Introducció a les infraestructures per a *big data*

PID_00286208

Remo Suppi Boldrito

Temps mínim de dedicació recomanat: 4 hores

Universitat Oberta de Catalunya

Remo Suppi Boldrito

Enginyer de Telecomunicacions. Doctor en Informàtica per la Universitat Autònoma de Barcelona (UAB). Professor del Departament d'Arquitectura de Computadors i Sistemes Operatius de la UAB.

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Josep Jorba Esteve

Primera edició: febrer 2022

Av. Tibidabo, 39-43, 08035 Barcelona
Autoria: Remo Suppi Boldrito
Producció: FUOC

Índex

Objectius

Els principals objectius d'aquest mòdul són els següents:

1. Conèixer els conceptes i les arquitectures bàsiques que permeten incrementar les prestacions d'un sistema informàtic.

2. Conèixer les principals estructures i agrupacions d'una infraestructura de processament de la informació, definir com s'han de moure les dades i saber quins avantatges i desavantatges té cadascuna.

3. Caracteritzar el concepte de *big data* (dades massives), quan s'aplica, quines característiques té i com s'utilitza per extreure informació basada en dades.

4. Analitzar els models de processament i d'interacció que es desenvolupen perquè els programadors puguin accedir a serveis distribuïts de manera estandarditzada i segura (API).

Conceptes més importants

L'estudiantat haurà de centrar la seva atenció en els següents conceptes fonamentals presentats en aquest mòdul:

- Prestacions d'una aplicació
- GPU
- Clúster
- Informàtica d'altes prestacions (*high performance computing*)
- Núvol (*cloud*)
- Dades massives (*big data*)
- IoT (*Internet of things*)
- Informàtica en la perifèria (*edge computing*)
- API (*application programming interface*)

1. Increment de les prestacions

Hi ha diverses tècniques per augmentar les prestacions d'un programa informàtic relacionades amb l'arquitectura de processador, com són, entre altres:

- la planificació dinàmica d'instruccions,
- la predicció dinàmica dels salts, i
- el *multithreading*.

Aquestes millores, acompanyades de l'increment del nombre de nuclis (*cores*), la programació en memòria compartida i els increments de memòria cau, permeten obtenir unes prestacions immillorables des del punt de vista de les aplicacions. Però com hem de procedir quan cal incrementar encara més aquestes prestacions?

La idea d'incrementar el **nombre de nuclis o de *threads*** té un límit que depèn del processador, però es podran obtenir millors prestacions si s'incrementa el nombre d'unitats de processament (CPU) i es transforma el sistema en un multinucli (*multicore*) amb múltiples CPU.

L'increment del nombre de *threads* –incrementant el nombre de *cores*, si és possible, o utilitzant les característiques d'*hyper-threading* dels processadors, si està disponible– està limitat per les característiques de la CPU, per la qual cosa per a determinats problemes és interessant contemplar opcions com les que ofereix una GPU.

> Una unitat de processament gràfic (GPU, *graphics processing unit*) és un coprocessador dedicat al processament de gràfics, però que es pot utilitzar per executar un determinat tipus d'aplicacions de propòsit general (GPGPU, *general-purpose computing on graphics processing units*).

Malgrat que una GPU és un processador dissenyat per al processament de gràfics 3D interactius, moltes de les seves característiques (per exemple, baix preu en relació amb la seva potència de càlcul, gran paral·lelisme, optimització per a càlculs en coma flotant) són molt interessants per incrementar les prestacions de determinades aplicacions en l'àmbit científic.

Hi ha diferències substancials entre les **arquitectures d'una GPU i d'una CPU**, i hem de ser conscients que no totes les aplicacions poden beneficiar-se dels avantatges que presenta una GPU. En concret, l'**accés a memòria** és una de les principals dificultats, ja que les CPU estan dissenyades per a l'accés aleatori

a memòria, la qual cosa afavoreix la creació d'estructures de dades complexes i la seva gestió mitjançant punters. En canvi, en una GPU l'accés a memòria està molt més restringit.

Memòria en un processador de vèrtexs i en un de píxels

En un processador de vèrtexs (la part d'una GPU dissenyada per transformar vèrtexs en aplicacions 3D) s'afavoreix el model de distribució, en el qual el programa llegeix en una posició predeterminada de la memòria, però escriu en una o diverses posicions arbitràries. En canvi, un processador de píxels, o fragments, afavoreix el model de recol·lecció, per la qual cosa el programa pot llegir en diverses posicions arbitràries, però escriure només en una posició predeterminada.

Per això el desenvolupador d'aplicacions GPGPU ha d'adaptar els accessos a memòria i les estructures de dades a les característiques de la GPU, utilitzant, per exemple, l'emmagatzematge de dades en una memòria 2D (on normalment s'emmagatzemaria una textura); l'accés a aquestes estructures de dades seria equivalent a una lectura o escriptura d'una posició en la textura, la qual cosa pot generar problemes perquè no es pot llegir i escriure en la mateixa textura, i si aquesta operació és imprescindible per al desenvolupament de l'algorisme, s'ha de fer en diferents seqüències.

Cal tenir en compte que encara que la majoria d'**algorismes** que es poden desenvolupar en una CPU també poden ser desenvolupats en una GPU, totes dues implementacions no seran igual d'eficients en les dues arquitectures. Els algorismes amb un alt grau de paral·lelisme, que no requereixen estructures de dades complexes i tenen una alta intensitat aritmètica, són els que més beneficis obtenen de la seva implementació en la GPU, i les seves prestacions són significativament millors que els equivalents en la CPU.

Quant al **desenvolupament del programari GPGPU**, inicialment es feia en llenguatge assemblador o bé en algun dels llenguatges específics per a aplicacions gràfiques (GLSL, Cg o HLSL), però avui dia s'utilitza CUDA (Nvidia), una extensió de C que permet la codificació d'algorismes en GPU de Nvidia. També, i cada vegada són més els programadors que ho fan, es pot emprar OpenCL, una combinació d'interfície i llenguatge de programació per al desenvolupament d'aplicacions paral·leles que puguin ser executades de manera transparent en diverses unitats de processament (CPU multinucli, GPU, etc.).

2. Infraestructura HPC i informàtica en núvol

Per incrementar encara més les prestacions, hi ha altres alternatives per al processament d'aplicacions que consisteixen en diferents **organitzacions del sistema d'informació**. Entre les principals, es poden esmentar les següents:

1) Clústers: s'aplica a agrupacions d'ordinadors units entre ells (generalment) per una xarxa d'alta velocitat que es comporten com si fossin un únic ordinador. S'utilitzen com a **entorns de processament d'altes prestacions** (HPC, *high performance computing*) i serveis per a aplicacions crítiques o d'alt rendiment, entre altres. El seu ús s'ha popularitzat per a aplicacions que requereixen un elevat temps de processament (per exemple, les científiques) i estan basats en infraestructura comercial (*blades* o *slices*) juntament amb una xarxa d'altes prestacions (per exemple, Infiniband), la qual cosa, gràcies a l'ús en la seva gran majoria de programari de codi obert (Linux o NFS, encara que cada vegada més es reemplaça per altres sistemes d'arxius distribuïts, com ara Ceph, Lustre, GlusterFS), de sistemes de gestió de cues (com SGE i Slurm) i de biblioteques (per exemple, OpenMPI per executar tasques distribuïdes sobre l'arquitectura o OpenMP per aprofitar la potencialitat dels sistemes multicore) permet desenvolupar i executar aplicacions concurrents o distribuïdes d'altes prestacions. Així es pot disposar d'una infraestructura que proveeix un alt rendiment, alta disponibilitat, amb balanceig de càrrega, eficient, escalable i a costos raonables.

