instancia el sistema operativo es quien tiene que implementarlo.

Una manera de hacerlo es asignar a cada usuario un nivel de seguridad, que se debe almacenar junto con otros datos específicos del usuario, como el UID y el GID.

Al momento de iniciar sesión, el shell del usuario adquiere su nivel de seguridad, y todos sus hijos lo heredan.

Si un proceso que se ejecuta en el nivel de seguridad k trata de abrir un archivo u otro objeto cuyo nivel de seguridad sea mayor que k, el sistema operativo debe rechazar el intento de apertura.

Todos los intentos similares de abrir cualquier objeto de un nivel de seguridad menor que k para escribir en él deben fallar.

Modelo Biba

Funciona de la misma manera que el anterior, sólo que lo que hace lo contrario al modelo anterior.

- **Regla 1**: Un archivo puede leer en su nivel o en uno superior. .
- **Regla 2**: Proceso puede escribir en su nivel o en uno inferior

Según el profesor esto no funciona en organizaciones civiles, porque se pueden escribir de forma çorrupta"hacia arriba. Puede interrogar hacia abajo y escribir hacia arriba cualquier cosa maliciosa. Sólo funciona en organizaciones militares.

2. Clase nº 6

Seguridad

Las amenazas tienen cuatro objetivos generales:

1. **Integridad de la confidencialidad**: se debe asegurara la integridad de la cofidencialidad de los datos (como por ejemplo el mail). Para pinchar las comunicaciones de los celulares, es muy difícil corromper esto.
2. **integridad de los datos**: Para pinchar las comunicaciones de los celulares es muy fácil. Los ataques van por el lado de la integridad, que por la confidencialidad, es más difícil vulnerar la confidencialidad.
3. **Disponibilidad del sistema**: capacidad de estar online las 24hs. Facebook, y demás redes sociales no siempre están online, siempre algunas veces, en cambio hay sistemas que se deben estar disponibles 100 % como por ejemplo los sistemas para sistemas nucleares, lo que se hace es tener REDUNDANCIA. Es decir que todo el sistema tenga una copia que sea productiva.
4. **Exclusión de usuarios externos**: sistema de computo debe garantizar que no se permita el ingreso de usuarios no deseados. 4 tipos de estos usuarios:
    - **Usuarios no tecnicos**: usuarios pueden entrar de forma casual,como una secretaria que se mete sin querer a la parte del sistema que no le está permitido
    - **Estudiante/programadores**: usuarios técniso,que toman como cosa personal el entrar en un sistema de cómputo.
    - **Fines económicos**: por ejemplo del virus que encriptaba el disco que pedía bitcoins para poder desencriptarlos. Se utiliza la extorsión.
    - **Espionaje/con fines militares**: por ejemplo USA con dilma en Brasil.
    - **Virus**: Ataques con virus,gusanos, etc.

Causas de la Perdida de datos:

1. **Accidentes**: por ejemplo, se prende fuego el edificio donde se hospedan los servidores de bases de datos.
2. **Errores de hard y software**: puede pasar que falle el hardware, dependiendo de la calidad del mismo, lo que se requiera, etc. Por ejemplo la disponibilidad depende del dinero que se invierta para tenerlo útil el tiempo que se requiera.
3. **Errores humanos**: por ejemplo, manipular la base de datos de forma equívoca, y se pasó todo el fin de semana arreglándolo.

Criptografía

La criptografía (literalmente *escritura oculta*) se ha definido, tradicionalmente, como el ámbito de la criptología que se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados. Estas técnicas se utilizan tanto en el arte como en la ciencia y en la tecnología. Por tanto, el único objetivo de la criptografía era conseguir la confidencialidad de los mensajes, para lo cual se diseñaban sistemas de cifrado y códigos.

La aparición de la informática y el uso masivo de las comunicaciones digitales, han producido un número creciente de problemas de seguridad. Las transacciones que se realizan a través de la red pueden ser interceptadas, y por tanto, la seguridad de esta información debe garantizarse. Este desafío ha generalizado los objetivos de la criptografía para ser la parte de la criptología que se encarga del estudio de los algoritmos, protocolos (se les llama protocolos criptográficos), y sistemas que se utilizan para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican.

La idea es que se tiene un mensaje en un texto llano, por ejemplo un caracter o una serie de caracteres como "Hola" por ejemplo. Se aplica a ese texto llano una transformación y se envía por algún canal, por ejemplo internet. Se envía el mensaje cifrado.

Del otro lado se aplica otra transformación (puede ser la misma a la inversa, en ese caso se habla de **transformaciones simétricas**) para poder volver al estado original del texto llano, quedadndo de forma igual al original. Cuando se aplica OTRA transformación del lado del receptor, se denomina **transformaciones asimétricas**.

Primeros intentos criptográficos

¿Cuál fue los primeros métodos criptográficos? Los griegos. Estos, tomaban una madera y le daban una forma especial (octoganal) tomando un cuero y lo envolvían en ángulo, lo enroscaban en ángulo, por lo que se escribía sobre el cuero enroscaado y cuando se desenroscaba, quedaba un texto inentendible. Se requería tener el palo en el lado del receptor. En este caso es simétrico porque se requiere el mismo palo para volver a formar el texto original. Este método se denomina: escítala.

