

Procesamiento Masivo de datos

Alex Di Genova

20/08/2024

Outline

- Big Data
- Sistemas distribuidos
 - Tipos y arquitecturas.
- Linux

Big Data

Visión global



Byte B	10^0	1
Kilobyte	$KB10^3$	1,000
Megabyte	$MB10^6$	1,000,000
Gigabyte	$GB10^9$	1,000,000,000
Terabyte	$TB10^{12}$	1,000,000,000,000
Petabyte	$PB10^{15}$	1,000,000,000,000,000
Exabyte	$EB10^{18}$	1,000,000,000,000,000,000

- Astronomía
- Genómica
- Redes Sociales
- Youtube



Stephens, Zachary D., et al. "Big data: astronomical or genomics?." *PLoS biology* 13.7 (2015): e1002195.



Visión global

Data Phase	Astronomy	Twitter	YouTube	Genomics
Acquisition	25 zetta-bytes/year	0.5–15 billion tweets/year	500–900 million hours/year	1 zetta-bases/year
Storage	1 EB/year	1–17 PB/year	1–2 EB/year	2–40 EB/year
Analysis	In situ data reduction Real-time processing Massive volumes	Topic and sentiment mining Metadata analysis	Limited requirements	Heterogeneous data and analysis Variant calling, ~2 trillion central processing unit (CPU) hours All-pairs genome alignments, ~10,000 trillion CPU hours
Distribution	Dedicated lines from antennae to server (600 TB/s)	Small units of distribution	Major component of modern user's bandwidth (10 MB/s)	Many small (10 MB/s) and fewer massive (10 TB/s) data movement

doi:10.1371/journal.pbio.1002195.t001

- Astronomia
- Genomica
- Redes Sociales
- Youtube



Stephens, Zachary D., et al. "Big data: astronomical or genomic?." *PLoS biology* 13.7 (2015): e1002195.

Visión global



Octubre 2020,
Sebastian
Steudtner
(German)
26 metros

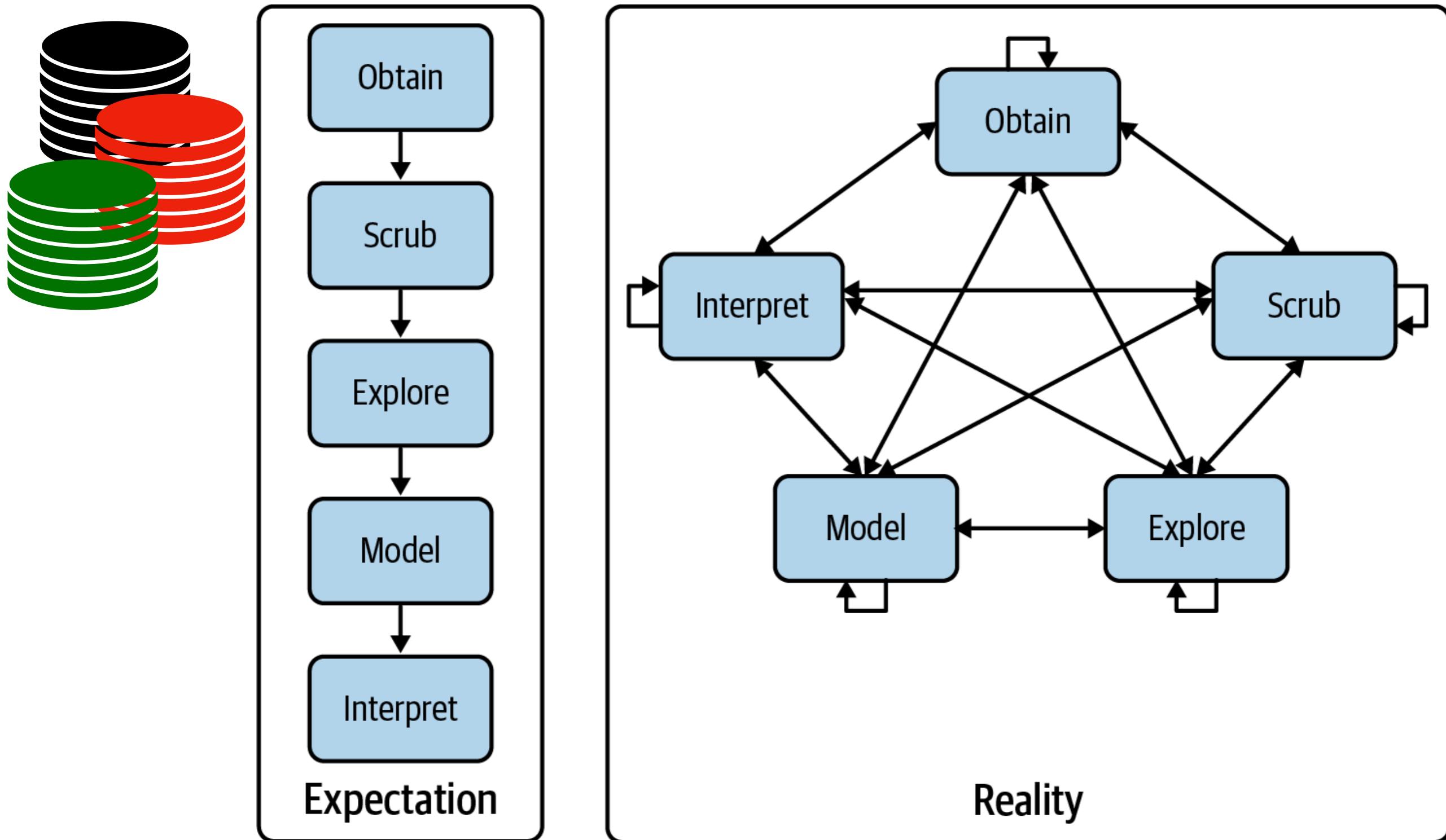
- Astronomía
- Genómica
- Redes Sociales
- Youtube
- Infraestructura (computadores, redes)
- algoritmos (software)

Qué es exactamente el “big data”?

- No es solamente la cantidad.
 - Velocidad (generación)
 - Variedad (tipos)
 - Información (extraer)
 - Datos:
 - Estructurados (SQL)
 - Semi-estructurados (archivos)
 - No estructurados (imágenes, texto, sonidos, video, etc)
- Las 3Vs:
 - Volumen: Son datos demasiado grandes para ser procesados y analizados utilizando métodos tradicionales.
 - Velocidad: Generación y transmisión (tiempo real).
 - Variedad: diversos formatos.
- Desafíos:
 - **Almacenamiento y Procesamiento**
 - **Calidad de los Datos: coherencia de los datos.**
 - **Privacidad y Seguridad**

Data science

Expectation vs Reality



DATA -> INFORMATION

Sistemas Distribuidos

Sistemas distribuidos



Gran cantidad de computadores conectados por una red de alta velocidad.

- Un sistema distribuido es una **colección de computadoras** independientes que aparece ante sus usuarios como un **solo sistema coherente**.

Sistemas distribuidos



Red

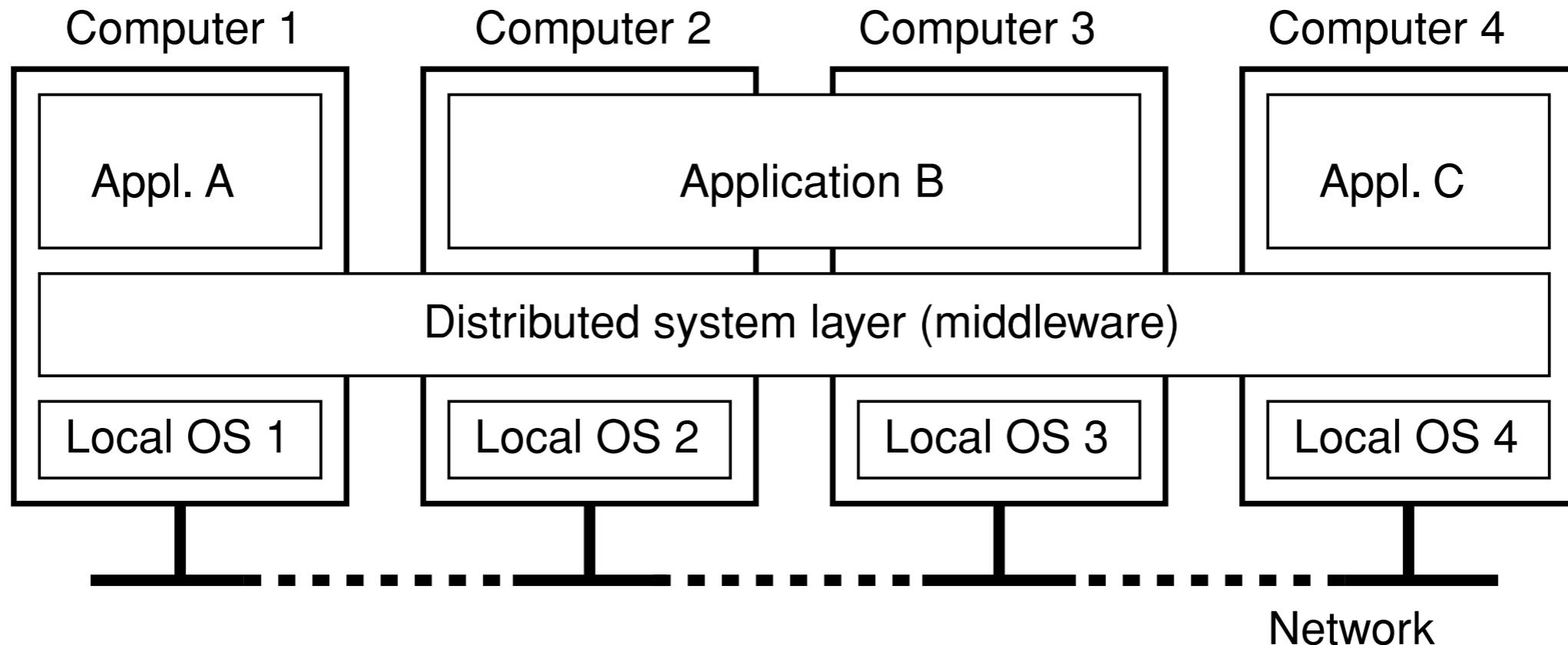
- Un sistema distribuido es una **colección de computadoras** independientes que aparece ante sus usuarios como un **solo sistema coherente**.
 - Las **computadoras necesitan colaborar**.
 - Cómo establecer esta **colaboración** se encuentra en el corazón del desarrollo de sistemas distribuidos.
- Dentro de un sistema distribuido, podrían existir computadoras de alto rendimiento hasta pequeños nodos (**heterogéneo**).
- No se hacen suposiciones sobre la forma en que se **interconectan las computadoras**.

Sistemas distribuidos



- Un sistema distribuido es una **colección de computadoras** independientes que aparece ante sus usuarios como un **solo sistema coherente**.
Red
- **Características**
 - Las diferencias entre las distintas computadoras y las formas en que se comunican están en su mayoría **ocultas** a los usuarios.
 - Los usuarios y las aplicaciones pueden **interactuar** con un sistema distribuido de manera consistente y uniforme
 - Relativamente fácil de **expandir o escalar**.
 - **Alta disponibilidad**.