Enllaços complementaris

Infiniband: <https://www.infinibandta.org>

SGE: <https://sourceforge.net/projects/ gridengine/>

Slurm: <http://slurm.schedmd.com>

OpenMPI: <https://www.open-mpi.org>

OpenMP: <https://www.openmp.org>

2) Superordinador: és una evolució funcional a gran escala d'un clúster (tot i que amb diferents arquitectures), però amb una capacitat de processament molt més elevada que aquest (i, per tant, que un ordinador de propòsit general). El seu rendiment es mesura en teraflops (10^{12} operacions de coma flotant per segon).

Els superordinadors més potents

El juny de 2020, el superordinador més potent, publicat en la llista del Top500 (els 500 superordinadors més potents del món), era el Supercomputer Fugaku, que és al RIKEN Center for Computational Science, al Japó, i es manté com a número u en l'última llista publicada el novembre de 2021. L'any 2020 tenia 7.299.072 nuclis, una potència de 415.530 teraflops i consumia 28.335 kW, i el 2021 va incrementar el nombre de *cores* a 7.630.848, amb una potència de 442.010 teraflops i un consum de 29.899 kW. A Catalunya hi ha el Marenostrum. La versió actual, el Marenostrum 4, el 2020 estava en la posició seixanta-tres de la llista i el novembre de 2021 en la posició 73, però ja està en fase d'actualització al Marenostum 5.

Enllaços complementaris

Llista del Top500: <https://www.top500.org/lists/top500>

Barcelona Supercomputing Center (Marenostrum): <https://www.bsc.es>

Actualització al Marenostum 5: <https://www.bsc.es/news/bsc-news/marenostrum-5-will-host-experimental-platform-create-supercomputing-technologies-%E2%80%9Cmade-europe%E2%80%9D>

És important esmentar que un superordinador pot estar basat en dos enfocaments diferents:

a) L'**evolució del clúster** (com, per exemple, el Marenostrum a Catalunya, el Finisterrae a Galícia o el Juqueen a Alemanya), en la qual els clústers s'utilitzen de manera dedicada per a la informàtica d'altes prestacions, comparteixen un espai físic i disposen d'una sèrie de recursos comuns, com ara xarxes de comunicació d'alta velocitat o un espai d'emmagatzematge compartit i distribuït.

Enllaços complementaris

Centro de Supercomputación de Galicia (Finisterrae): <https://www.cesga.es>

Juqueen: <https://www.fz-juelich.de/ias/ jsc/EN/Expertise/Supercomputers/de-installedSystems/JUQUEEN/ Configuration/Configuration.html>

b) La **distribució mitjançant una xarxa** formada per milers o centenars de milers d'ordinadors discrets connectats per internet, que es dediquen de manera no exclusiva a la solució d'un problema comú (generalment, cada equip rep i processa un conjunt de tasques petites i trasllada els resultats a un servidor central que els integra en una solució global).

Distribució per mitjà d'una xarxa

Són representatius de distribució mitjançant una xarxa projectes com Seti@home (http://setiathome.ssl.berkeley.edu), Folding@home (https://foldingathome.org), World Community Grid (http://www.worldcommunitygrid.org) o Climateprediction.net (http://www.climateprediction.net), molts d'ells basats en Boinc (una arquitectura programari de codi obert que permet que voluntaris aportin els seus recursos a un problema comú –paradigma conegut com *volunteer computing*–); segons la pàgina dels seus desenvolupadors, la potència de processament generada per aquesta arquitectura en tots els projectes que la utilitzen és de mitjana en vint-i quatre hores: 30.286 petaflops; voluntaris actius: 86.116, amb 933.264 ordinadors (novembre de 2020).

Enllaços complementaris

Boinc: <https://boinc.berkeley.edu>

Potència de processament de Boinc: <https://boinc.berkeley.edu/computing.php>

3) ***Grid***: es refereix a una infraestructura que permet la integració i l'ús col·lectiu d'ordinadors d'alt rendiment, xarxes i bases de dades propietat de diferents institucions, però units per una capa de programari (*middleware*) comú que permet utilitzar-los com si fos un únic superordinador. Amb la finalitat de col·laborar aportant els recursos informàtics gestionats per cada institució, les universitats o els laboratoris d'investigació s'associen per formar *grids* i així cedir els seus recursos temporalment per disposar també del processament equivalent a la unió de totes aquestes infraestructures quan el necessiten.

La idea del *grid* va ser establerta per Ian Foster i Carl Kesselman, els precursors en la creació de Globus Toolkit (http://toolkit.globus.org/toolkit), considerat com la primera eina per construir *grids* (avui dia obsolet i retirat).

Lectura complementària

Ian Foster; Carl Kesselman (2014). *The History of the Grid* [en línia]. <http://www.ianfoster.org/wordpress/wp-content/uploads/2014/01/History-of-the-Grid-numbered.pdf>

Projectes *grid*

Entre els grans projectes *grid* cal citar CrossGrid, EU-DataGrid o, més recentment, el projecte EGI-InSPIRE (https://www.egi.eu/about/a-short-history-of-egi), tots ells finançats per la Unió Europea.

És important destacar que el *grid* és un tipus d'infraestructura d'informàtica, l'àmbit d'utilització i propòsit natural de qual és el **científic (*e-science*)**, usat en universitats i laboratoris d'investigació, mentre que el núvol neix de la idea de la prestació de serveis per als negocis d'empreses i usuaris (*e-business*) des d'altres empreses, però tots dos són similars en paradigmes i estructures per gestionar grans quantitats de recursos distribuïts. Molts experts consideren aquestes dues vies com les millors per a l'ús massiu de recursos (sota diferents

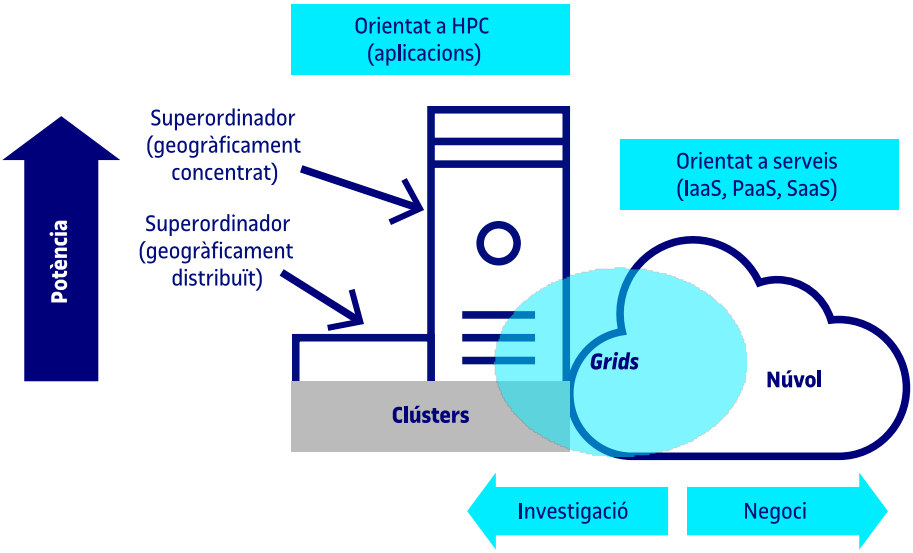
models): una per a l'e-ciència exclusivament (*grid*) i una altra per a les empreses (núvol). És important destacar que es comencen a oferir **serveis encreuats**, és a dir, altes prestacions (HPC) en el núvol.

