

## LPIC-2 / Examen 201 - Capacidad del Sistema

### 200.2 Predecir necesidades futuras de recursos

#### Teoría

La predicción de necesidades futuras de recursos, también conocida como Planificación de Capacidad (Capacity Planning), es el proceso de estimar cuántos recursos de hardware (CPU, RAM, almacenamiento, ancho de banda de red) necesitará un sistema o servicio en el futuro. Esto es esencial para:

- **Evitar Problemas de Rendimiento o Caídas:** Asegurar que el sistema pueda manejar cargas crecientes antes de que los recursos se agoten.
- **Optimizar la Inversión:** Comprar o asignar recursos de manera eficiente, evitando gastos innecesarios.
- **Planificar Migraciones o Escalabilidad:** Determinar cuándo es necesario migrar a hardware más potente o escalar horizontalmente (añadir más servidores).

#### Metodología para la Planificación de Capacidad:

1. **Monitorización Continua:** Recopilar datos de rendimiento de los recursos clave (CPU, memoria, disco, red) de forma regular y a lo largo del tiempo.
2. **Análisis de Datos Históricos:** Examinar los datos recopilados para identificar:
  - **Tendencias:** ¿El uso de CPU, memoria o disco está creciendo linealmente, exponencialmente o se mantiene estable?
  - **Patrones:** ¿Hay picos de uso en ciertos momentos del día, días de la semana o meses del año?
  - **Niveles de Uso Promedio y Pico:** Conocer tanto el uso típico como los máximos alcanzados.
3. **Identificación de Cuellos de Botella:** Basado en el análisis, determinar qué recurso es probable que se agote primero bajo una carga creciente.
4. **Proyección:** Estimar cuándo el recurso identificado como cuello de botella alcanzará su capacidad máxima aceptable, basándose en las tendencias de crecimiento y las proyecciones del negocio o del uso de la aplicación.
5. **Planificación de la Acción:** Basado en la proyección, determinar cuándo y qué tipo de actualización o escalado se necesita.

#### Herramienta Clave para Datos Históricos: **sar** (System Activity Reporter)

El comando **sar**, parte del paquete **sysstat**, es la herramienta estándar en Linux para recopilar, reportar o guardar datos de actividad del sistema. Es ideal para el análisis histórico porque puede ejecutarse periódicamente (generalmente a través de cron) y guardar los datos en archivos.

1. **Recopilación de Datos con **sar** y **sysstat**:**

- El paquete `sysstat` incluye el demonio `sadc` (System Activity Data Collector) que recopilamos los datos y el script `sa` (System Activity Reporter) que los guarda y gestiona.
- La recopilación automática se configura típicamente con trabajos cron que ejecutan `sadc` y `sa` a intervalos regulares.
- Los archivos de datos históricos se guardan en directorios como `/var/log/sysstat/` o `/var/log/sa/`. El nombre de los archivos suele incluir el día (ej: `sa01`, `sa02...`).

**2. Configuración de `sysstat` (Diferencias Debian vs. Red Hat):** La forma en que se habilita y configura la recopilación automática de datos con `sysstat` varía:

- **Rama Debian/Ubuntu:**
  - Instalar paquete: `sudo apt install sysstat`.
  - Habilitar recopilación: Editar `/etc/default/sysstat` y cambiar `ENABLED="false"` a `ENABLED="true"`.
  - Cron jobs: Los scripts de `sysstat` suelen instalar trabajos cron en `/etc/cron.d/sysstat` que ejecutan la recopilación y limpieza de datos.
  - Archivos de datos: `/var/log/sysstat/`.
- **Rama Red Hat/CentOS/Fedora:**
  - Instalar paquete: `sudo dnf install sysstat` (o `sudo yum install sysstat`).
  - Habilitar recopilación: La recopilación se habilita por defecto al instalar el paquete y habilitar/iniciar el servicio o su temporizador de `systemd`.
  - Cron jobs / Systemd Timers: Los scripts de `sysstat` instalan timers de `systemd` (ej: `sysstat-collect.timer`, `sysstat-summary.timer`) o trabajos cron (en sistemas SysVinit o `crond` tradicional) que ejecutan la recopilación.
  - Archivos de datos: `/var/log/sa/`.
- **Nota:** Después de habilitar la recopilación, puede que necesites esperar un tiempo (ej: un día) para que se generen datos históricos significativos.

**3. Reportar Datos con `sar`:**

- `sar [opciones] [<intervalo> [<conteo>]]`: Muestra datos en tiempo real (si no se especifica un archivo) o desde archivos históricos.
- **Opciones Comunes (para el tipo de reporte):**
  - `-u`: Uso de CPU (promedio de todos los núcleos).
  - `-q`: Carga promedio y longitud de la cola de ejecución.
  - `-r`: Uso de memoria y swap.
  - `-b`: Estadísticas de E/S de bloques (velocidad de transferencia, promedio de solicitudes pendientes).
  - `-d`: Estadísticas de E/S por dispositivo.

- `-n <palabra_clave>`: Estadísticas de red. Palabras clave comunes: DEV (estadísticas por interfaz), EDEV (errores de interfaz), TCP, UDP.
- `-H`: Estadísticas de páginas grandes (Hugepages).
- `-W`: Estadísticas de swapping.
- **Opciones Comunes (para seleccionar el origen de datos/tiempo):**
  - `-f <archivo_datos>`: Lee datos de un archivo específico (ej: `/var/log/sysstat/sa01` para los datos del día 1 del mes actual).
  - `-s HH:MM:SS`: Hora de inicio del reporte.
  - `-e HH:MM:SS`: Hora de fin del reporte.
  - `--iface <interfaz>`: Filtra estadísticas de red para una interfaz específica.
- **Ejemplo:** `sar -u -f /var/log/sysstat/sa05 -s 10:00:00 -e 11:00:00`: Muestra el uso de CPU del archivo `sa05` (del día 5) entre las 10 AM y las 11 AM.

### Análisis para Predicción:

Una vez que tienes datos históricos con `sar`, puedes analizarlos:

- Reportes de `sar -u` a lo largo de semanas/meses: ¿Crece el `%usr` o `%sys` con el tiempo? ¿Cuándo ocurren los picos de uso (`%usr` + `%sys` consistentemente altos)? ¿Es el `load average` alto durante esos picos?
- Reportes de `sar -r` o `sar -W`: ¿Disminuye la memoria libre? ¿Aumenta el uso de swap (`swpd`)? Una actividad alta de swap puede indicar necesidad de más RAM.
- Reportes de `sar -d` o `sar -b`: ¿Aumenta la velocidad de E/S de disco (`rMB/s`, `wMB/s`)? ¿Aumenta el porcentaje de uso del disco (`%util`)? Un `%util` cercano al 100% con `await` alto indica que el disco es un cuello de botella.
- Reportes de `sar -n DEV`: ¿Aumenta el tráfico de red (`rxMB/s`, `txMB/s`)? ¿Aumenta la tasa de paquetes con errores (`rxerr/s`, `txerr/s`)?

Combinando estos datos con el crecimiento conocido (ej: número de usuarios, transacciones, tamaño de base de datos), puedes proyectar cuándo los recursos actuales serán insuficientes.