Sistemas distribuidos

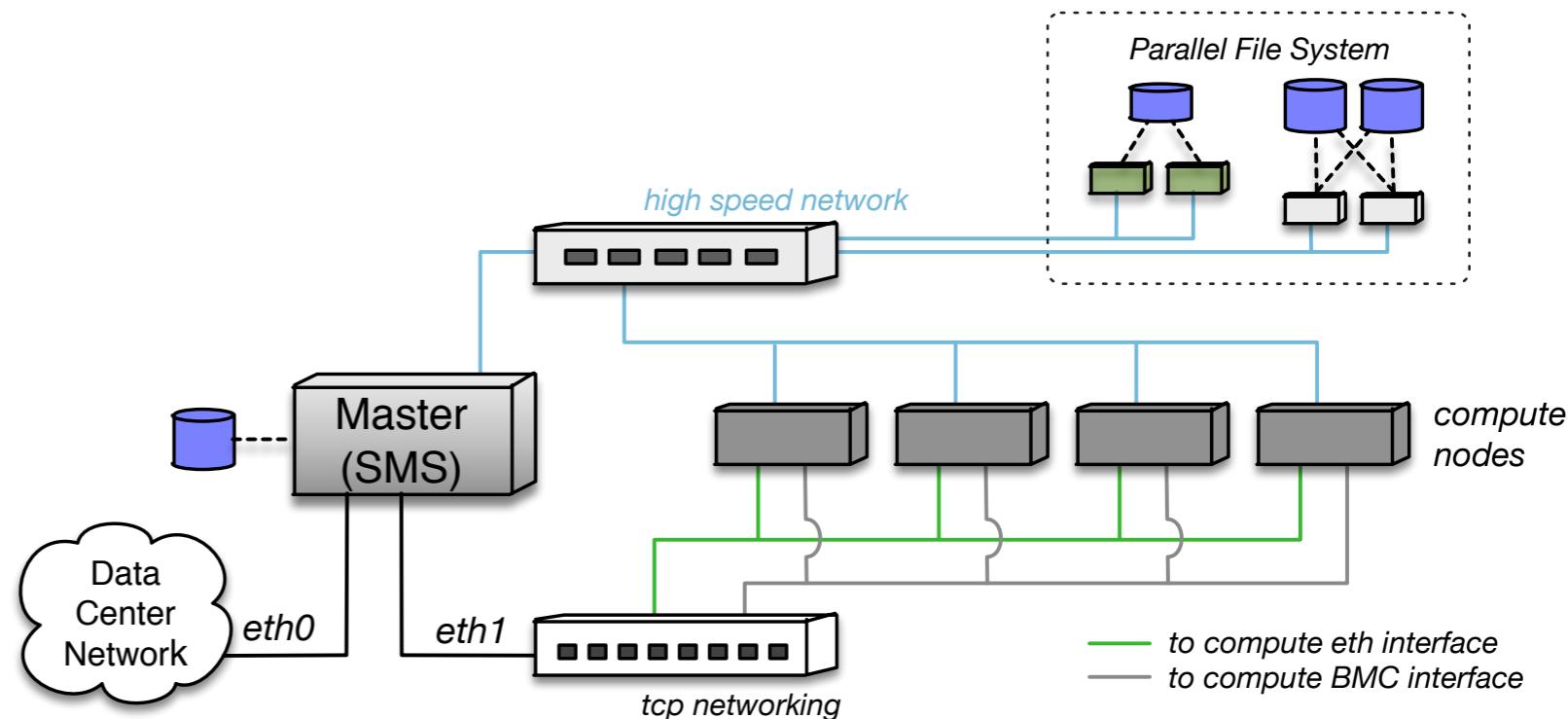


- El middleware (software) se extiende sobre varias máquinas y ofrece a cada aplicación/usuario la misma interfaz para interaccionar con el SD.
- Un sistema distribuido debe hacer que los recursos sean fácilmente accesibles (1); debería ocultar razonablemente el hecho de que los recursos se distribuyen a través de una red(2); debe ser abierto – Interface definition language (3); y debe ser escalable(4).

Sistemas distribuidos

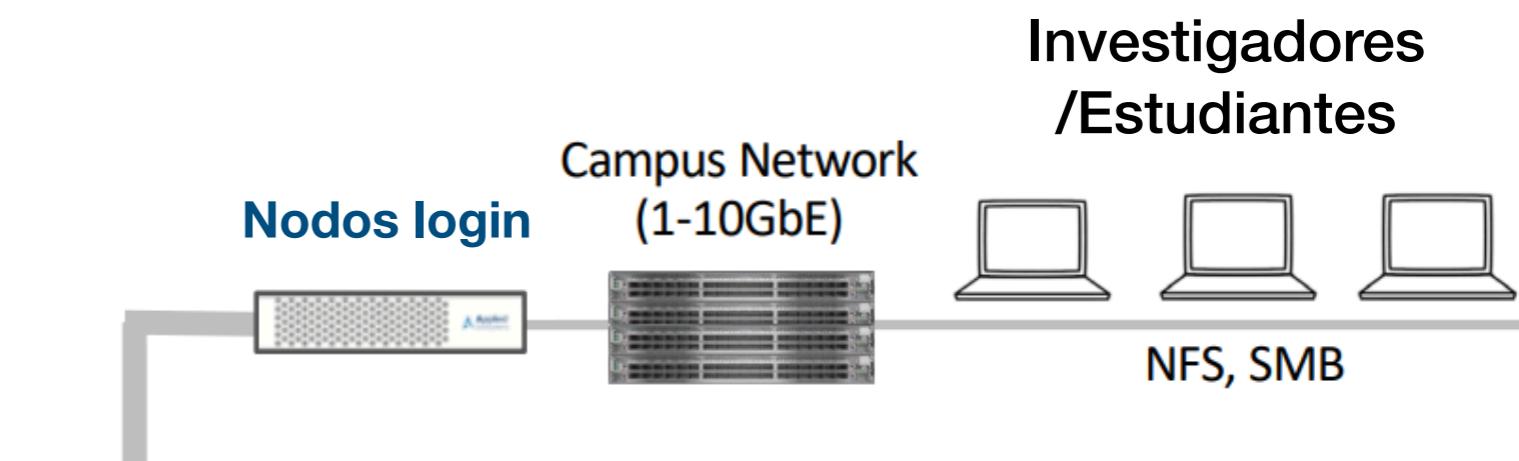
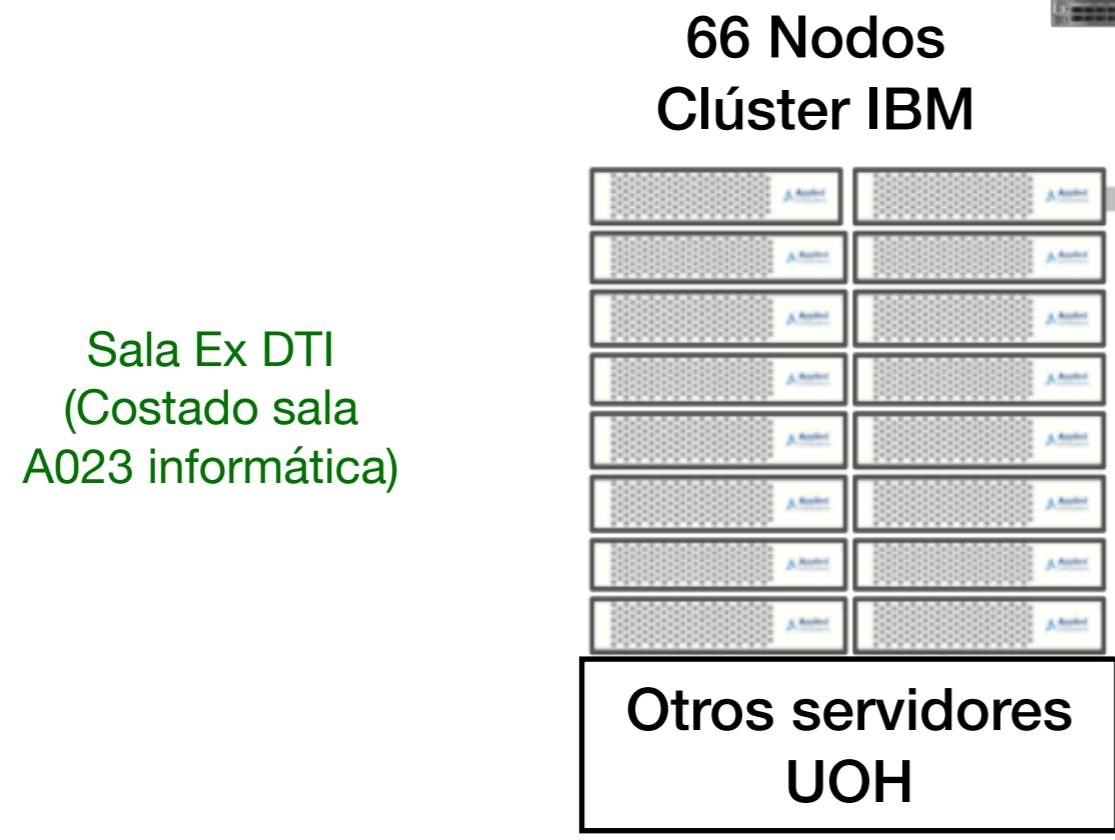
Tipos

- Sistemas de cómputo distribuido
 - Clúster de cómputo
 - Hardware similar
 - Red local de alta velocidad
 - Mismo sistema operativo



The **master** handles the allocation of nodes to a particular parallel program, maintains a batch queue of submitted jobs, and provides an interface for the users of the system.

Arquitectura HPC-UOH



Red Infiniband
40 G/s

Servidor de Metadata

Servidores storage

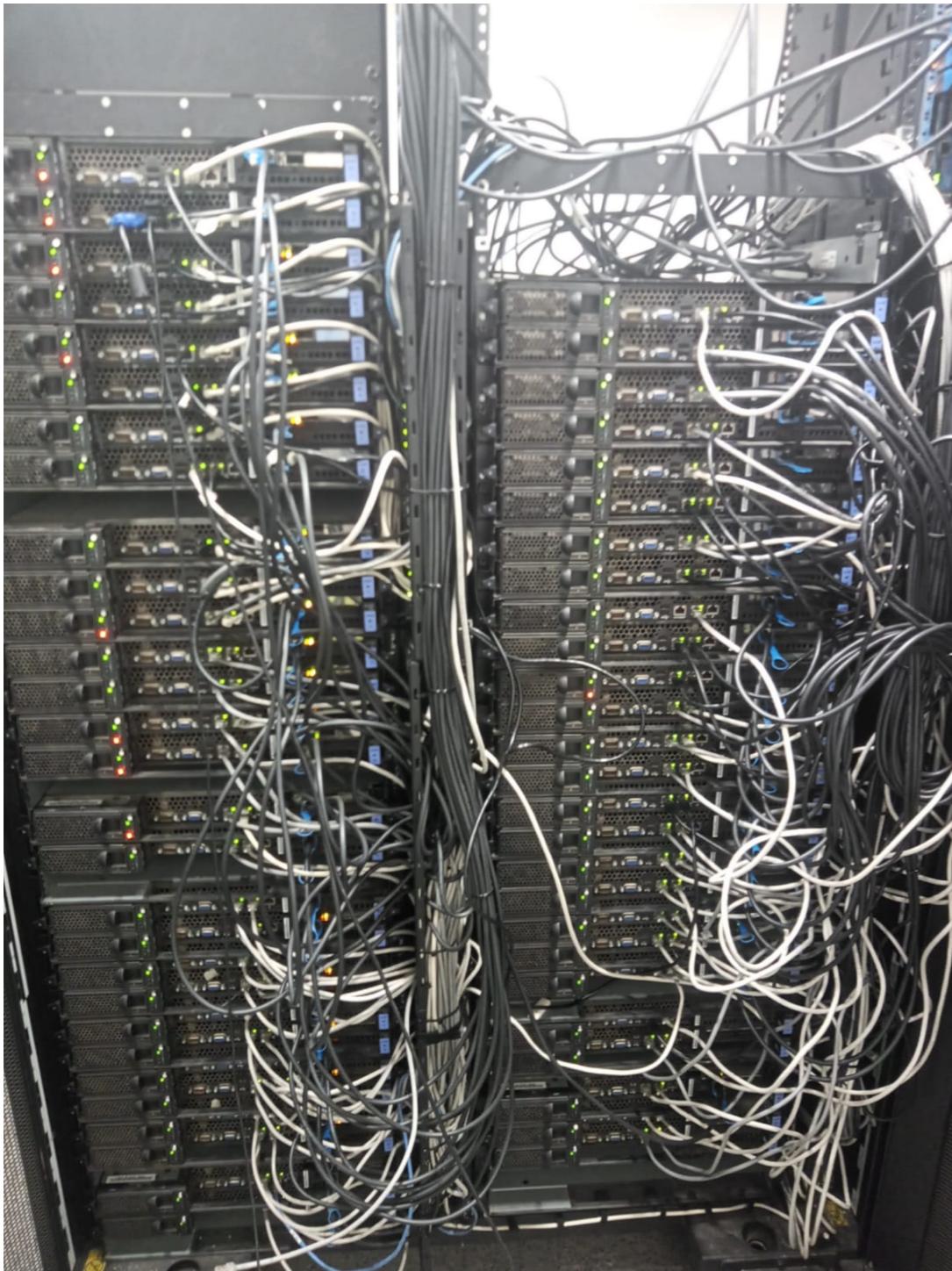
Storage
(196 Tb RAW)