El sistema consistía en dos varas del mismo grosor que se entregaban a los participantes de la comunicación. Para enviar un mensaje se enrollaba una cinta de forma espiral a uno de los bastones y se escribía el mensaje longitudinalmente, de forma que en cada vuelta de cinta apareciese una letra de cada vez. Una vez escrito el mensaje, se desenrollaba la cinta y se enviaba al receptor, que sólo tenía que enrollarla a la vara gemela para leer el mensaje original.

Método César

Se toman una cadena de caracteres y le suman un offset (por ejemplo +3). Por lo que si el texto llano es la palabra "HOLA", entonces aplicado éste método de criptografía se obtendría: "KRÑD" (Mensaje encriptado).

Siempre depende del abecedario que se utiliza. Como el que envía suma 3, entonces el receptor debería de restarle 3. Este método también es simétrico. En el examen entra un ejercicio en C y lo que se hace es sumarle un número a cada caracter mediante un *for*.

One Time Pad (Bloque de Uso único)

Consiste en una **variación** del método del césar. Siguiendo con el ejemplo anterior, entonces al primer caracter se le suma +3, al segundo se le hace -1, al tercero -2, y al cuarto +2, por lo que la palabra "HOLA" quedaría "KPJB" (mensaje encriptado) siendo la la clave: +3, -1, -2, +1.

Es simétrico también porque se le debe invertir las operaciones del lado del receptor. El problema que tiene éste método, que la clave es tan larga como el texto enviado, si esto se utilizaría en internet, se debería de tener el doble de ancho de banda para enviar un texto.

Red de Faistel: La transformación se realiza mediante hardware, venían diseñado en cajas.. Dos conceptos que se utiliaba en esto. Se trabajaba con OR exclusiva:

- **Confusión**: de alguna manera de lo que entra salga cambiado en la salida.
- **Difusión**: el peso de cada bit,se viera repartido su influencia de manera pareja al bit de salida. Que un bit influya en todos los bits de salida.

Sustitución por palabra clave

Se deben seguir 4 pasos para este tipo de método criptográfico:

- **Generación de la palabra clave**: Se toma una palabra, como por ejemplo ARTURO JAURETCHE que va a ser nuestra clave. Se le aplica el algoritmo a la calve. Se toma el primer caracter A y se pregunta "¿esta es la priemra vez que aparece en la cadena? SI, entonces se deja. Se toma el siguiente caracter R y se pregunta lo mismo. NO, se deja la R. La T ¿apareció anteriormente? NO Se continúa este paso hasta acabar con todas las letras de la palabra clave, en caso de que se repita se elimina de la palabra clave.

- **Nuevo Abecedario**: La palabra obtenida del punto anterior sería: "ARTUOJECH" ya que las demás letras fueron apareciendo de forma repetida. Se copia el abecedario que se tendrá en el examen:

  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

  A R T U O J E C H B D F G I K L M N P Q S V W X Y Z

  Para la obtención del segundo abecedario se procedió a preguntarse por cada letra: ¿Aparece la A del abecedario? Si, por lo que se pasa al siguiente. ¿La B aparece? NO,entonces baja

  Ver que Las últimas letras VWXYZ aparecen tal cual como el abecedario, esto es porque la mayor letra es la U dentro de la palabra clave, por por lo que para que quede todo el abecedario modificado deberían de tener muchas letras y tener la Z. Quedan dos abecedarios: el de arriba como si fuera el de texto llano y el de abajo el codificado.

- **Obtención del mensaje codificado**: Una vez obtenido el abecedario codificado, se sitúa la palabra llana "HOLA" sobre el abecedario y se obtiene *canjean* las del abecedario normal con las del abecedario modificado: ver que la palabra resultante es "CKFA".

Ver que ambos abecedarios en la parte final de la V hasta la Z quedan iguales. Para que esto sea mejor, se debe tener una letra más alto,en ARTURO JAURETCHE, el caracter más alto es la U; por eso es que es preferible que haya una Z para asegurara que el abecedario encriptado esté modificado completamente.

La clave sería ARTURO JAURETCHE. En el examen viene una clave, se hace el primer paso y luego el abecedario La palabra considerada como texto llano que se debe ENCRIPTAR NO va a tener sentido, se puede pedir encriptar tipo JOGGSA. Eso hay que hacer como con la palabra HOLA del ejemplo anterior.

Métodos Asimétricos

Hasta ahora el receptor aplica una transformación inversa; acá se utiliza un método asimétrico y se obtiene el mismo resultado.

Se deben buscar dos números primos lo más grandes posibles: P y Q, donde $N = P * Q$

Luego se buscan dos números E y D tal que: $E * D * mod((P-1)*(Q-1)) = 1$

Siendo P = 3 y Q = 5:

$P = 3 y Q = 5 \Rightarrow N = 3 * 5 \Rightarrow N = 15$

Siendo E = 3 y D = 11, se aplica la siguiente ecuación: $E * D * mod((P-1)*(Q-1)) = 1$

Se pregunta:

$$3 * 11 * mod((3-1)*(5-1)) = 1?$$

Se obtiene: $33 * mod(8) = 1?$ por lo que se pregunta: Si se divide 33 por 8, ¿el resto es 1?