HPC en el núvol

AWS HPC (https://aws.amazon.com/hpc) és un exemple d'altes prestacions en el núvol: instàncies amb base en maquinari específic i d'altes prestacions.

Després d'aquesta caracterització dels recursos i la seva utilització, podem situar gràficament cadascuna d'aquestes arquitectures en funció de la seva potència de processament i del tipus de dedicació (vegeu figura 1).

Figura 1. Arquitectures en funció de la seva potència de processament i el tipus de dedicació



Com es veurà en els següents mòduls, la principal tecnologia que dona suport a la informàtica en núvol (*cloud computing*) és la virtualització. La **capa de virtualització**, coneguda també com *hipervisor*, separa un dispositiu físic en un o més dispositius «virtuals» (anomenats *màquines virtuals*), que el client pot assignar, utilitzar i gestionar de manera molt simple, com si es tractés de màquines físiques, però amb els consegüents avantatges (aïllament, fàcil manteniment, posada en marxa, etc.). Avui dia tots els sistemes usen tècniques de virtualització assistides per maquinari (extensions del processador VT-x o AMD-V), per la qual cosa és possible tenir màquines virtualitzades molt eficients i configurades (mitjançant tècniques de clonat o *pool of VM in stand-by*) per ser assignades a un usuari sota demanda i en pocs segons.

Hipervisors

Alguns exemples d'hipervisors són KVM (Linux) (https://www.linux-kvm.org/page/main_page), VirtualBox (Linux, OS X, Windows i codi obert) (https://www.virtualbox.org), VMware ESXi (amb un versió gratuïta per a ús no comercial) (https://www.vmware.com/products/esxi-and-esx.html) o Hyper-V (disponible per a Windows 10) (https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v).

Un pas addicional que ha permès un increment notable de l'eficiència i la disponibilitat, la qual cosa ha comportat una revolució en els entorns de desenvolupament, ha estat la virtualització en l'àmbit del sistema operatiu per a la creació d'un sistema escalable de múltiples entorns independents amb un mínim impacte en la utilització dels recursos.

La virtualització del sistema operatiu (SO) permet que el seu nucli (*kernel*) pugui albergar múltiples instàncies d'usuari aïllades, on cadascuna actua com un «contenidor», amb les seves biblioteques i serveis, però utilitzant el SO subjacent.

Els noms que es donen a les instàncies són el de *contenidors*, *virtualization engines* (VE) o *jails* (per exemple, *FreeBSD jail* o *chroot jail*). Als ulls dels usuaris, aquest «contenidor» és similar a un servidor real i tindrà tots els elements necessaris, com si es tractés del servidor real o virtualitzat, però amb només una petita despesa de memòria, CPU i disc. Un dels desavantatges dels contenidors és que en compartir el SO no es poden tenir contenidors amb sistemes operatius que no comparteixin el mateix nucli (per exemple, si el *host* és Linux, no podrem posar-hi un contenidor amb Windows –la qual cosa sí que seria possible si fos una màquina virtual–).

Programari de virtualització del SO de codi obert

Entre el programari de virtualització del SO de codi obert podem citar Docker (https://www.docker.com), LXC/LXD (https://linuxcontainers.org), OpenVZ (https://openvz.org) i FreeBSD Jails (https://www.freebsd.org/doc/en_us.iso8859-1/books/handbook/jails.html), o, en programari propietari, Virtuozzo (basat en OpenVZ) (https://virtuozzo.com).

La informàtica en núvol també comparteix característiques amb altres models o paradigmes de processament distribuït (alguns de futur) com els següents:

1) *Dew computing*: nou paradigma informàtic distribuït que considera que els **equips locals** (ordinadors de sobretaula, portàtils i dispositius mòbils) proporcionen un entorn propici per a microserveis independents dels serveis en el núvol, i que aquests poden col·laborar amb els serveis. És a dir, aquest paradigma planteja la distribució de les càrregues de treball entre servidors del núvol i els dispositius discrets o locals per utilitzar plenament el potencial de tots dos. Alguns autors parlen més de *mobile cloud computing*, en la qual s'integren i es distribueixen els serveis i l'emmagatzematge de les dades en els dispositius mòbils i en els proveïdors de serveis en el núvol (algunes experiències acosten més aquest paradigma al *volunteer computing*).

2) *Edge computing*: és un paradigma d'informàtica distribuïda que proporciona serveis de dades, processament, emmagatzematge i aplicacions més a prop dels dispositius client o en els dispositius pròxims a l'usuari. És a dir, processa i emmagatzema les dades sobre els dispositius en la vora de la xarxa (per exemple, dispositius mòbils) en comptes d'enviar les dades a un lloc en el núvol. Es considera un tipus d'informàtica distribuïda **de proximitat**, en la qual cadascun dels dispositius connectats a la xarxa pot processar les dades i transmetre'n només un conjunt que siguin d'interès o fer-ho tan sols en situacions excepcionals (alarmes o notificacions), amb capacitat autònoma i sense dependència del servidor en el núvol. És un model que, amb el creixement que s'espera de la IoT (*Internet of things*), permetrà que els sistemes en el núvol i les xarxes no es col·lapsin davant l'increment exponencial de dades a transmetre, emmagatzemar i processar. Alguns autors ho diferencien del *fog computing* perquè en aquest s'utilitzen agrupacions o multituds d'usuaris finals (o dispositius pròxims a l'usuari) per emmagatzemar dades (en comptes de fer-ho sobre el núvol), i per la comunicació (en comptes de fer-ho per internet), el control, la configuració, el mesurament i la gestió (en comptes de ser controlat principalment per *gateways* de xarxa, com ocorre en la xarxa LTE, per exemple).

3) *Peer-to-peer computing*: aquest paradigma planteja l'ús d'una arquitectura distribuïda sense la necessitat d'una coordinació central per al processament i emmagatzematge de dades. Els participants són els **proveïdors i consumidors** de recursos (en contrast amb el model tradicional de client-servidor) i tots s'ajuden per processar, emmagatzemar i comunicar les dades de manera igualitària. Els *peers* posen una part dels seus recursos (potència de processament, emmagatzematge o amplada de banda de xarxa) directament a la disposició d'altres *peers* de la xarxa sense la necessitat d'una coordinació central pels servidors. Aquest tipus de paradigma ha tingut molta acceptació en serveis de gestió de continguts (Bittorrent, Spotify), compartició d'arxius (Gnutella, Edonkey), diners digitals (Bitcoin, Peercoin) i anonimització (I2P), entre altres.