- Mejoras Hardware
 - RAM: 128Gb
 - Disco : 196 Tb
 - Servidores dedicados
 - 1TB RAM
 - 5000 GPUs

HPC-UOH

Kütral



- Kütral hardware:
 - A total of 1792 cores (2.6 GHz)
 - 1 PB of storage
 - 13 TB of RAM
 - InfiniBand network of 40 G/s.

HPC-UOH

Kütral software

Acceso a los Servicios HPC Kutral



MediaWiki



Nextcloud



Open OnDemand



Ticket Kutral



Ganglia



<https://kutral-hpc.uoh.cl/>

Kütral

A user friendly HPC

```
jcanan@zenbook:~
```

```
[jcanan@zenbook ~]$ █
```

```
l q   a K .  θ 1 y   + F   M { n   I   e x   | 9   a P
r   k \ V   o +   θ A   c / ;   0   x )   3 p   3 m #
T   C 5 1   L c   i Q f   R M n   k   K 1   t l   8 + x   x
P   h   u   B V   5 a |   0 A p   ,   d 2   ] } a a $ F   3 -
C l w   |   l s   d 4 T   Q ] I   g   P c   r   Z   j &   S F
l U `   F K   U   k X   h = s   :   ; i   L   { s   5 !
B 8 j   P   d   S X o   k ^   # * b K   [   I )   } o   h {
V w #   u   H   v G v   J   & < 4   \   a r   I ]   U $
8 C 0   E m   % f   V   x [   d   F   t :   $   {   ' V   Q 4
^ ^ |   N ?   b   U   5 @   r   v R
# / p   P Q   n   7   z 4 U   8   K   b J P
i - %   ? e   < r h   9 C &   L   f   r :
0 k 5   x   7 v e   G > c   7   N G   E c 2 =
9 o   4   ! B s   3 ? M   F   f X   A = 0 s
!   !   4 7 g   P 4 s   {   9 ^   c $ =
j   j   G Z   M K i   3 ; 6   n   ? <   _ ;
t   t   + y   * . ^   Y K   |   c   6 g   j a E
s   s   a s   3 0 -   j 2 9   6 5   > X   H   n y

```



HPC-UOH

Kütral software

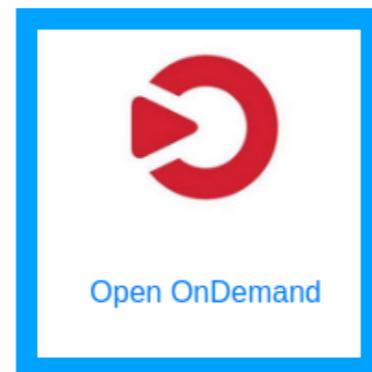
Acceso a los Servicios HPC Kutral



MediaWiki



Nextcloud



Open OnDemand



Ticket Kutral



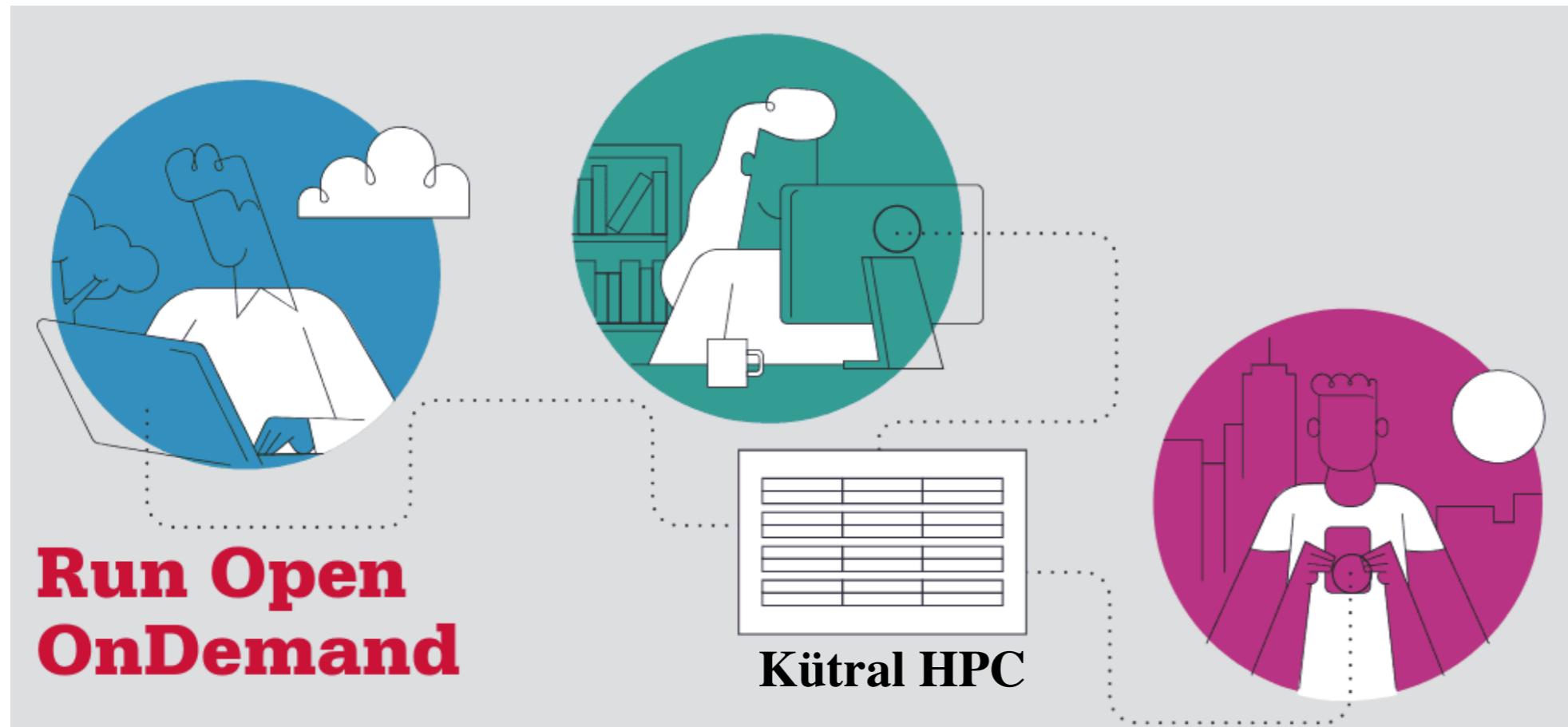
Ganglia



<https://kutral-hpc.uoh.cl/>

Kütral a user friendly HPC

Open Ondemand



- Access Kütral through the web to compute from anywhere, on any device.
 - Zero installation
 - Easy to use
 - Compatible with any device

- One Click to:
 - Files
 - Jupyter
 - Rstudio
 - IGV
 - Cytoscape
 - ...

Kütral a user friendly HPC

Open Ondemand



Acceso a los Servicios HPC Kutral



MediaWiki



Nextcloud



[Open OnDemand](#)