Efectivamente el resto es 1. Esto quiere decir que los 4 números sirven para aplicar el método de encriptación del método asimétrico.

Los 4 anteriores números sirven para encriptar números en un rango entre: $0 \leq X \leq N-1$

Siendo N un solo bit que es 15. Para Cuanto más grande sean los primos P y Q será más grande el número que se podrá encriptar. En este ejemplo irá entre 0 y 14.

Para Encriptar

Se elije M = 8. C = número encriptado; y M = número llano.

Para encriptar se aplica: $C = M^E Mod(N)$

Entonces:

$$C = 8^3 mod(15) \Rightarrow \frac{512}{15} = 34,13$$

Se hace en la calculadora: $15 * 34 = 510$ y la diferencia entre 512 y 510 es 2, por lo que C = 2.

Para Desencriptar

$$M = C^d mod(N)$$

$$M = 2^{11} mod(15)$$

$$M = \frac{2048}{15} = 136,53$$

Se toma $136 * 15 = 2040$. La diferencia entre 2040 y 2048 es 8 y es el número encriptado $M = 8$

Lo interesante es que se encriptó con el par E y N Y se desencriptó con D y N. El que desencripta NO sabe de E. En el examen le viene P Q D Y E, se verifica que el resto de 1, con la siguiente ecuación:

$$E * Dmod((P-1)*(Q-1))) = 1$$

Si NO da 1, entonces FINALIZA AHI. Si cumple entonces se da un número y se debe verificar que esté DENTRO DEL RANGO, en el ejemplo anterior por ejemplo 27 estaría fuera del rango ya que va sólo de 0 a 14.

Y para aplicar encriptar y desenpcriptar se utilizan las fórmulas antes vistas.

Chip TPM

Circuito integrado diseñado que iba en las placas madres que hacía criptografía por hardware. Entraba una palabra y salía encriptado. Le gustó a Microsfft para que tenga esto el mother y que se utilice para poder verificar el software para que el ususario pague. Implementación de un método criptográfico mediante un hardware

Autenticación: las formas de autenticar identidad son:

- Sabe (usaurio y contraseña de home banking)
- Tiene (la tarjeta SUBE o la tarjeta para fichar)
- Es (cuandos se pide las huellas dactilares).

Contraseñas: en un tiempo el inicio de seción se pedía: Usuario y contraseña. Se hacía una pantalla que decía que el error o estaba en el usuario o en la contraseña, eso es tener una ventaja porque se sabía por dónde estaba el error y dónde se había ingresado algo bien. Por lo tanto se debe utilziar un error que sólo diga .^{er}ror en la sesión".

TELNET: se logueaba con un usuario y se pedía una IP. Era como un usuario remoto, que pedía una IP para pegarle, el tema es que los hackers tomaban una lista de nombres enomres y las constraseñas eran esos mismos nombres + números. Y se realizaban scripts para poder bombarder IP públicas como por ejemplo de las universidades.

$Y = f(x)$. La idea es que X sea la contraseña, y F es una función criptográfica, e Y es el archivo de cómputo. De ida, es decir teniendo la contraseña es fácil, pero sin tener la contraseña es difícil. La idea es que el archivo que guarda las contraseñas (en este caso Y) aplique un método criptográfico a las contraseñas guardadas.

Lo que hace es aplicarle n transformaciones, por ejemplo: $y = f(f(f(x)))$ donde se aplique $n = 3$ transformaciones.

En la próxima vuelta que se inicia el sistema lo que se hace es n = 2, $y = f(f(x))$

Y así cada vez que se inicializa la máquina. El tema es que se supo qué función se aplicaba por lo que quedó.

Retro-Respuestas: se realizan preguntas que sólo el usuario sabe "¿Cómo se llama tu perro?"

Ctrl+Alt+Supr: es una combinación por hardware para poder iniciar sesión de forma directa, esto se hace para que no haya ningun software malicioso que se interponga entre medio de el login para obtener los datos de la cuenta y la contraseña.

Ataques desde el interior: programador antes de irse de una empresa que deje una "bomba lógicaÜn programa que explote dentro de X cantidad de días. Eso se hace dejando corriendo en un servidor preguntando por la fecha constantemente,cuando le da verdadadero esa fecha, que borre los archivos de la base de datos .