Lectures complementàries

David Edward Fisher; Shuhui Yang (2016). «Doing More with the Dew: A New Approach to Cloud-Dew Architecture» [en línia]. *Open Journal of Cloud Computing* (vol. 3, núm. 1). <https://www.ronpub.com/publications /OJCC_2016v3i1n02_Fisher.pdf>

Atta ur Rehman Khan; Mazliza Othman; Sajjad Ahmad Madani; Samee Ullah Khan (2014). «A Survey of Mobile Cloud Computing Application Models» [en línia]. *IEEE Communications Surveys & Tutorials* (vol. 16, núm. 1). <http://ieeexplore.ieee.org/document/ 6553297>

Pedro García López i altres (2015, octubre). «Edge-centric Computing: Vision and Challenges» [en línia]. *ACM SIGCOMM Computer Communication Review* (vol. 45, núm. 5, pàg. 37-42). <https://doi.org/10.1145/2831347.2831354>

Jeff Burt (2014, 29 de gener). «Cisco Moving Apps to the Network Edge for Internet of Things» [en línia]. *eWeek*. <https://www.eweek.com/networking/ cisco-moving-apps-to-the-network-edge-for-internet-of-things>

Rüdiger Schollmeier (2001). «A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications» [en línia]. *Proceedings First International Conference on Peer-to-Peer Computing* (pàg. 0101). <https://doi.org/10.1109/P2P.2001.990434>

4) ***Volunteer computing***: com s'ha esmentat anteriorment quan parlàvem dels superordinadors distribuïts, aquest tipus de processament distribuït es basa en el fet que els **usuaris** aporten els seus recursos (sobretot processament i emmagatzematge) per a un determinat projecte del qual formen part. El primer registre que es té d'aquest tipus d'informàtica distribuïda és de 1996, el *Great Internet Mersenne Prime Search* (cerca de nombres primers que compleixen amb $2^n - 1$, per exemple, 3, 7, 31, etc.), seguit de distributed.net (el 1997). Després van venir-ne molts altres, entre ells Bayanihan (http://groups.csail.mit.edu/cag/bayanihan), els desenvolupadors del qual van ser els qui van proposar el nom de *volunteer computing*. En l'actualitat, molts d'ells estan basats en l'arquitectura Boinc (desenvolupada per la Universitat de Califòrnia, a Berkeley), amb projectes com els ja esmentats (Seti@home) o els que han desenvolupat la seva arquitectura programari, com Folding@home (Universitat d'Stanford), que el maig de 2016 van anunciar que havien arribat a la marca de cent petaflops (×86).

Lectura complementària

Luis F. G. Sarmenta (2001). *Volunteer computing* [en línia]. Cambridge: Massachusetts Institute of Technology. <http://people.csail.mit.edu/lfgs/papers/sarmenta-phd-mit2001.pdf>

3. Informàtica en núvol

Com a complement del punt anterior, i atès que ja hem parlat de *núvol* (una altra paraula també de moda), aquest concepte s'ampliarà per establir algunes idees i referències quant a què significa la informàtica en núvol (*cloud computing*), els avantatges i desavantatges d'aquesta manera de disposar i de gestionar recursos que ja forma part de les nostres vides (xarxes socials, comptes de correu, aplicacions ofimàtiques, etc.) i que permet només amb un navegador (o una aplicació d'un mòbil) poder processar les nostres dades i gestionar-ne la informació de manera remota.

Cloud computing és una proposta tecnològica adoptada per la societat en general com a forma d'interacció entre proveïdors de serveis, gestores, empreses o administracions i usuaris finals per a la prestació de serveis i la utilització de recursos en l'àmbit de les tecnologies de la informació i la comunicació (TIC), i sustentat per un model de negoci viable econòmicament.

Una definició interessant que aporta més concreció sobre la informàtica en núvol és la del National Institute of Standards & Technology (NIST):

> «*Cloud computing* és un model que permet accés ubic, a conveniència, sota demanda i per xarxa a un conjunt compartit de recursos informàtics configurables (per exemple, xarxes, servidors, emmagatzematge, aplicacions i serveis) que poden ser ràpidament aprovisionats i alliberats amb el mínim esforç d'administració o sense interacció amb el proveïdor de serveis».

Aquest paradigma, inqüestionable quant a la seva acceptació actual, vincula proveïdors, usuaris i tota la seva cadena intermèdia de transformació i de processament de la informació mitjançant una xarxa que, en gran part dels casos, és internet. És habitual que els usuaris utilitzin serveis en el núvol (tant si són serveis empresarials com personals), com ara el correu electrònic, les xarxes socials o l'emmagatzematge d'arxius (fotos, vídeos, ofimàtica), i que els usin des de diversos dispositius i amb diferents interfícies. Aquest paradigma és tan comú que els usuaris digitals (joves generacions) ja no coneixen una altra manera de fer-ho: d'una banda, han nascut amb aquesta tecnologia que, juntament amb la del mòbil, ha transformat la societat actual i, de l'altra, la interconnectivitat o usabilitat de diferents interfícies o mitjans d'accés són essencials per al seu quefer quotidià (compres, oci, serveis administratius, negoci i una llarga llista d'altres coses).

3.1. Breu història

Sense oblidar les grans companyies precursores que van oferir aquest tipus de serveis, com Yahoo! el 1994 i Google el 1998 (motor de cerca), Hotmail el 1997 (correu), Amazon el 2002 i AWS el 2006, el concepte enfonsa les seves arrels en la dècada de 1960 i en les idees de John McCarthy (professor emèrit de la Universitat d'Stanford i cofundador del Laboratori d'Intel·ligència Artificial del MIT), qui, el 1961, durant un discurs per al centenari del MIT, va predir que l'**ús a temps compartit** (que estava en auge en aquell moment) dels ordinadors podria conduir a un futur en el qual la potència de processament, i fins i tot d'aplicacions específiques, podria ser venut com un **servei**.

Aquesta idea, encara que des d'un punt de vista diferent, també va ser expressada per altres investigadors, especialment per Joseph Carl Robnett Licklider (conegut com JCR o simplement Lick), una de les figures més importants en l'àmbit de la ciència computacional i recordat particularment per ser un dels primers que va imaginar la informàtica interactiva moderna i la seva aplicació en diferents activitats amb una visió de la interconnexió global d'ordinadors (molt abans que fos implementada). JCR va treballar i aportar finançament en projectes com Arpanet (predecessor d'internet) o la interfície gràfica d'usuari, que van permetre l'accés a serveis i recursos a persones no especialistes, idees que, en opinió d'experts actuals, s'assemblen molt a la informàtica en núvol tal com la coneixem avui dia.

Tanmateix, la història de la informàtica en núvol també se sustenta en visionaris com John Burdette Gage. Durant la seva estada laboral a Sun Microsystems, Burdette va vaticinar que «***the network is the computer***», i més recentment, George Gilder (cofundador del Discovery Institute) va escriure el 2006 l'article «The information Factories», en el qual afirmava que el PC ja no era el centre d'atenció i d'emmagatzematge i processament de la informació, lloc que havia cedit al núvol, que seria el responsable d'emmagatzemar les dades de cada individu del planeta i al qual accedirien almenys una vegada en la vida. L'any 2013, Gilder va publicar *Knowledge and Power: The Information Theory of Capitalism and How It is Revolutionizing Our World*, obra en la qual relacionava l'economia, el capitalisme i la teoria de la informació d'Alan Turing i Claude Shannon, i com afectarien la societat.

Les evolucions des d'aquests anys seixanta inicials han estat notables i intrèpides, però probablement la més destacable en els aspectes més vinculats al núvol actual és el **desenvolupament de la Web**[1] i la seva evolució més recent a la Web 2.0 i el desenvolupament d'internet (que a Espanya no va tenir un impacte substancial fins a principis de la dècada de 1990 amb la transformació de RedIris i la connexió plena a internet).

(1)Tim Berners-Lee proposa les idees el 1989, però no va ser fins al 1990 quan es va crear un prototip de la *World Wide Web*.

A partir d'aleshores comencen a sorgir una sèrie de **serveis**, bàsicament aplicacions centrades en la cerca i el correu, en pàgines com Wandex (1993, que pretenia ser un programa per mesurar la dimensió d'internet desenvolupada

pel MIT, però que va acabar essent el primer cercador), Yahoo! (1994), Altavista (1995, que ha anat canviant de propietaris, des d'Overture, Yahoo i ara Microsoft), Hotmail (1996, posteriorment comprat per Microsoft el 1997, quan ja tenia sis milions d'usuaris de correu) o més recentment el totpoderós Google (1998).