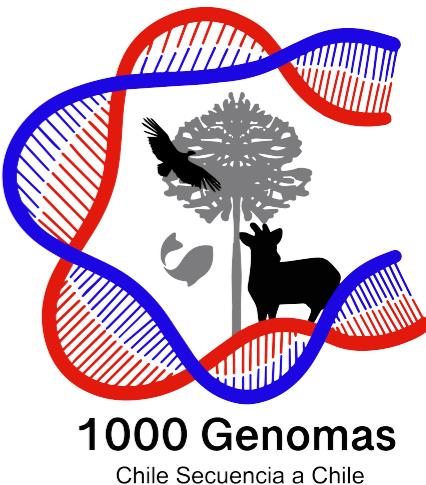


Tic



HPC-UOH

Kütral software



Acceso a los Servicios HPC Kutral



MediaWiki



Nextcloud



Open OnDemand



Ticket Kutral



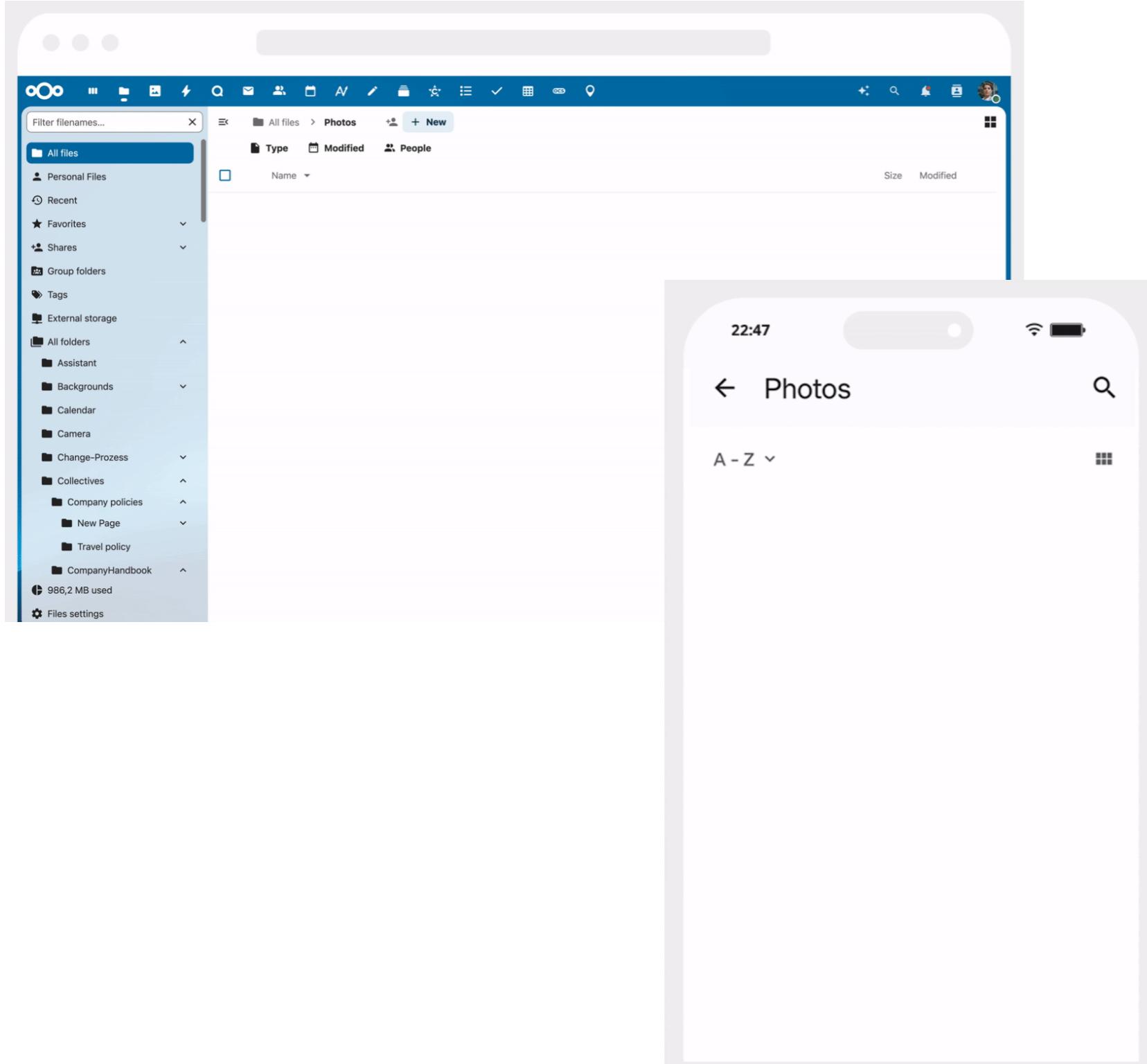
Ganglia



<https://kutral-hpc.uoh.cl/>

Kütral a user friendly HPC

File sharing and collaboration



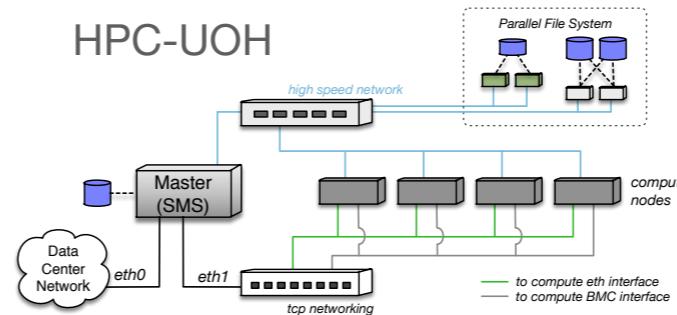
- 1000GCh Cloud
- TALK
- OnlyOffice
- Mail/Calendar
- draw.io
- File sync

Sistemas distribuidos

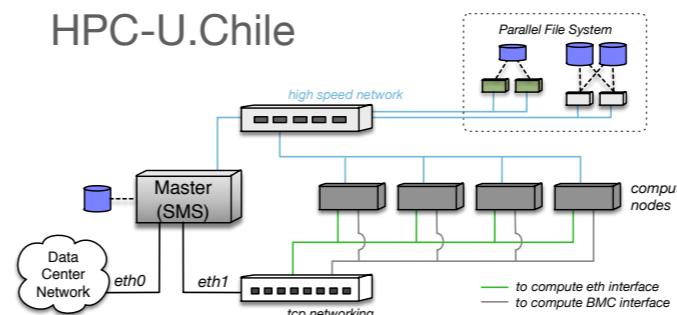
Tipos

- Sistemas de cómputo distribuido
 - Clúster de cómputo
 - Hardware similar
 - Red local de alta velocidad
 - Mismo sistema operativo
 - Grid de computo
 - Alto grado de heterogeneidad
 - Federación de sistemas de cómputo (clúster)

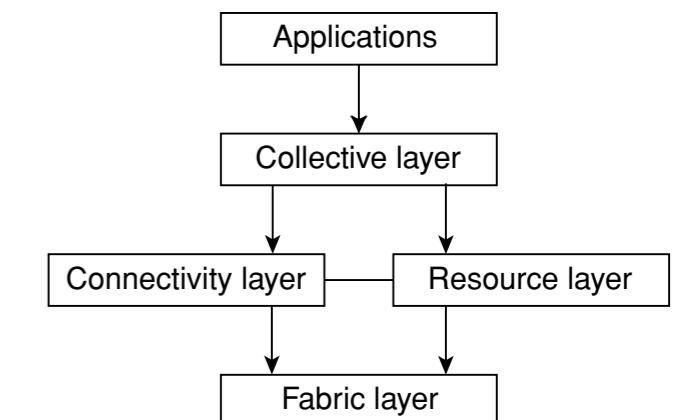
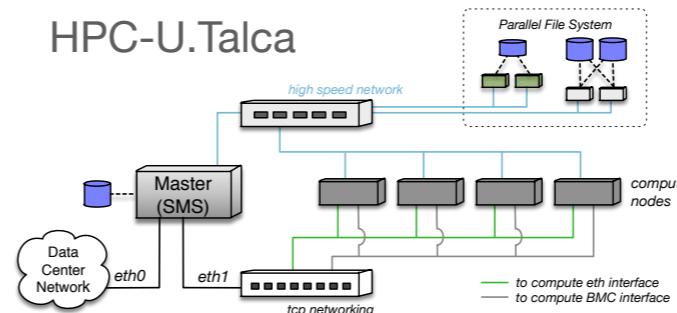
Universidades (organización virtual)



HPC-U.Chile



HPC-U.Talca



- Fabric layer:
 - proporciona interfaces a los recursos locales en un sitio específico
- Connectivity layer
 - protocolos de comunicación para soportar transacciones de red que abarcan el uso de múltiples recursos.
- Resource layer
 - responsable de administrar un solo recurso
- Collective layer:
 - maneja el acceso a múltiples recursos y generalmente consiste en servicios para el descubrimiento de recursos, asignación y programación de tareas

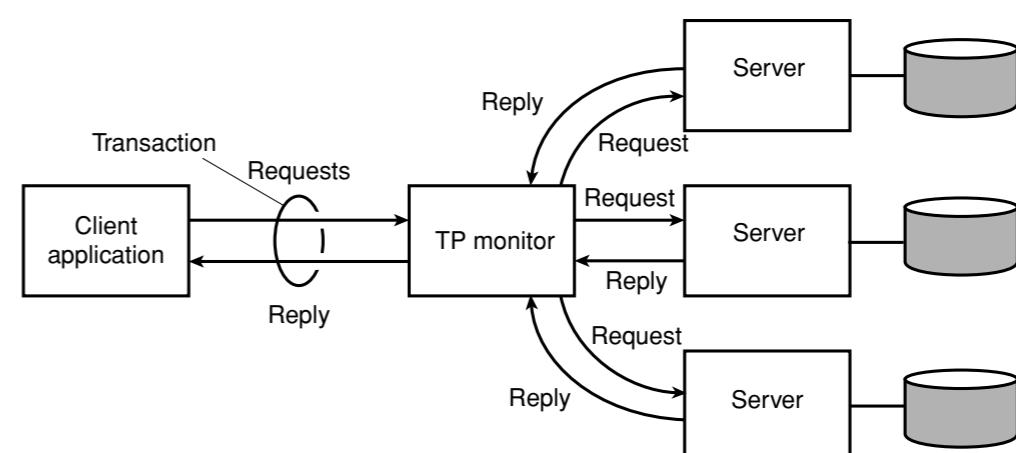
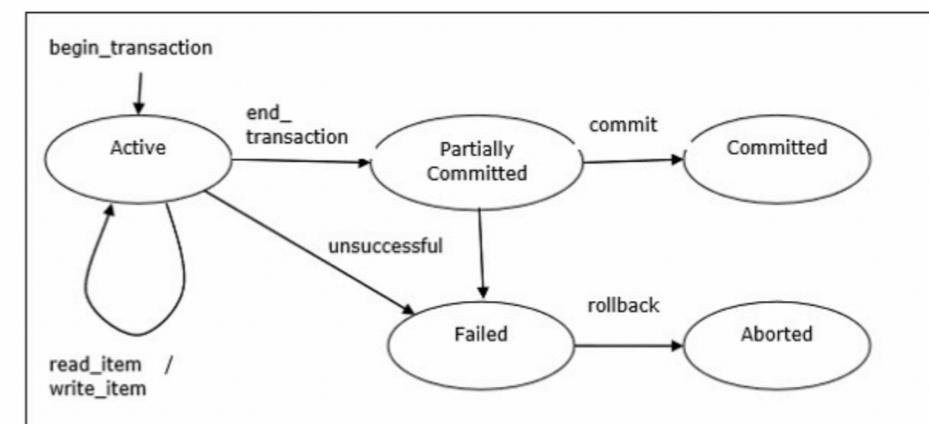
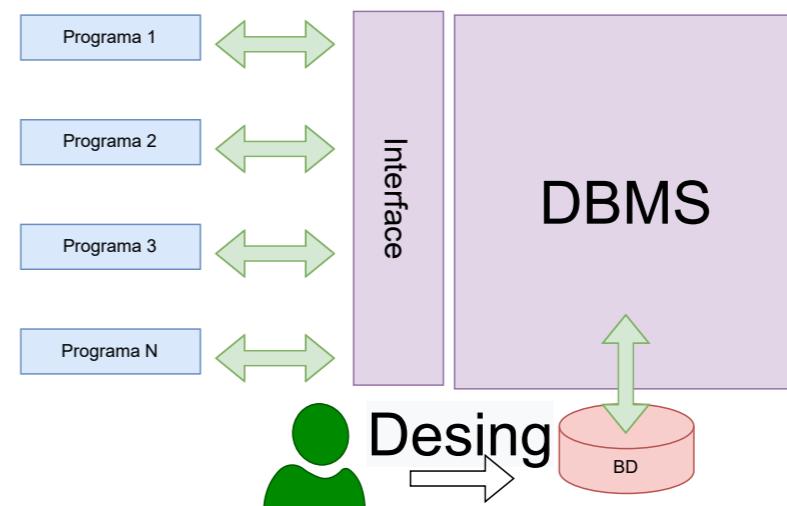