Multiprocesadores

Las disciplinas que requieren mucho poder de cómputo (cómputo masivo):

- **Astronomos**: modelan por ejemplo todo el sistema solar, el sentido en que se mueven y todo los cálculos pertinentes a al modelo. Se puede modelar tanto hacia el futuro como para el pasado.
- **Biologos**: Se requieren simulaciones para modelados de la vida natural.
- **Ingenieros**

Limitantes

Limitantes para generar un procesado más poderoso:

- **Velocidad del Reloj**: todos ls citcuitos funcionan con una frecuencia que le va marcando el tiempo. ¿se puede aumentar la velocidad del reloj? Si, pero es difícil extraer el calor con altas frecuencias del reloj. La otra alternativa para mejorar el circuito es bajar el tamaño de los circuitos. **bajar el tamaño de los transistores**: todos los circuitos lógicos que se conocen. con un and, or y con el not se puede hacer casi todo. para que funcione esto se tienen los transisotes, el cual es un circuito que se basa en las compuertas antes mencionadas. transistores símil a los ladrillos con los que está construido un circuito. cuanto más chico sea el ladrillo más espacio en la pared. el mismo circuito pero es más chico, por lo que todo el área que sobra se utiliza para mejorar el circuito. lo que sobraba le mandaban la memoria cache, que aumenta mucho el rendimiento del microprocesador. ahora casi la mitad del circuito es memoria cache. también tiene un límite ir aumentando la mem cache. **¿por qué no se aumenta el tamaño de la pared o área del procesador?**: eso es una cuestión de cómo se fabrican los microprocesadores. todos los micros se fabrican con silicio, que es como un vidrio que se hacen como cilindros de silicio y lo van cortando en fetas,como si fueran un panqueque. cuando se calienta un material se desordenan los átomos, se vuelve blando, cuando se enfría de forma muy rápido, no se le da tiempo para que los átomos se ordenen y quedan desordenados en estructuras cristalinas, por lo que faltan átomos por esta falta de orden, lo que hace todo esto es que quede más duro, por eso el herrero es que cuando forja una espada la enfría de repente. cuando se genera estas barras de silicio se enfria de forma muy despacio para que queden la menor cantidad de errores posibles para que cuando se proyecta el micro no termine funcionando de forma incorrecta. se trata de que cuando se fabrique, se haga de forma con menos errores posibles sobre el silicio. si se hace más grande el micro, cuando haya un error el descarte cuando hay un error en el silicio, se debe descartar una pedazo más grande que si se hiciera un pedazo más chico. es una cuestión costo beneficio para la empresa de fabricación de procesadores. **Se hace más chico el ladrillo**: se achica hasta que deja de funcionar como transistor. se fabrican con cosas denominada sustratos, por lo que cuanto más chico se hace, deja de funcionar como tal el material. hay un límite en los materiales de construcción. se llegó al límite por lo que en vez de achicar más el transistor, lo que se hace es ir poniendo más. Lo que dio como resultado la computación paralela.

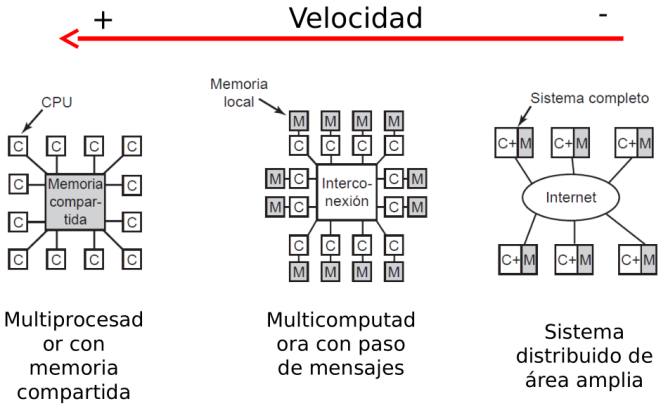
Computación paralela

Se tienen tres formas:

1. Una forma de hacerla es tener una memoria compartida junto con n cantidades de core. (mayor velocidad, porque tiene una memoria compartida que tiene distitnos cores). Esto se parece a un quad core, parecida a las memorias utilizadas en el celular.

2. Se tienen los cores con algunsa memorias privadas + la memoria compartida. Esto es parecido a la estructura de la Play Station. Cada micro tiene asociada una memoria privada.

3. Se tiene conectada cada CPU mediante internet (menor velocidad porque está en el medio internet, que comunica cada nodo, el cual es un procesador)



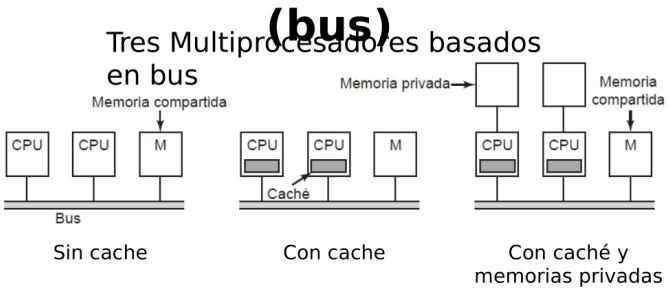
Multiprocesadores UMA

Multiprocesador: sistema de computo en que dos o más CPU comparten una RAM en común. Es decir los primeros dos puntos del punto anterior (sin el de internet)

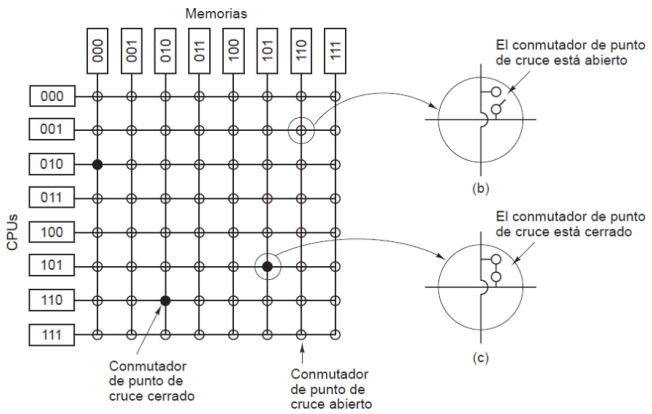
Tipos de multiprocesadores

UMA: Acceso uniforma a memoria. Cuando todos los procesadores tienen la misma capacidad de acceso a la memoria. Tipos de UMA:

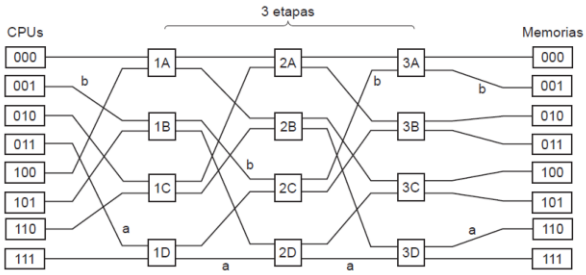
- **BUS**: forma más común de hacer UMA. El problema que tiene es que el cuello de botella es el bus, si se tiene muchos procesadores y lo quieren utilizar todos a la vez entonces se tiene que esperar a que se desocupe para utilizarlo. Acceso uniforme a memoria porque todos están de forma equidistante al acceso a la memoria.
- Bus con varios procesadores conectadas junto con una memoria compartida. Arquitectura vieja, sacada del libro
- Cada procesador tiene una memoria cache asociada. Parecido a una arquitectura de procesador de celular.
- También existe esta última forma con memoria privada cada uno de los procesadores. Arquitectura tipo PlayStation. Es acá donde se pone mucha energía en desarrollar cosas de vanguardia. El problema que se tiene también al poder comunicar tantos procesadores es quién está modificando el pedazo de memoria accedida por un



Barras cruzadas: se tienen los procesadores puestos en parte izquierda y como en una grilla distitnas memorias, entonces para que cada procesador quiera acceder a determinada memoria compartida, se debe cortocircuitear en la grilla para poder acceder a la misma. Sería parecido a un excel, donde cada fila es un microprocesador y cada columna vendrían a ser microprocesadores.



Multietapas: si un microprocesador 1 quiere comunicarse con la memoria 5 entonces cada uno de los micros tienen que proceder a habilitar y desabilitar las entradas y salidas en todo el



NUMA: Acceso NO uniforme a memoria.

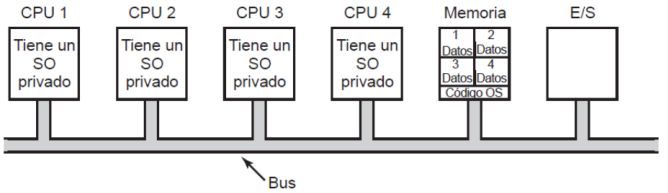
Ley de Moore

Moore era gerente de INTEL. Que determinó una relación en la cual cada dos años o 24 meses la cantidad de transistores por chip se iban duplicando. La cuestión es que después del 2000 se saturó, llegó a un límite, donde practicamente no se mejora tanto. Hoy en día NO se cumple la ley de Moore porque no sigue habiendo una mejora constante que impacte de manera significativa a la tecnología; no se puede construir chips más chicos, por lo que la mejora no es tan considerable como en los años 70, 80 y 90.

Sistemas Operativos para Multiprocesadores

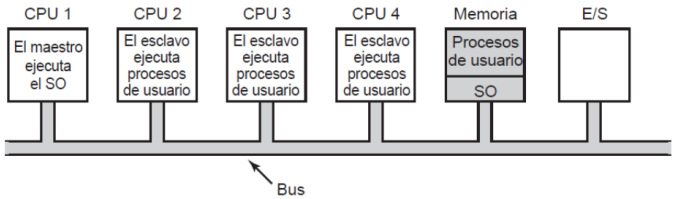
Si se tienen varios procesadores ¿qué sistemas operativos se le pueden hacer correr que permita este tipo de arquitectura? Deben ser unos SO para multiprocesadores. Se tienen 3 tipos de SO para multiprocesadores:

1. Cada micro implementa un SO privado y todos estos micros están conectados mediante un bus. También tiene una memoria compartida conectada al bus, entonces como cada micro corre una instancia del SO. Entonces cada micro utiliza una parte asignada para él mismo de la memoria.



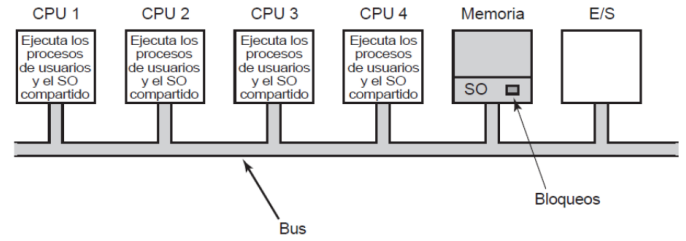
Problemas de distribución de recursos: posiblemente se cargue mas un procesador que otro, desperdicio de disco por el uso de la cache

2. Se tiene otra forma de que haya un micro que tenga un SO privado que sea maestro y todos los demás microprocesador son esclavos que tienen un proceso esclavo corriendo (en el maestro corre por ejemplo el SO principal y en los otros micros corren el Explorer, el Crhome y otro programa). También se tiene una memoria principal que comparten entre todos los micros mediante un BUS. La memoria está dividia entre la memoria del SO y la memoria para correr un proceso de usuario. El problema que tiene es que el bus es un cuello de botella si todos tienen que interactuar con el micro Maestro.