Per a alguns autors, una de les grans fites vinculades al núvol és l'arribada, el 1999, de **Salesforce.com**,[2] una de les primeres empreses a oferir al seus clients el que avui dia es coneix per SaaS (*software as a service*) a partir d'un programari per a l'automatització de vendes mitjançant una simple pàgina web. Alguns experts consideren que aquest servei actualment es pot comparar amb un PaaS (*plataform as a service*) i altres el consideren una plataforma híbrida que ja ofereix les dues variants. Això va permetre mostrar una «nova» forma de negoci mitjançant internet i generar el que actualment és una realitat en diferents nivells d'aplicacions, serveis i recursos a la xarxa, coneguda com a informàtica en núvol (*cloud computing*).

(2)Empresa que ha estat considerada la més innovadora en els últims quatre anys en el rànquing elaborat per Forbes.

L'any 2002, Amazon va anunciar la seva infraestructura en la xarxa i l'ampliació el 2006 amb el llançament d'**Amazon Web Service (AWS)**, que incorporava *elastic compute* (conegut com a EC2) i emmagatzematge en el núvol (*storage cloud* o S3), els serveis més importants d'AWS per proveir instàncies d'un servidor sota demanda per a processament i emmagatzematge, respectivament. Aquests serveis orientats a negoci van permetre a les petites empreses i a particulars usar equips en els quals executar les seves aplicacions amb un model de monetització basat en el «pagament per ús». L'any 2009, Google i altres grans proveïdors van començar a oferir aplicacions basades en navegador, que es van fer molt populars i van guanyar en seguretat i servei. També altres grans proveïdors van sortir al mercat amb infraestructures pròpies en el núvol.

Proveïdors amb infraestructures pròpies en el núvol

Microsoft Azure es va anunciar l'octubre del 2008 (tot i que no va començar a donar serveis fins al 2010); IBM, el 2011, va llançar SmartCloud Framework per donar suport al seu projecte *Smarter Planet* i Oracle, el 2012, va anunciar Oracle Cloud.

Quant a les plataformes més representatives (no les úniques) per construir *clouds*, destaquem les següents:

- El 2005 es va començar a treballar en un projecte d'investigació orientat a aquest àmbit i basat en programari de codi obert: **OpenNebula** (http://opennebula.org), que va veure la llum el 2008. Aquest any es va formar el consorci Opennebula.org, finançat per un projecte de la UE (7th FP), que el 2010 va arribar a tenir 16.000 màquines virtuals (MV) configurades.

- L'any 2010, Rackspace i la NASA van llançar una iniciativa de programari de codi obert en el núvol, **OpenStack** (https://www.openstack.org), que va néixer amb l'objectiu d'oferir a les organitzacions la possibilitat de muntar els seus serveis de núvol sobre maquinari estàndard (el 2012 es va crear

l'OpenStack Foundation per promoure aquest programari en la comunitat, amb dos-cents socis, gran part de tots els actors importants del món TIC).

- També l'any 2010, una empresa anomenada Cloud.com va alliberar el seu producte orientat a la creació de *clouds*, **CloudStack** (en el qual havia estat treballant en secret durant els anys anteriors), sota GPLv3. El 2011, Cloud.com va ser adquirida per Citrix Systems, i CloudStack va passar a ser un projecte de la fundació Apache i a distribuir-se sota l'*Apache Software License* (https://cloudstack.apache.org).

Com a resum, podem dir que en l'actualitat és poc freqüent que una institució o una empresa mitjana o gran no tingui externalitzats alguns serveis en el núvol per als seus treballadors o usuaris finals o no utilitzi el núvol per a alguns dels aspectes vinculats al negoci (correu, web, intranet, etc.), excepte en les que el negoci principal depengui de la privacitat de les dades o aquestes estiguin especialment protegides.

3.2. Característiques, avantatges i desavantatges

Si tenim en compte les opinions dels experts (per exemple, Jamie Turner), a més del Web 2.0, les grans revolucions que han facilitat l'evolució de la informàtica en núvol són l'avenç de les tecnologies de virtualització (tècnica que permet posar més d'un ordinador amb el seu sistema operatiu executant-se en el mateix maquinari i compartir recursos), la proliferació a un cost acceptable de l'amplada de banda i els estàndards d'interoperabilitat de programari per facilitar que qualsevol recurs o servei pugui avui dia tenir com a base el núvol.

No obstant, aquestes opinions tan favorables cap al núvol han de considerar-se en un marc adequat i no caure en la temptació de creure que el núvol ho pot contenir i proveir tot. Com qualsevol sistema, té els seus elements positius (bàsicament accés a recursos i a serveis per part dels usuaris sense ser experts per a la seva instal·lació i administració, flexibilitat d'ús i adaptativa i preus acceptables), però també punts febles que convé tenir molt presents, especialment el seu taló d'Aquil·les: la disponibilitat 24/7 de l'accés a la xarxa i l'amplada de banda.

> Sense xarxa o amb una xarxa deficient el núvol no existeix.

Les característiques clau per a la informàtica en núvol són les següents (en ordre alfabètic):

1) Agilitat i autoservei: rapidesa i capacitat de proveir recursos (de manera gairebé immediata) sense grans intervencions ni accions per part de les dues parts. És a dir, l'usuari pot, de manera no assistida, aprovisionar capacitats de processament, emmagatzematge o xarxes segons necessiti i automàticament,

sense necessitat d'interacció humana amb el proveïdor de serveis. Aquests recursos han d'estar disponibles en un interval curt de temps (segons o, com a màxim, pocs minuts) en la xarxa.

2) Cost: atesa l'eficiència i l'automatització en la gestió i l'administració de recursos, els preus per als proveïdors en el núvol són reduïts i poden traslladar-los a l'usuari final, de manera que aquest no ha de fer front a les despeses derivades de la implantació d'infraestructura civil i dels serveis ni a la inversió en màquines, programari i recursos humans, la qual cosa fa que tot el model sigui molt favorable. A més, com que el control i el monitoratge són automàtics, es poden fer servir mesures amb un cert nivell d'abstracció, molt apropiades per a aquest tipus de servei (per exemple, comptes o comptes actius, emmagatzematge, processament o amplada de banda), de manera que és possible ajustar el cost a models de pagament per ús, pagament per peticions, etc., i es proporciona transparència tant per al proveïdor com per al consumidor del servei utilitzat.

3) Escalabilitat i elasticitat: aquests conceptes es refereixen a l'aprovisionament de recursos en (gairebé) temps real i a l'adaptabilitat a les necessitats de càrrega i d'ús d'aquests. És a dir, si podem preveure la càrrega i la utilització, no cal aprovisionar tots els recursos des de l'inici, com si es tractés d'una inversió local, sinó planificar-los per a quan la demanda ho requereixi, amb la conseqüent reducció del cost.

4) Independència entre el dispositiu i la ubicació: independitzar el recurs de l'accés i facilitar que els recursos puguin usar-se des d'una xarxa local, corporativa o un altre tipus, i des de diferents dispositius (capacitat o tipologia).

5) Manteniment i llicències: el model d'actualitzacions i de llicències se simplifica, ja que el programari estarà en el servidor del núvol i no hi haurà res instal·lat en el dispositiu de l'usuari final.