HTCondor
Software Suite

Sistemas distribuidos

Tipos

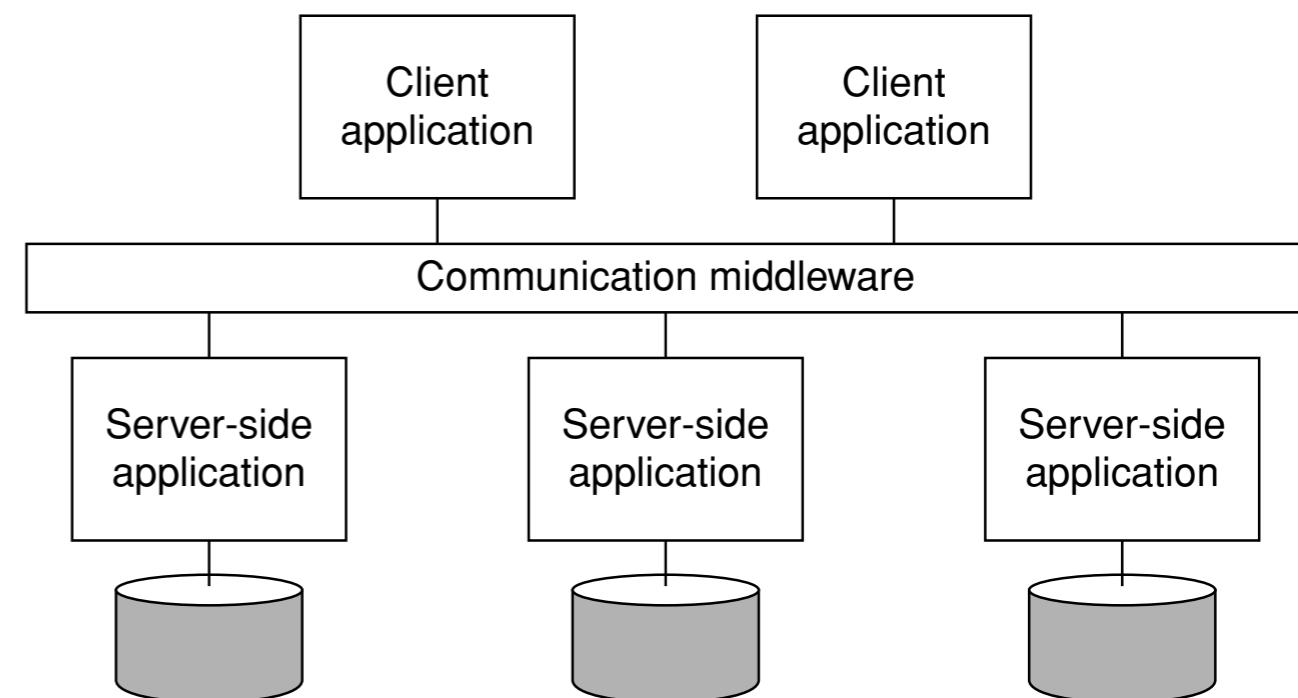
- Sistemas de información distribuido
 - un servidor que ejecuta una aplicación (que a menudo incluye una base de datos) y la coloca a disposición de programas remotos, usuarios o clientes.
 - Sistemas de procesamiento de transacciones
 - La idea clave es que se ejecutarán todas o ninguna de las solicitudes.
 - Propiedades ACID (Atomic, Consistent, Isolated, Durable)
 - Monitor de procesamiento de transacciones
 - Permite que una aplicación acceda a múltiples servidores/bases de datos mediante un modelo de programación transaccional.



Sistemas distribuidos

Tipos

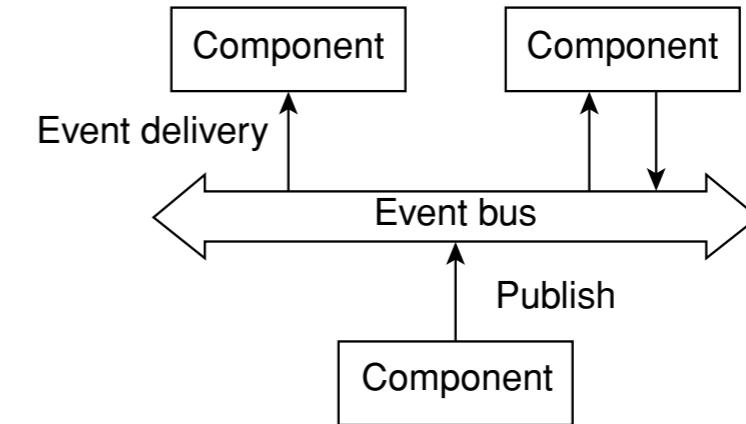
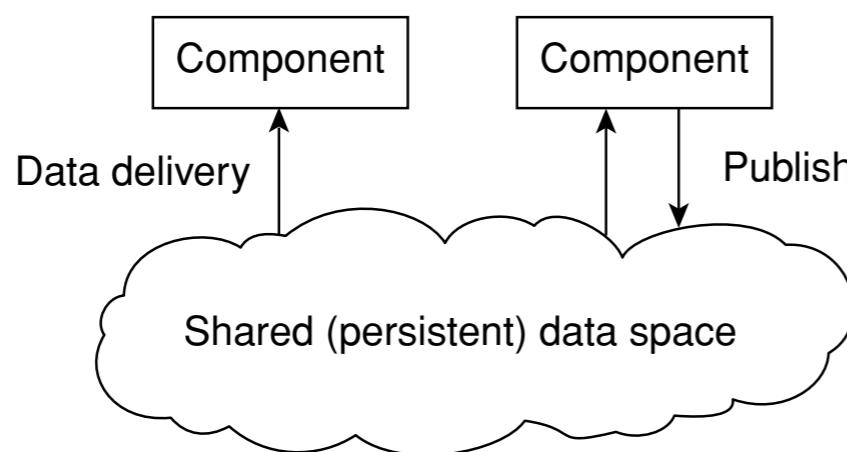
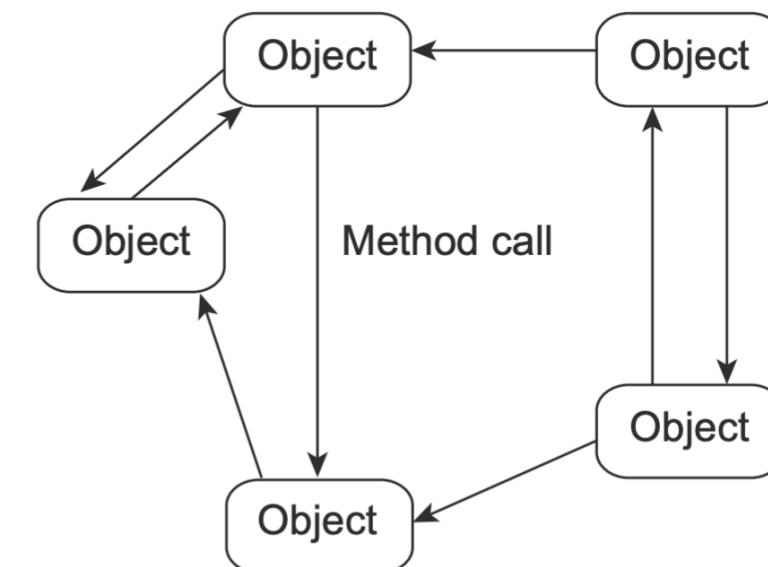
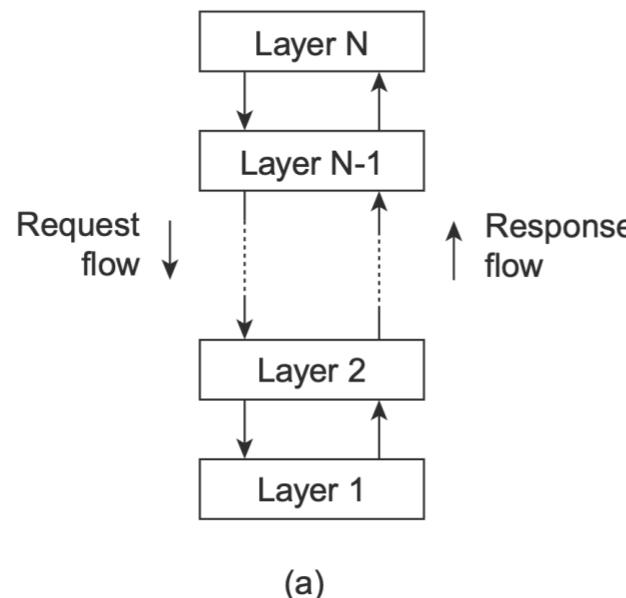
- Sistemas de información distribuido
 - Sistemas de procesamiento de transacciones
 - Sistemas de integración de aplicaciones.
 - Comunicación entre aplicaciones.
 - Idea: aplicaciones intercambian directamente información.
 - Métodos de llamadas remota (Resource Manager Interface)



Sistemas distribuidos

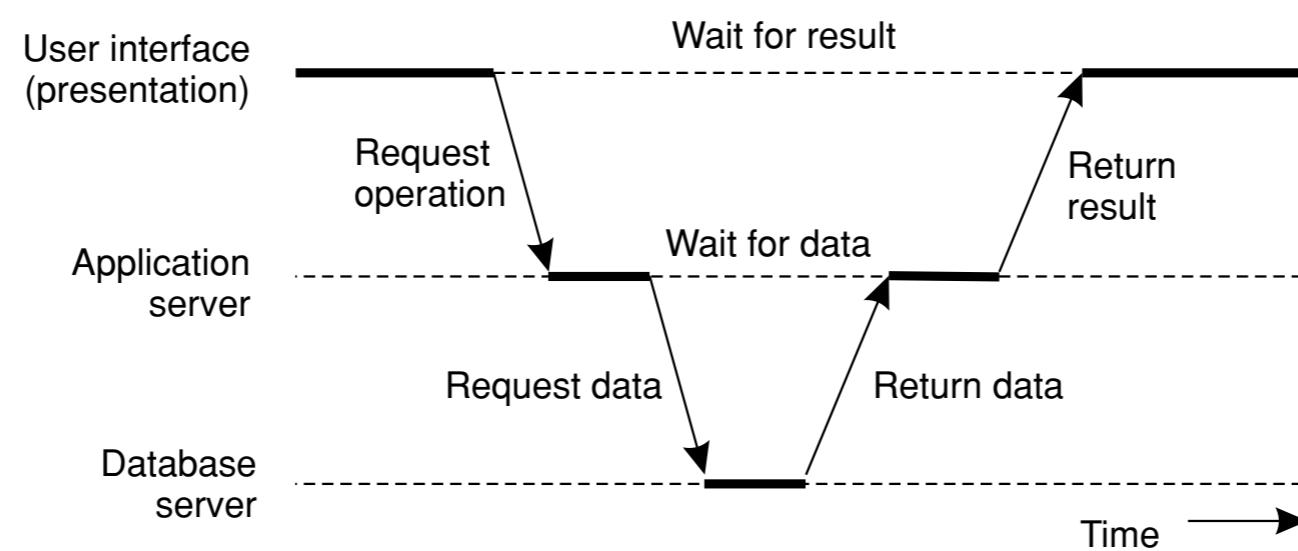
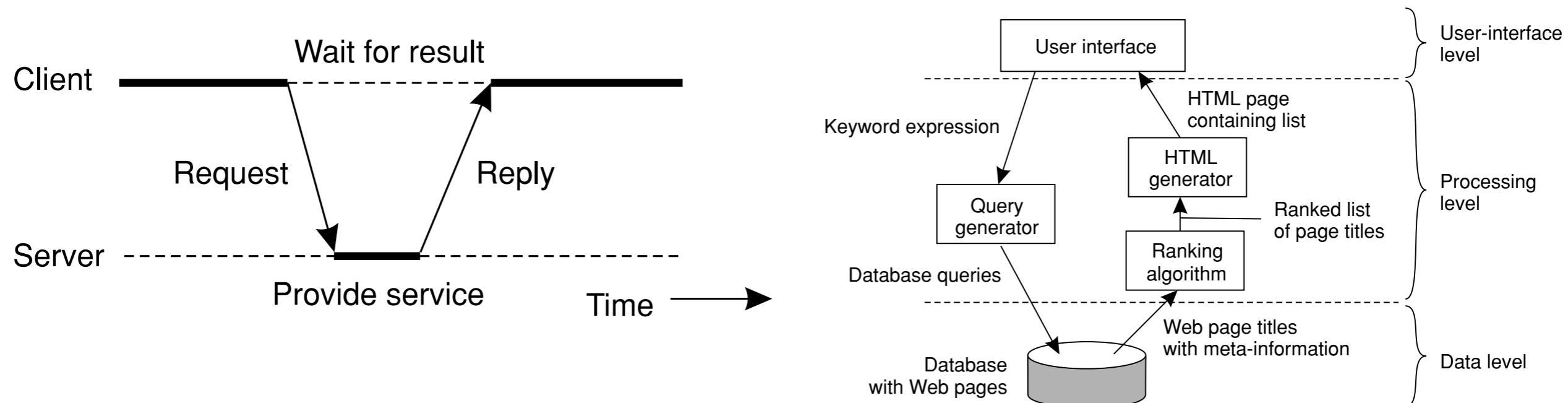
Arquitecturas

- Necesitamos definir cómo deben organizarse los diversos componentes de software y cómo deben interactuar.