Problemas: si hay muchos CPUs: CPU1 -> cuello de botella

3. Cada uno de los micros se pueden correr una instancia del proceso de usuario y luego un proceso del sistema operativo. La memoria sigue esstando dividida entre el proceso de SO y de memoria. [Imagen de celular de abajo de todo]



Problemas: si dos o mas CPUs están ejecutando código del SO al mismo tiempo, problemas de ejecución, paginado, interbloqueo.
Solución: **MUTEX**

Multicomputadora

Cluster: se toman n CPU conectadas por una red y se generan lo que se denominan Beowulf Cluster. Se juntan máquinas viejas y se conectan vía switchs. Los rendimientos se miden contando los millones de operaciones de punto flotante que puede hacer el cluster por minuto.

Casos emblemáticos de PlayStation: cuando queda parada en un determinado protector de pantalla, se conecta desde internet solicitando poder de cómputo, cuando el usuario vuelve, el resultado se envía por internet. El poder de cómputo de la PS en todo el mundo es comparable con las computadoras listadas en top500.org.

Formas de generar un Sistema Distribuido

- Red Estrella donde en el medio hay un swtich.
- Red comunicada en forma de anillo, en IBM esto era muy utilizado. Token RING, los datos daban vueltas en forma circular. El problema que tenía que es que si se corta el cable, las computadoras quedan incomunicadas. Quedó sólo implementado en los microprocesadores de IBM.
- Mesh es una Grilla, que interconecta cada CPU como si fuera una matriz. También se utiliza de forma toroidal, es decir que cada CPU se pueda comunicar con las de más abajo.

Granularidad: tiempo de computo sobre el tiempo de comunicación.

Se tiene un cluster y se le hace correr un programita y el programa está corriendo en cada nodo y muy de vez en cuando comparte información con otro nodo. Si casi siempre se utiliza mucho tiempo de computo y poco tiempo comunicandose entonces la granularidad es grande.

Si se tiene mucha comunicación y poca proceso de cómputo, entonces la **granularidad es muy baja**.

Granularidad gruesa: Más tiempo cuando se está más tiempo computando que comunicandose Granularidad fina: Más tiempo cuando se está más tiempo comunicandose que computando. Cuando da 1 es que tiene el mismo tiempo haciendo cuentas que comunicándose.

Dependiendo de cómo es el problema que uno quiera abocarse se debe hacer analizar el tipo de granularidad. Si se quiere tener mucho poder de cómputo o mucho poder de comunicación.

Las preguntas del examen puede ser: Si se tiene un sistema de granularidad gruesa y se quiere implemnetar un cluster de PS en Internet ¿es posible? Si, porque se tiene mucha capcidad de cómputo y poca velocidad en internet.

- **Lib de C MPICH**: librería que se utiliza para hacer programas para correr en n nodos de un cluster. Con esta lib permite distribuirla de una forma distribuida.

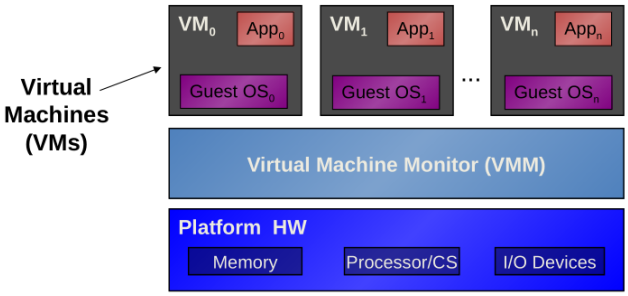
Virtualización

Virutalización: *una máquina virtual es un duplicado de una máquina real, eficiente y aislada.* Significa que lo que hace la máquina virtual es agregar una capa extra.

- **Duplicado**: La MV se debería comportar de forma idéntica a la máquina real, excepto por:
    1. La existencia de menos recursos disponibles (incluso diferentes entre ejecuciones).

2. Diferencias de temporización al tratar con dispositivos.

- **Aislado**: Se pueden ejecutar varias MV sin interferencias.
- **Eficiente**: La MV debería ejecutarse a una velocidad cercana a la del HW real.
  - Requiere que la mayoría de las instrucciones se ejecuten directamente por el Hardware.



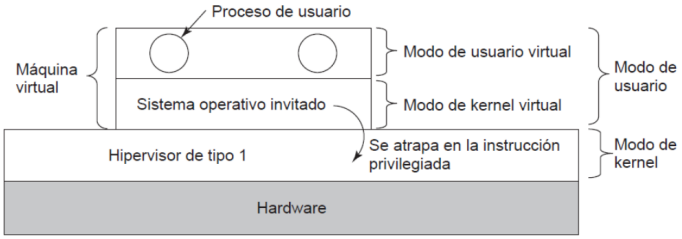
Windows corre sobre cierto Set de instrucciones (depende de cómo esté fabricado el micro va a tener la instrucción de ADD Ra, Rb, Rc) y que quizás en INTEL es de una forma, y en MAC esté implementado de forma distinta. Por lo tanto es necesaria determinados hardware para poder correr determinados Sistemas operativos.