6) Rendiment i gestió de recursos: control exhaustiu i monitoratge eficient dels serveis per aconseguir una alta disponibilitat i una utilització òptima dels recursos de manera automàtica. Aquestes característiques permeten reduir al màxim les ineficiències i aporten control, notificació immediata, transparència i seguiment, tant al proveïdor com a l'usuari final.

7) Seguretat: es pot incrementar la seguretat perquè les dades estan centralitzades. La seguretat de l'aplicació correspon al seu responsable, mentre que al proveïdor li correspon la responsabilitat de la seguretat física. És a dir, la seguretat serà com a mínim la mateixa que la dels sistemes tradicionals, però podrà ser millorada i caldrà estipular quin nivell es desitja en el contracte de prestació de serveis, el SLA (*service level agreement*).

8) Virtualització com a base per a l'aprovisionament: això permet compartir i optimitzar l'ús del maquinari reduint els costos, l'energia consumida i l'espai, la qual cosa facilita el desplegament ràpid de serveis i garanteix l'alta disponibilitat (serveis en espera) i la mobilitat per càrrega (moviment de màquines virtuals a servidors més ociosos quan la càrrega augmenta).

Per convèncer els més reticents, podem enumerar un conjunt d'avantatges que aporta el núvol com a paradigma (en ordre alfabètic):

1) Fàcil integració i acceptació: gràcies al seu desenvolupament i al fet de basar-se en estàndards, es pot integrar amb molta més facilitat i rapidesa a la resta de les aplicacions empresarials. L'usuari final queda totalment convençut perquè se li permet accedir des de qualsevol dispositiu i sense requeriments especials segons les seves necessitats.

2) Reducció de l'ús d'energia (consum eficient): atesa la tecnologia usada i l'eficiència en l'ús dels recursos (afavorida per la virtualització), només es consumeix el necessari, a diferència dels centres de dades tradicionals, en els quals hi ha un consum fix no dependent de la càrrega i de la utilització.

3) Reducció de riscos i rapidesa: no cal una gran inversió ni l'adequació de llocs i d'entorns i, a més, és possible accelerar al màxim la implantació de nous serveis en les diferents etapes de desenvolupament, fins i tot en la producció.

4) Servei global: ubiqüitat i accés als serveis des de qualsevol lloc, amb temps d'inactivitat reduït al mínim i amb alta disponibilitat de recursos.

5) Simplicitat: rols i responsabilitats molt ben definides, la qual cosa comporta una separació de les activitats ben estipulada (per exemple, proveïdor de continguts, proveïdor d'aplicació, proveïdor d'infraestructura i usuari final). Tot això es tradueix en una manera òptima de treballar i menys inversió per a totes les parts.

Encara que els avantatges són evidents i molts usuaris d'aquesta tecnologia l'empren sobretot per l'abaratiment dels costos de servei, comença a haver-hi opinions contràries que consideren que un núvol públic pot no ser l'opció més adequada per a determinada mena de servei o infraestructura. Alguns dels principals desavantatges que esgrimeixen els experts són els següents (per ordre alfabètic):

Lectura complementària

Paul Haefele (2012). «EC2 is 380% more expensive than internal cluster» [en línia]. *Deep Value*. <http://deepvalue.net/ec2-is-380-more-expensive-than-internal-cluster>

1) Centralització: tant de les dades com de les aplicacions, la qual cosa genera una dependència del proveïdor. Si aquest no disposa de la tecnologia adequada (monitoratge i detecció) ni dels recursos apropiats (alta disponibilitat), es poden generar talls o inestabilitats en el servei. En aquests casos, pren una especial rellevància el SLA (*service level agreement*), que especificarà a què està obligat el proveïdor i les indemnitzacions a les quals haurà de fer front per això.

2) Confiabilitat: la «salut» tecnològica i financera del proveïdor serà un element clau en la continuïtat del seu negoci, i també del nostre, per la qual cosa les decisions que prengui afectaran directament el negocio client i l'empresa. Aquesta última podria quedar a la mercè d'un mercat molt dinàmic quant a fusions i a monopolis (o pseudomonopolis), amb el consegüent impacte que això podria tenir en els costos dels serveis.

3) Dependència d'un proveïdor (*vendor lock-in*): és un dels grans problemes detectats i que en l'actualitat s'ha demostrat com un dels més freqüents. En dependre d'un proveïdor de productes i de serveis de manera contractual, moltes vegades els clients no poden usar els serveis d'altres proveïdors per la penalització dels elevats costos que implicaria el canvi, encara que això comporti una atraient reducció dels seus costos o l'accés a millors prestacions. Molts desenvolupadors o usuaris finals són reticents al núvol sobretot per aquest motiu.

4) Disponibilitat: el principal punt feble d'una infraestructura en el núvol és l'accés a internet. Si no es disposa d'un accés de confiança i amb una amplada de banda acceptable, el núvol deixa de ser efectiu.

5) Escalabilitat: com més clients tingui el proveïdor, a més usuaris haurà de proveir el maquinari, amb la sobrecàrrega del sistema que això implica; si el proveïdor no disposa d'un pla d'escalabilitat a mitjà i llarg termini, de manera que pugui assegurar un creixement sostenible des del punt de vista de les necessitats dels clients, es pot arribar a la saturació dels serveis, amb la posterior degradació i pèrdua de prestacions.

6) Especialització o qualificació: la necessitat de serveis «especials» o qualificats podria tenir una prioritat molt baixa (i originar el retard en la seva implementació) per al proveïdor si la demanda és poca.

7) Risc i privacitat: les dades «sensibles» de negoci no resideixen en les instal·lacions de les empreses, per la qual cosa la seva seguretat no depèn dels recursos humans, sinó del proveïdor del servei. Si s'assumeix que les dades són d'alt valor en un context vulnerable, el risc pot ser molt alt pel possible robatori (còpia), accés a la informació (lectura) o destrucció (esborrat).

8) Seguretat: el fet que la informació hagi de travessar diferents canals i serveis fa que cadascun d'ells pugui convertir-se en un focus d'inseguretat. Si bé això es pot resoldre mitjançant canals i serveis segurs, la possibilitat d'una errada en la cadena de xifrat de la informació augmenta. A més, és fàcil que el propietari de la informació desconegui per complet què ha passat i on s'ha produït l'errada.

Com es pot veure, entre característiques, avantatges i desavantatges pot haver-hi elements que entren en contradicció (per exemple, el tema de la seguretat). Així doncs, l'equip encarregat de prendre les decisions de l'empresa haurà de sospesar acuradament les diferents possibilitats, amb els avantatges i els riscos.

Des d'un punt de vista global, la informàtica en núvol és una tecnologia que aporta beneficis: després d'una adequada planificació, després de valorar bé tots els factors que influeixen en el negoci, i deixant de banda els conceptes superficials (tothom ho té, tothom ho utilitza, baix cost, expansió il·limitada, etc.), pot ser una elecció adequada per als objectius de l'empresa, per al negoci i per a la prestació de serveis TIC que necessita.

La figura 2 mostra una visió general dels actors i dels elements en joc en la informàtica en núvol:

- **Infraestructura**: recursos de maquinari i programari que gestiona el proveïdor i que seran usats per les empreses i els clients.