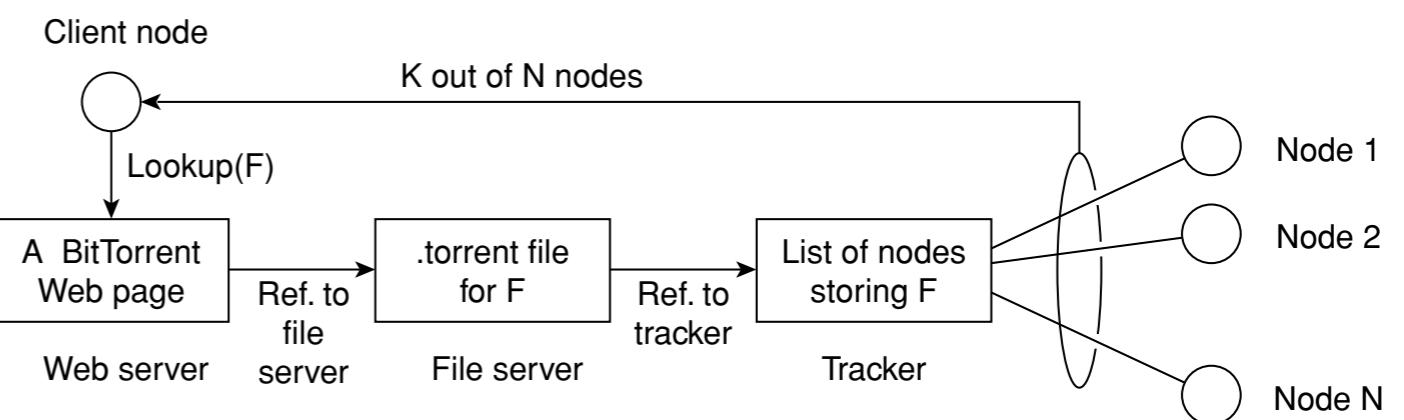
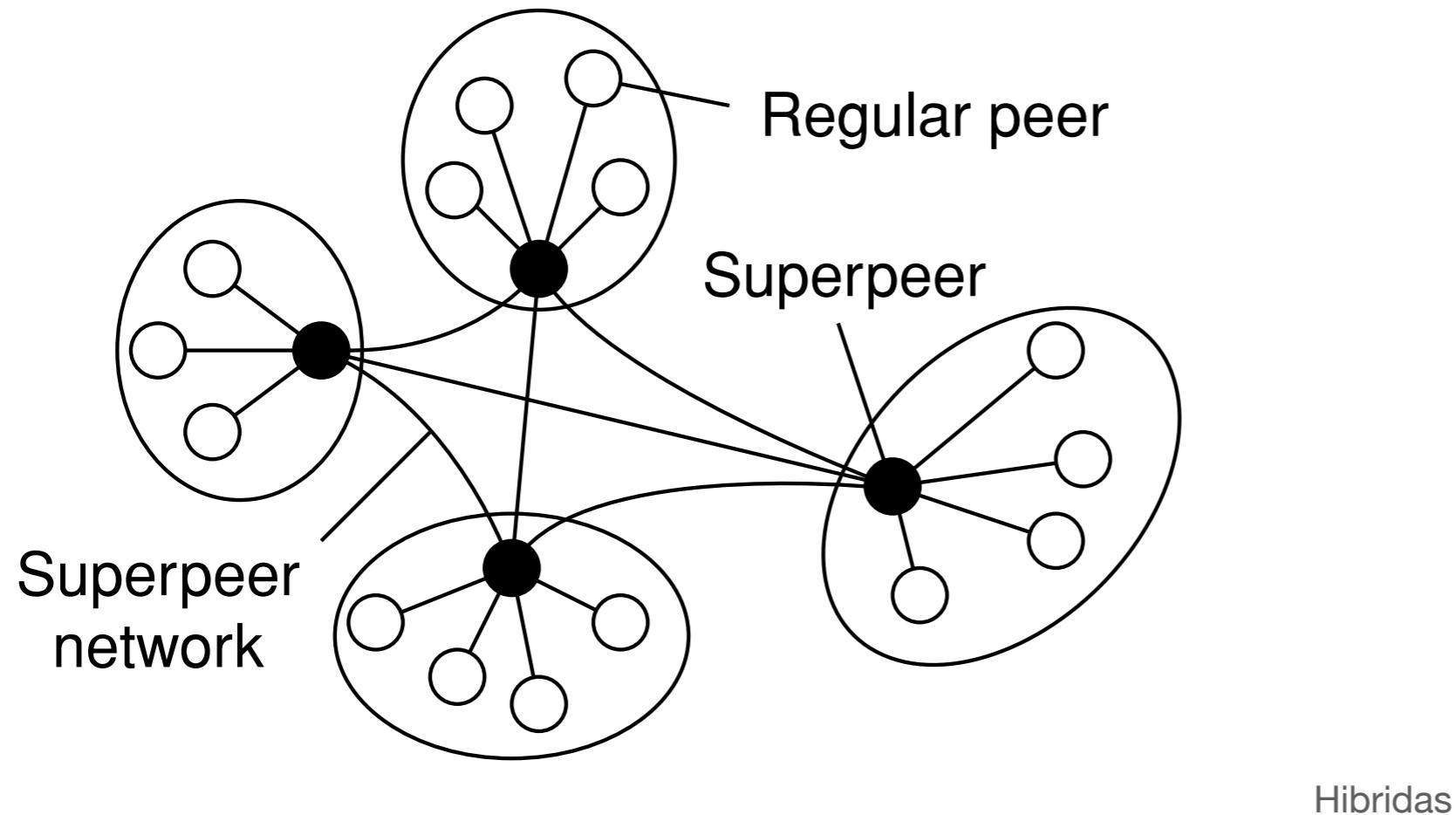
Sistemas distribuidos

Arquitecturas centralizadas



Sistemas distribuidos

Arquitecturas no-centralizadas



Linux

Linux

A operating system

User Processes

Graphical User Interface Servers Shell

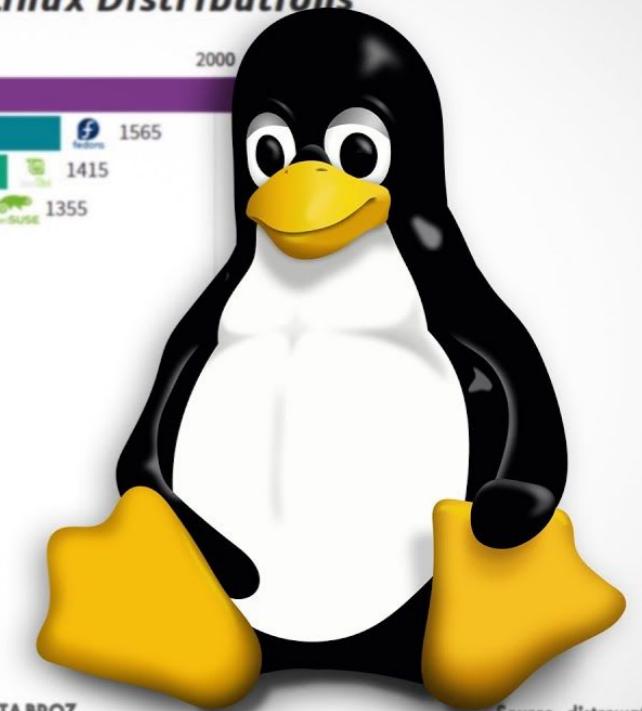
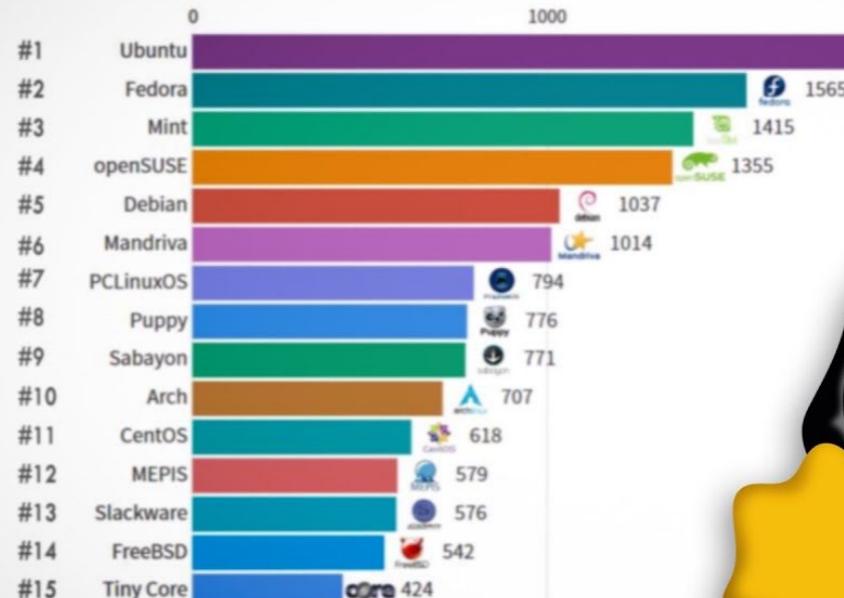
Linux Kernel