Entre el Hardware INTEL Y AMD el Set de Instrucciones sea exactamente igual, no puede haber alguna variación, ya que agregándole un bit ,rompe todo. ¿Cuál es la diferencia? Que pueda ser un mejor la ALU o más rápida, pero el set de instrucciones debe ser igual.

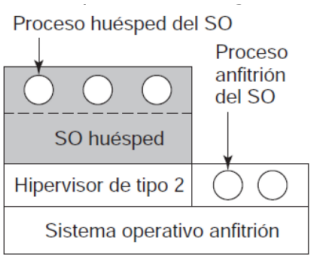
La máquina virtual, lo que hace es agregarle una capa abstracta entre el hardware y un SO, el SO (windows, por ejemplo) tiene que mirar para abajo y ver que hay determinados set de instruciones que requiere, pero la máquina virtual convierte ese set de instrcciones que necesita windwos en el set de instrucciones de MAC por ejemplo, en caso de quese esté corriendo una VM sobre MAC

Hipervisores

Hipervisor tipo 1: Donde se tiene el HW (a la derecha se tiene el SO que se está instalado) y el modo kernel interactua con el HW de forma normal. Pero hacia la derecha se tiene una VM que interactua con el hardware de la máquina, se instala a nivel de SO, y sobre esta VM puede correr el SO que uno quiera (SO invitado se denomina), donde también tiene su respectivo modo Usario y Kernel Virtual, porque corren sobre la VM.



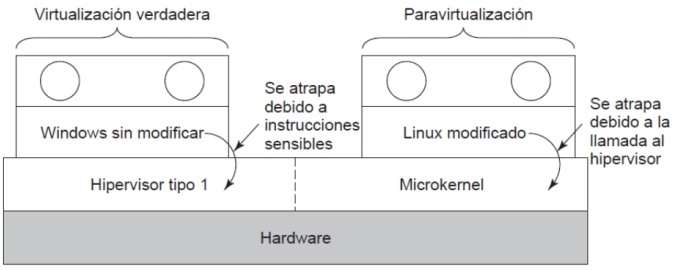
Hipervisor tipo 2: Donde se tiene el HW y se tiene un SO sobre este mismo. Y luego en MODO usuario está corriendo una VM de tipo Hipervisor tipo 1 que está corriendo en modo usuario. EJemplo de esto es una Java virtual Machine.



La idea de la VM es que reproduzca el HW real.

Paravirtualización

- Se modifica el código fuente del SO invitado, de manera que en vez de ejecutar instrucciones sensibles, realiza llamadas al hipervisor.
- El hipervisor debe definir una API para SO invitado.
- El hipervisor pasa a ser un microkernel



Para virtualización es que se modifique el código de la VM para poder hacer más poderosa la virtualización de determinado HW para un determinado SO. De esta forma se genera mejor la eficiencia (ver mejor en las filminas porqu elo explicó por encima).

Migración en vivo: consiste en tener un HW con una VM con un SO corriendo arriba y luego otro HW con su VM, sin ningun SO, entonces el SO primero, se puede pasar por red y ponerlo en el segundo, todo SIN parar el SO. Esto se hace para poder pasar el SO de un lado al otro y usarlo por ejemplo en servidores, y de esta forma hacerles mantenimiento.

Cuadro Comparativo

NODO

- **Multiprocesadores** (como la maquina que está en el celular): cada nodo es un microprocesador o core.
- **Multicomputadora**: cada nodo es un CPU entero (memoria, almacenamiento, disco tradicional, interfaz de red) pero de forma parcial.
- **Sistema distribuido**: cada nodo es una CPU TOTAL.

PERIFERICOS:

- **Multiprocesadores** (como la maquina que está en el celular): todos los perifericos estan compartidos, se comparten entre los micro que los comparten perifericos
- **Multicomputadora**: es lo mismo que el Multiprocesadore, pero puede a veces que no compartir el disco.
- **Sistema distribuido**: Los nodos no comparten nada, como por ejemplo el cluster de PS, donde no comparten nada entre nodos, entre PS. Los periféricos no se comparten.

UBICACION

- **Multiprocesadores** (como la maquina que está en el celular): Los nodos que están dentro de un celular, están dentr ode un mismo lugar, misma ubicación.
- **Multicomputadora**: generalemnte están en el mismo cuarto.
- **Sistema distribuido**: justamente están distribuidos a nivel mundial, o en el mismo pais, o mismo continente.

COMUNICACION:

- **Multiprocesadores** : se comparte memoria.
- **Multicomputadora**: una red dedicada. Depende el cluster cómo sea, se puede tener un switch o una máquina más sofisticada con una red dedicada.
- **Sistema distribuido**: red tradicional, se utiliza internet, con algunsa veces con fibra óptica dedicada.