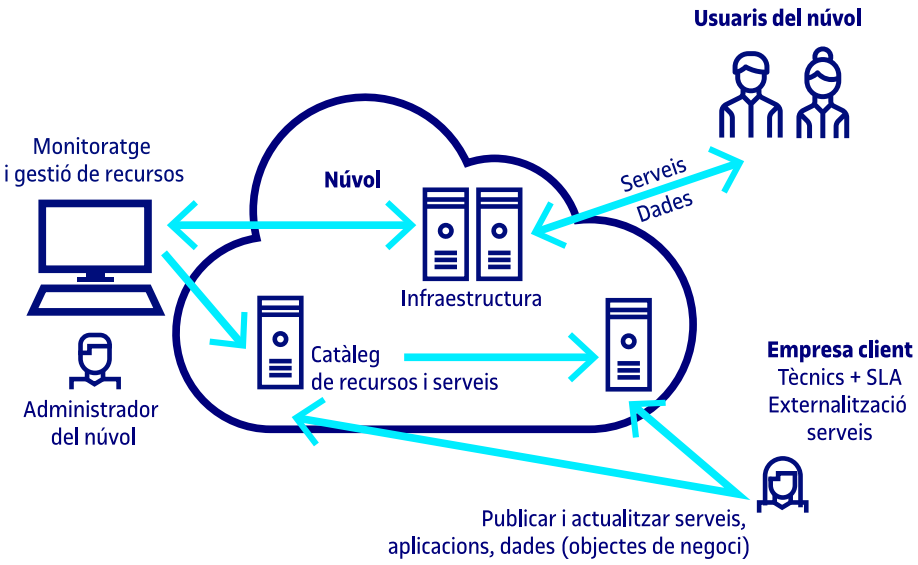
- **Catàleg de serveis**: elements oferts pel núvol que seleccionaran els clients del proveïdor de serveis per categories o prestacions.

- **Administrador del núvol**: cos tècnic de professionals que garantiran les prestacions, la seguretat, l'estabilitat, la disponibilitat, etc. dels serveis comercialitzats a partir d'un SLA signat amb cada client.

- **Empresa client**: usuaris que duen a terme una externalització dels seus serveis TIC i publiquen i actualitzen les seves aplicacions o dades en el núvol. Es basen en una garantia del servei que ofereix el proveïdor i que forma part d'un acord de prestació de serveis (SLA), el cost dels quals es basarà generalment en polítiques similars al pagament per ús.

- **Usuaris**: clients de l'empresa que ha posat els seus serveis en el núvol, que poden saber o no que aquests serveis són proveïts des del núvol. Només accedeixen a un servei, una aplicació o dades, com si entressin en els servidors de l'empresa que proveeix el servei.

Figura 2. Actors i elements en joc en la informàtica en núvol



És important destacar que els usuaris (de vegades anomenats *usuaris finals*) generalment no són clients del proveïdor de serveis del núvol, sinó de l'empresa que proveeix els serveis en el núvol.

Clients d'Amazon AWS

Entre els principals clients d'Amazon AWS (és a dir, parlem d'empreses que tenen l'etiqueta «milions de dòlars» associada a alguna característica –ingressos, actius totals, finançament, valoració o beneficis–) hi ha Adobe Systems, Airbnb, Alcatel-Lucent, Aon, Autodesk, BMW, Bristol-Myers, Canon, Capital One, Comcast o Docker, entre molts altres. Per a Adobe Systems, per exemple, el proveïdor del núvol i de la infraestructura és AWS, i l'empresa que contracta és Adobe, que comercialitza els productes mitjançant Creative Cloud (diferent programari de disseny gràfic, edició de vídeo, disseny web i serveis en el núvol), que arriben als usuaris (finals) per una quota econòmica mensual.

3.3. Classificació

Hi ha diferents taxonomies per descriure els serveis i la forma de desplegament del núvol. Com a models de desplegament podem enumerar els següents:

Lectura recomanada

En el següent document es defineixen quatre models de desplegament i tres de servei en el núvol:

Peter Mell; Timothy Grance (2011, setembre). «The NIST Definition of Cloud Computing» [en línia]. *National Institute of Standards and Technology*. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800 -145.pdf>

1) Núvol públic: el sistema és obert per a ús general sota diferents models de negoci (des de pagament per recursos, per ús, quota o lliure). Aquí el recurs és mantingut i gestionat per un tercer, i les dades i les aplicacions dels diferents clients comparteixen els servidors, els sistemes d'emmagatzematge, les xarxes i altres infraestructures comunes. Normalment, els recursos s'utilitzen i es gestionen per internet. En general, els clients no saben amb qui comparteixen la infraestructura. L'ús es contracta per mitjà d'un catàleg de recursos disponibles i el seu aprovisionament és automàtic (o semiautomàtic), després d'haver complert una sèrie de tràmits quant al model de pagament i d'SLA.

El propietari del núvol (proveïdor de recursos i de serveis) pot ser una empresa privada, una institució acadèmica, una organització governamental o una combinació de totes, generalment en funció del tipus de clients i dels recursos que s'han de proveir. Tècnicament pot haver-hi poques diferències (o cap) amb algun dels altres models (especialment amb el privat); tanmateix, sí que

pot haver-hi diferències substancials pel que fa a la seguretat: el núvol públic pot estar disponible en una xarxa oberta com internet i sense mecanismes de xifrat.

Núvol públic

Com a exemple de serveis de núvol públic, podem esmentar AWS (Amazon Web Services), Microsoft Azure, Google Compute, Rackspace, AliBaba Cloud o IBM-SoftLayer, entre altres, que operen la seva infraestructura i tenen els seus recursos accessibles per internet. També hi ha el CSUC (Consorci de Serveis Universitaris de Catalunya), que ofereix, sota diferents models d'explotació, recursos de núvol per a institucions acadèmiques i d'investigació de Catalunya (https://www.csuc.cat/ca/serveis/serveis-en-nuvol).

2) Núvol privat: en aquest cas, la infraestructura funciona exclusivament per a una organització o empresa i s'utilitza per les seves unitats de negoci o departaments. L'empresa pot ser propietària, administradora i operadora, però algun d'aquests elements també pot estar subcontractat a un tercer. És l'opció més favorable per a les companyies que necessiten una alta protecció de dades i un alt nivell de servei, ja que els recursos i la seva gestió són sota el control i la cura de l'empresa o de l'organització mateixa.