System Calls Process Management Memory Management
Device Drivers

Hardware

Processor (CPU) Main Memory (RAM) Disks Network Ports

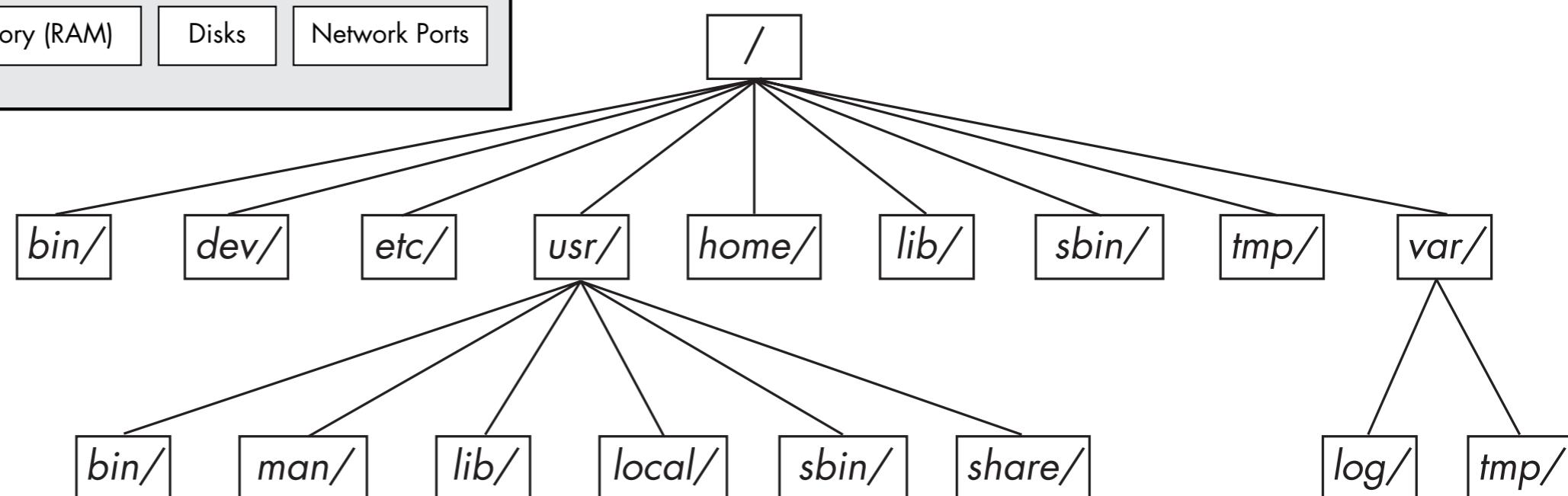
Most Popular Linux Distributions



Values: Hits Per Day in Distrowatch

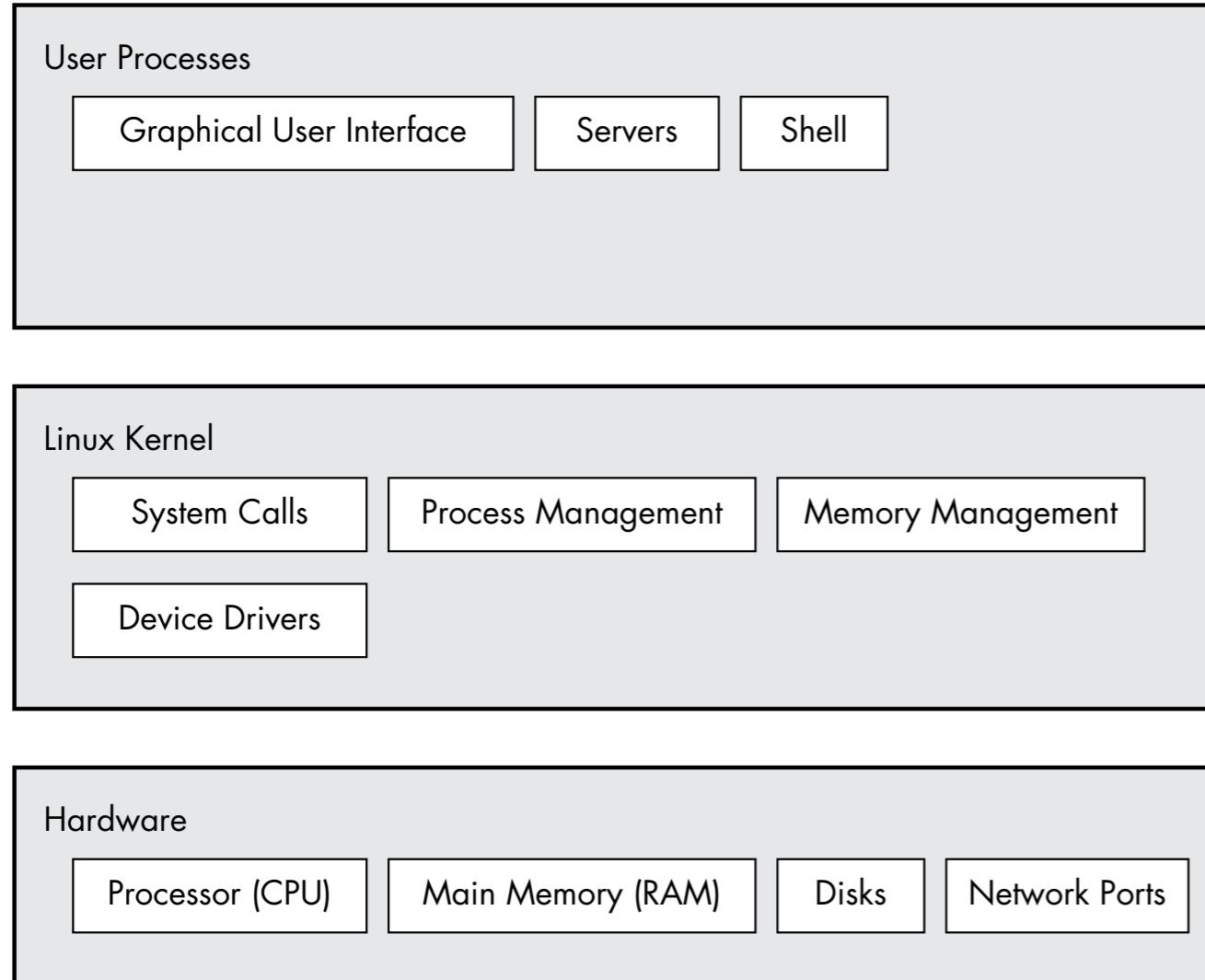
DATA BROZ

Source: distrowatch



Linux

A operating system



- The *shell* is a program that runs commands.
 - The *shell* also serves as a small programming environment.

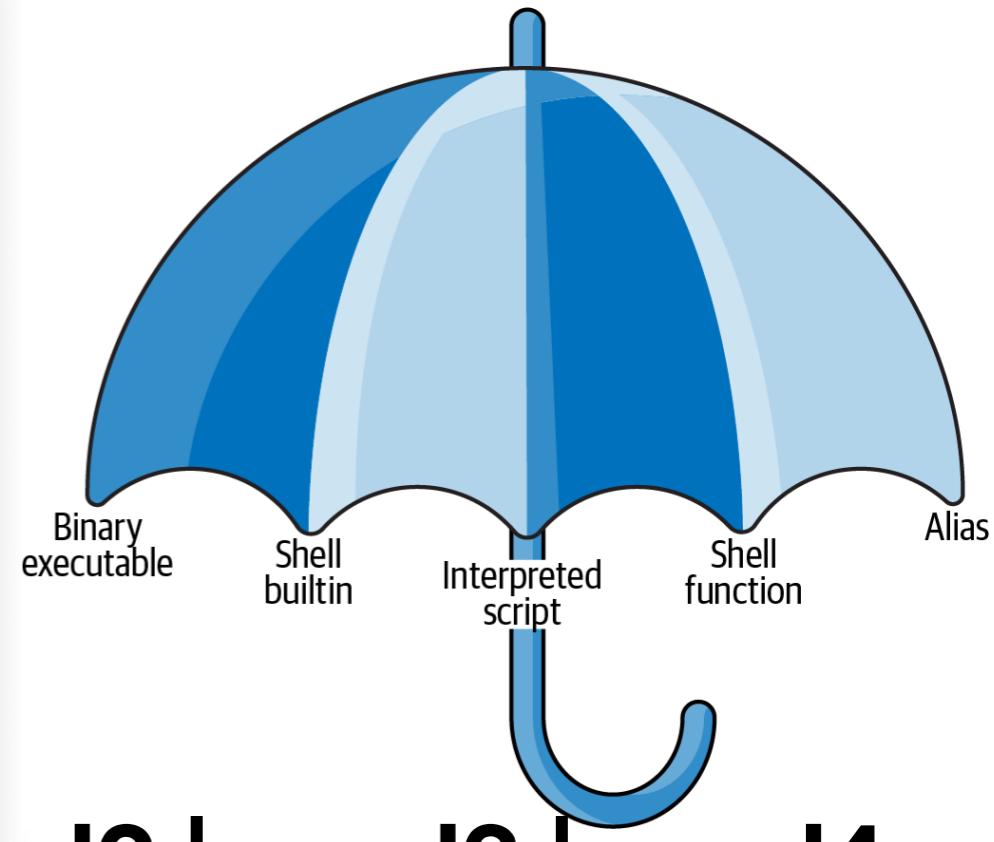


Linux

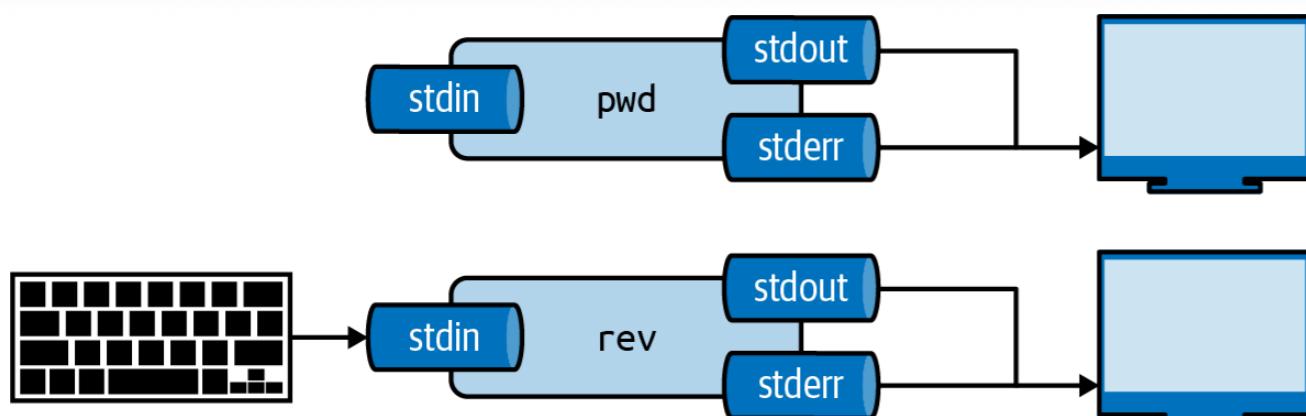
Shell command line

```
clases - adigenova@host04:~ - ssh adigenova@172.16.105.104 - 80x23
[adigenova@host04 ~]$ echo "The command line is the force DCBI" | cowsay -f tux
< The command line is the force DCBI >
-----
\ \
 .--.
|o_o |
|:_-/
| / \ |
( | ) |
\_\_)=\_\_/
[adigenova@host04 ~]$
```

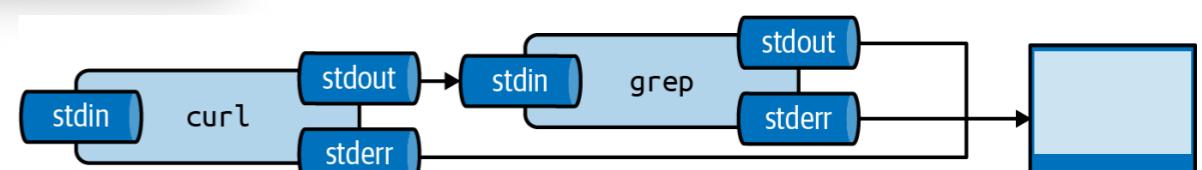
Command-line tool



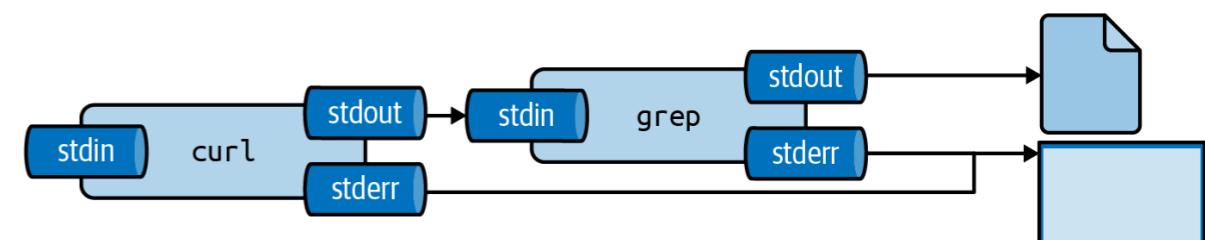
cmd1 | cmd2 | cmd3 | cmd4 ...