SISTEMAS OPERATIVOS:

- **Multiprocesadores** : el mismo, un dual core, pero corre android (en el celular por ejemplo)
- **Multicomputadora**: generalmente el mismo, en un claster, se hace correr el mismo SO para todos los nodos.
- **Sistema distribuido**: distinto, un cluster con un SO que funciona de determinada manera, pero otro cluster de otro lado puede tener otro HW, con otro SO más nuevo pero ambos funcionan intercambiándose de forma efectiva la carga.

Nota ARM: ellos te dan un circuito base, te brindan una herramienta de software para modificar ese circuito base, que le agregeue más harwarea al procesador, por lo que se tiene un procesador dedicado. Luego cada usuario que diseño ese circuito lo manda a fabricar DONDE mejor le convenga. Procesadores que NO están fabricadas con la última tecnología a diferencia como en INTEL, por lo que se vuelve muchísimo más barato.

Middleware

Las aplicaciones comunes de Internet incluyen el acceso a computadoras remotas (mediante el uso de telnet, ssh y rlogin).

El acceso a información remota (mediante el uso de World Wide Web y FTP, el Protocolo de Transferencia de Archivos)

Comunicación de persona a persona (mediante el correo electrónico y los programas de chat)

El problema con todas estas aplicaciones es que cada una tiene que reinventar la rueda.

Una de las formas en que un sistema distribuido puede obtener cierta medida de uniformidad frente a los distintos sistemas operativos y el hardware subyacente es tener un nivel de software encima del Sistema Operativo: Middleware. Es decir, son Sistemas Operativos para sistemas distribuidos.

Por ejemplo:

- Se suponen 4 HW distintos: INTEL, AMD, MAC, ARM (micros que tienen nuestros celulares)
- Corriendo arriba 4 SO distintos: Windows, Linux, MacOS, Android.
- Se requiere: Correr un programa que se quiere hacer correr por n nodos distintos (los anteriores) en hardware como en SO, por lo que se implementa un Middleware que corra en todos los nodos distintos, y sobre este Middle corre por encima de este Middle. TOdo el SIstema distribuido se conecta por internet. Ejemplo más común de Middle: internet, gmail por ejemplo: sin importar el HW o el SO que se utilice, todos pueden visualizar Gmail.

Grid

Es básicamente la infraestructura que se necesita para poder acceder a ese computo masivo.

Analogía

Se realiza una analogía entre una red eléctrica y una red de cómputo masivo.

Red Eléctrica

- **Infraestructura**: crentral terminca, represa hidro, tendidos de alta tensión donde está la demanda, tendido urbano y la red hogareña por último.
- **Red eléctrica**: es transparete, se enchifa la plancha y no se sabe de dónde viene esa electricidad.
- **Es penetrante**: porque está en todos lados, no hay lugar donde no llegue.
- **Es servicio**: se solicita y vienena instalarlo

Red de Cómputo Masivo

- **Infraestructura**: primero un cluster con n nodos conectados entre si en algún lugar del planeta, se requiere una red conectada a internet, y luego hasta que llega a la terminal del usuario.
- **Será Transparente**, un lugar donde se pude acceder que requiera computo masivo, introducir el programa que corra sin saber en dónde se está corriendo, sólo se consumen los datos. Esto es TODAVIA un DESEO.
- **Será penetrante** también (expresión de deseo) de forma que se pueda acceder desde cualquier lado.

- **Es servicio**: se contrata a una empresa y se puede utilizar el computo masivo, conviene alquilar en vez de construir un cluster completo y dedicado. Red Clara: conecta Latinoamerica Geant: conecta Europa

¿Qué se se corre sobre la grid? Aparece el concepto de Cloud Computing, se accede desde la red a una abstracción de la grid. Cuando nos metemso al google Drive que no se sabe dónde está corriendo o está situado fisicamente, nos conectamos subimos una foto, archivo y laburamos sobre la nube, pero sin saber dónde se hospeda.

Conceptos

Grid: relacionado al hardware
Clound Computing: es la abstracción de la grid, es como el software que corre sorbre la grid.
Tipos de clound computing:

- Pública
- Privada
- Híbrida: como por ejemplo lo de la facultad, parte privada y parte pública.

Clound Computing:

- SAAS: software com un servicio. software de alto nivel para solucionar problemas a alto nivel
- PAAS: se requiera "tunear"la app brindando más información del cluster para poder mejorar la performance
- IAAS: se requiera toda la información del cluster para que pueda correr de la mejor manera sobre ese cluster.

Empresas que brindan este servicio: amazon, facebook, google, tienen servidores ociosos, lo que hacen es vender cómputo masivo. IBM, DELL, Sun Microsistem: los que fabrican hw todos tienen un cluster y ofrecen cómputo masivo porque dicen mira los servidores que fabrico, es para poder testearlo y que los usuarios lo testeen para poder luego comprarlos.

Clientes de estos servicios

- Universidades por sobre todo
- Por ejemplo empresas Loreal: simulacion de procesos orgánicos,
- Cuando se fabrican circuitos integrados, para probarlo antes de mandarlo al mercado, se simula por software las entradas aleatorias para ese circuito.