Every tool has three standard streams: standard input (stdin), standard output (stdout), and standard error (stderr)



The output from a tool can be piped to another tool



The output from a tool can be redirected to a file

Linux

Basic shell commands

Command Information

```
man chmod          # Display page manual of a command  
man -fl--whatis chmod      # Display short description about a command  
man -kl--apropos permission # Display all related commands from a specific keyword  
  
chmod --help        # Display usage options of a command
```

Command History

```
history           # View all previous commands  
history | grep foo # View the commands using a specific word  
history | head -nl--lines 3 # View the first 3 executed commands  
history 3         # View the last 3 executed commands  
history -d 99     # Clear a command from a specific line  
history -c        # Clears all history commands  
!!                # Run the last command executed
```

Creating Directories

```
mkdir foo          # Create a directory  
mkdir foo bar      # Create multiple directories  
mkdir -pl-parents foo/bar # Create nested directory  
mkdir -pl-parents {foo,bar}/baz # Create multiple nested directories
```

Deleting Directories

```
rmdir foo          # Delete non-empty directory  
rm -rl--recursive foo      # Delete directory including contents  
rm -rl--recursive -fl--force foo # Delete directory including contents, ignore nonexistent files and never prompt
```

Navigating Directories

```
pwd               # Print current directory path  
ls                # List directories  
ls -al--all       # List directories including hidden  
ls -l             # List directories in long form  
ls -t             # List directories by modification time, newest first  
stat foo.txt      # List size, created and modified timestamps for a file  
stat foo          # List size, created and modified timestamps for a directory  
tree              # List directory and file tree  
tree -a           # List directory and file tree including hidden  
tree -d           # List directory tree  
  
cd foo            # Go to foo sub-directory  
cd                # Go to home directory  
cd ~              # Go to home directory  
cd -              # Go to the previously chosen directory  
pushd foo          # Go to foo sub-directory and add previous directory to stack  
popd              # Go back to directory in stack saved by `pushd`
```

Moving Directories

```
cp -R l --recursive foo bar      # Copy directory  
mv foo bar                      # Move directory  
  
rsync -zl--compress -vl--verbose /foo /bar # Copy directory, overwrites destination  
rsync -avz /foo username@hostname:/bar # Copy local directory to remote directory  
rsync -avz username@hostname:/foo /bar # Copy remote directory to local directory
```

Linux

Basic shell commands

Creating Files

```
touch foo.txt      # Create file or update existing files modified timestamp  
touch foo.txt bar.txt # Create multiple files  
touch {foo,bar}.txt # Create multiple files  
touch test{1..3}   # Create test1, test2 and test3 files  
touch test{a..c}   # Create testa, testb and testc files  
  
mktemp           # Create a temporary file
```

Moving Files

```
cp foo.txt bar.txt          # Copy file  
mv foo.txt bar.txt          # Move file  
  
rsync -zl--compress -v /foo.txt /bar    # Copy file quickly if not changed  
rsync -zl--compress -v /foo.txt /bar.txt # Copy and rename file quickly if not changed
```

Deleting Files

```
rm foo.txt            # Delete file  
rm -fI--force foo.txt # Delete file, ignore nonexistent files and never prompt
```

Standard Output, Standard Error and Standard Input

```
echo "foo" > bar.txt      # Overwrite file with content  
echo "foo" >> bar.txt     # Append to file with content  
  
ls exists 1> stdout.txt   # Redirect the standard output to a file  
ls noexist 2> stderr.txt   # Redirect the standard error output to a file  
ls > out.txt 2>&1         # Redirect standard output and error to a file  
ls > /dev/null             # Discard standard output and error  
  
read foo                 # Read from standard input and write to the variable foo
```

Reading Files

```
cat foo.txt            # Print all contents  
less foo.txt           # Print some contents at a time (g - go to top of file)  
head foo.txt           # Print top 10 lines of file  
tail foo.txt           # Print bottom 10 lines of file  
tail -fl--follow foo.txt # Print bottom 10 lines of file updating with new data  
open foo.txt           # Open file in the default editor  
wc foo.txt             # List number of lines words and characters in the file
```

Sorting Files

```
sort foo.txt          # Sort file (ascending order)  
sort -rl--reverse foo.txt # Sort file (descending order)  
sort -nl--numeric-sort foo.txt # Sort numbers instead of strings  
sort -tl--field-separator: -k 3n /foo/foo.txt # Sort by the third column of a file
```

- <https://github.com/trinib/Linux-Bash-Commands>

Linux

Basic shell commands

Finding Files

Find binary files for a command.

```
type -a wget          # Display all locations of executable  
which -a wget         # Display all locations of executables  
whereis wget          # Find the binary, source, and manual page files
```

locate uses an index and is fast.

```
updatedb              # Update the index  
  
locate foo.txt        # Find a file  
locate --ignore-case  # Find a file and ignore case  
locate f*.txt         # Find a text file starting with 'f'
```

find doesn't use an index and is slow.

```
find /path -name foo.txt      # Find a file  
find /path -iname foo.txt    # Find a file with case insensitive matching  
find /path -name "*.*"       # Find all text files  
find /path -name foo.txt -delete # Find a file and delete it  
find /path -type f -name foo.txt # Find a file  
find /path -type d -name foo  # Find a directory  
find /path -type l -name foo.txt # Find a symbolic link  
find /path -type f -mtime +30   # Find files that haven't been modified in 30 days  
find /path -type f -mtime +30 -delete # Delete files that haven't been modified in 30 days
```

Find in Files

```
grep 'foo' /bar.txt          # Search for 'foo' in file 'bar.txt'  
grep 'foo' /bar -rl--recursive # Search for 'foo' in directory 'bar'  
grep 'foo' /bar -ll--files-with-matches # Show only files that match  
grep 'foo' /bar -LI--files-without-match # Show only files that don't match  
grep 'Foo' /bar -il--ignore-case # Case insensitive search  
grep 'foo' /bar -xl--line-regexp # Match the entire line  
grep 'foo' /bar -vl--invert-match # Show only lines that don't match  
grep 'foo' /bar -cl--count    # Count the number lines that match  
grep 'foo' /bar -nl--line-number # Add line numbers  
grep 'foo' /bar --colour     # Add colour to output  
grep 'foo\lbar' /baz -R      # Search for 'foo' or 'bar' in directory 'baz'
```

Replace in Files

```
sed 's/fox/bear/g' foo.txt      # Replace fox with bear in foo.txt and output to console  
sed 's/fox/bear/gi' foo.txt     # Replace fox (case insensitive) with bear in foo.txt  
  
sed 's/red fox/blue bear/g' foo.txt # Replace fox with bear in foo.txt and save in bar.txt  
sed 's/fox/bear/g' foo.txt > bar.txt # Replace fox with bear and overwrite foo.txt  
sed -il--in-place 's/fox/bear/g' foo.txt # Replace the 10th line of the file  
sed -il--in-place '10s/find/replace/' foo.txt # Replace in the file 10-20 lines  
sed -il--in-place '10,20s/find/replace/' foo.txt
```

Symbolic Links

```
ln -sl--symbolic foo bar      # Create a link 'bar' to the 'foo' folder  
ls -l                         # Show where symbolic links are pointing
```

Linux

Basic shell commands

Disk Usage

```
df           # List disks, size, used and available space  
df -hl--human-readable # human readable format
```

```
du           # List current directory, subdirectories and file sizes  
du /foo/bar # List specified directory, subdirectories and file sizes  
du -dl--max-depth # List current directory, subdirectories and file sizes within the max depth  
du -d 0     # List current directory size
```

Memory Usage

```
free         # Show memory usage  
free -hl--human # Show human readable memory usage  
free -hl--human --si # Show human readable memory usage in power of 1000 instead  
free -sl--seconds 5 # Show memory usage and update continuously every five seconds
```

Shutdown and Reboot

```
shutdown      # Shutdown in 1 minute  
shutdown now  # Immediately shut down  
shutdown +5   # Shutdown in 5 minutes  
  
shutdown -rl--reboot # Reboot in 1 minute  
shutdown -rl--reboot now # Immediately reboot  
shutdown -rl--reboot +5 # Reboot in 5 minutes  
shutdown -c    # Cancel a shutdown or reboot  
  
reboot       # Reboot now  
reboot -f    # Force a reboot
```

Identifying Processes

```
top          # List all processes interactively  
htop         # List all processes interactively  
ps ax       # List all processes  
pidof foo   # Return the PID of all foo processes  
  
CTRL+Z      # Suspend a process running in the foreground  
bg           # Resume a suspended process and run in the background  
fg           # Bring the last background process to the foreground  
fg 1         # Bring the background process with the PID to the foreground  
  
sleep 30    # Sleep for 30 seconds and move the process into the background  
jobs        # List all background jobs  
jobs -p     # List all background jobs with their PID  
  
lsof        # List all open files and the process using them  
lsof -itcp:4000 # Return the process listening on port 4000
```

Killing Processes

```
CTRL+C      # Kill a process running in the foreground  
kill PID    # Shut down process by PID gracefully. Sends TERM signal.  
kill -9 PID # Force shut down of process by PID. Sends SIGKILL signal.  
pkill foo   # Shut down process by name gracefully. Sends TERM signal.  
pkill -9 foo # force shut down process by name. Sends SIGKILL signal.  
killall foo # Kill all process with the specified name gracefully.
```

HTTP Requests

```
wget https://example.com/file.txt          # Download a file to the current directory  
wget -O!--output-document foo.txt https://example.com/file.txt # Output to a file with the specified name
```

Linux

Basic shell commands

User Management

```
sudo su          # Switch to root user  
sudo foo         # Execute commands(has permission denied) as the root user  
sudo nano /foo/foo.txt    # Open directories and files(is not writable) as the root user  
su username     # Switch to a different user  
  
passwd          # To change the password of a user  
adduser username      # To add a new user  
userdel username     # To remove user  
userdel -rl--remove username   # To remove user with home directory and mail spool  
usermod -al--append -G --groups GROUPNAME USERNAME # To add a user to a group  
deluser USER GROUPNAME        # To remove a user from a group  
  
last            # Display information of all the users logged in  
last username    # Display information of a particular user  
w               # Display who is online
```

Terminal Multiplexers

Start multiple terminal sessions. Active sessions persist even after log out.

```
tmux      # Start a new session (CTRL-b + d to detach)
```

```
tmux ls     # List all sessions
```

```
tmux attach -t 0 # Reattach to a session
```

```
screen      # Start a new session (CTRL-a + d to detach)
```

```
screen -S foo # Start a new named session
```

```
screen -ls    # List all sessions
```

```
screen -R 31166 # Reattach to a session
```

System Information

```
uname -s          # Print kernel name  
uname -r          # Print kernel release  
uname -m          # Print Architecture  
uname -o          # Print Operating System  
uname -a          # Print all System info  
  
lsb_release -a      # Print distribution-specific information  
dpkg --print-architecture # Print-architecture by name  
  
cat /proc/cpuinfo    # Show cpu info  
cat /proc/meminfo     # Show memory info
```

Secure Shell Protocol (SSH)

```
ssh hostname       # Connect to hostname using your current user name (p 22)  
ssh -i foo.pem hostname  # Connect to hostname using the identity file  
ssh user@hostname    # Connect to hostname using the user over the default SSH port 22  
ssh user@hostname -p 8765  # Connect to hostname using the user over a custom port  
ssh ssh://user@hostname:8765 # Connect to hostname using the user over a custom port
```

Secure Copy

```
scp foo.txt ubuntu@hostname:/home/ubuntu      # Copy foo.txt into the specified remote directory  
scp ubuntu@hostname:/home/ubuntu/foo.txt .    # Copy foo.txt from the specified remote directory
```

Visita a Cluster UOH



Consultas?

Consultas o comentarios?

Muchas gracias