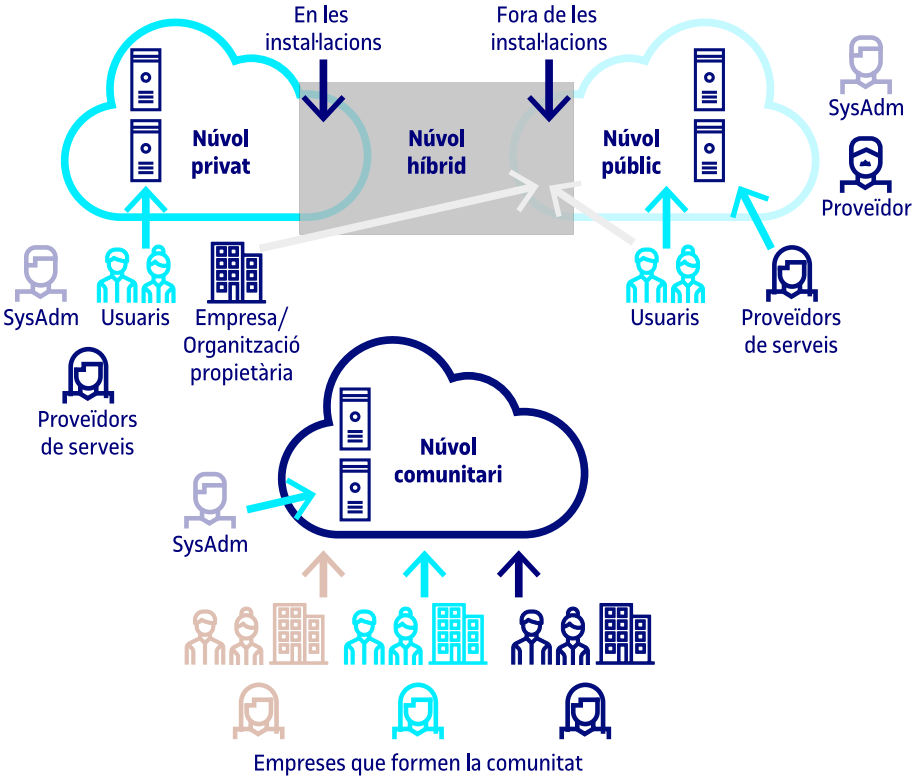
Encara que aquest model resol alguns problemes de cert tipus d'empreses (per exemple, les que tenen una legislació especial, com ara les públiques), aquestes han d'assumir el cost de la inversió (CAPEX). Com a contrapartida, disposen d'una infraestructura sota demanda gestionada per personal propi (o extern, però sota els seus criteris) que controla quines aplicacions usar i on han d'executar-se, de manera que mantenen la privacitat de la seva informació; això permet definir les polítiques d'accés i evitar el *lock-in* de les dades, així com la dependència d'un proveïdor esmentada anteriorment.

3) Núvol híbrid: es combinen models públics i privats, de manera que el propietari disposa d'una part privada (amb accés controlat i sota el seu control) i d'una altra de compartida, encara que d'una manera controlada. Els núvols híbrids ofereixen la possibilitat d'escalar molt de pressa, amb aprovisionament extern sota demanda, però impliquen una gran complexitat a l'hora de decidir com es distribueixen les dades i les aplicacions en una part o una altra. Encara que es tracta d'una proposta atractiva per a les empreses, habitualment el seu ús no passa d'aplicacions simples, sense restriccions de sincronització i amb bases de dades sofisticades (per exemple, les implementacions de correu empresarial o les aplicacions d'ofimàtica).

4) Núvol comunitari: els serveis estan dissenyats perquè els pugui utilitzar una comunitat, una organització o una empresa sota el mateix alineament d'objectius o de negoci (per exemple, bancs, distribuïdors, arquitectes, etc.), o que requereixin unes característiques específiques (per exemple, seguretat i privacitat). Les empreses que formen part de la comunitat poden ser propietàries de la infraestructura, així com gestores o operadores; alguns d'aquests rols també es poden subcontractar a tercers, però sota les indicacions i les regles que aplica la comunitat.

La figura 3 mostra un esquema d'aquestes quatre formes d'implantació i de desenvolupament dels núvols.

Figura 3. Formes d'implantació i de desenvolupament dels núvols



Una altra de les classificacions habituals és a partir del **nivell de servei** que presten. En aquest cas, les tres tradicionals són les següents:

1) IaaS (*infraestructure as a service*).
2) SaaS (*software as a service*).
3) PaaS (*platform as a service*).

Tanmateix, en l'actualitat es poden trobar altres extensions o derivades, com poden ser BaaS (*business as a service*), StaaS (*storage as a service*), DaaS (*desktop as a service*), DRaaS (*disaster recovery as a service*), MaaS (*màrqueting as a service*). Alguns autors els donen un nom global: XaaS (*everything as a service*) per referir-se a la creixent diversitat de serveis disponibles en el núvol mitjançant internet. Un cas que afirma aquestes consideracions és el d'aPaaS (*application platform as a service*), que permet un ràpid desenvolupament i aprovisionament d'aplicacions com a plataforma específica per a la codificació, aprovisionament i desplegament d'aplicacions, i que suporta el cicle de vida complet proporcionant una manera més ràpida de crear aplicacions.

En la **infraestructura com a servei (IaaS)** hi ha la capa de recursos bàsica (generalment màquines virtuals, xarxes i emmagatzematge) en la qual el client podrà instal·lar els seus SO i les seves aplicacions per implementar un servei o executar les aplicacions. El client no podrà gestionar o controlar el maquinari

subjacent, però sí el SO, l'emmagatzematge i les aplicacions i els serveis implementats en aquestes màquines, així com alguns aspectes de la connectivitat (subxarxes, tallafocs, dominis, etc.).

Proveïdors d'IaaS

Exemples de grans proveïdors d'IaaS (amb desenes a milers de MV) són Amazon EC2 (https://aws.amazon.com/free), Google Compute Engine (https://console.cloud.google.com/freetrial) o DigitalOcean (https://www.digitalocean.com) com a servei representatiu per a PIMES (en unitats de MV).

La reflexió de l'usuari en aquesta modalitat seria: «**Per què comprar, instal·lar i provar infraestructura cada *n* anys (obsolescència) i no llogar-la i pagar per ús?**». Un dels principals atractius de la IaaS és que els recursos estan disponibles sense fer obra civil (centre de dades), amb mínims recursos humans (no especialistes en TIC), sense una gran inversió inicial, escalable, eficient i a un cost acceptable.

La **plataforma com a servei (PaaS)** és l'encapsulació d'un ambient de desenvolupament i de la provisió d'una sèrie de mòduls que proporcionen una funcionalitat horitzontal (persistència de dades, autenticació, missatgeria, etc.). Una estructura bàsica d'aquest tipus podria consistir en un entorn que contingui serveis o aplicacions, biblioteques i API per a una finalitat específica (per exemple, per a una tecnologia de desenvolupament en particular sobre Linux i un servidor web més una base de dades i un ambient de programació com Perl o Ruby). És a dir, el client no gestiona ni controla la infraestructura (servidors, sistemes operatius ni emmagatzematge, ni cap mena d'elements de xarxa o de seguretat, etc.), però té el control de les aplicacions desplegades i de la configuració de l'entorn d'execució de les mateixes (bases de dades i *middlewares*). És habitual que una PaaS pugui donar serveis en tots els aspectes del cicle de desenvolupament i de proves de programari o en el desenvolupament, la gestió i la publicació de continguts.

Proveïdors de PaaS

Alguns exemples de PaaS són Codeanywhere (https://codeanywhere.com), Google App Engine (https://cloud.google.com/products), Heroku (https://www.heroku.com) o OpenShift (https://www.openshift.com).

Es podria dir que en aquesta modalitat el desig de l'usuari és una cosa així com «**necessito el programari OPQ instal·lat en el sistema operatiu RST i la base de dades UVW amb l'API XYZ**», per la qual cosa rebria tot el conjunt (HW, SO, base de dades, biblioteques, API, seguretat, GUI i altres eines) a punt per usar en pocs segons i desenvolupar aplicacions empresarials o mòbils, pàgines web, continguts, etc.

Finalment, el **programari com a servei (SaaS)** està en la capa abstracta (si bé hi ha una tendència força estesa a considerar que el SaaS és la capa més simple del PaaS, o la més baixa) i caracteritza una aplicació completa que s'ofereix com un servei. En general, les aplicacions sota aquest model de servei són accessibles per mitjà d'un navegador web i l'usuari no en té el control, encara

que en alguns casos se li permet fer algunes configuracions. D'aquesta manera, el client no necessita instal·lar l'aplicació en els seus ordinadors i així elimina el suport i manteniment del maquinari i del programari, a més de millorar el control i la gestió de les llicències, si són necessàries.

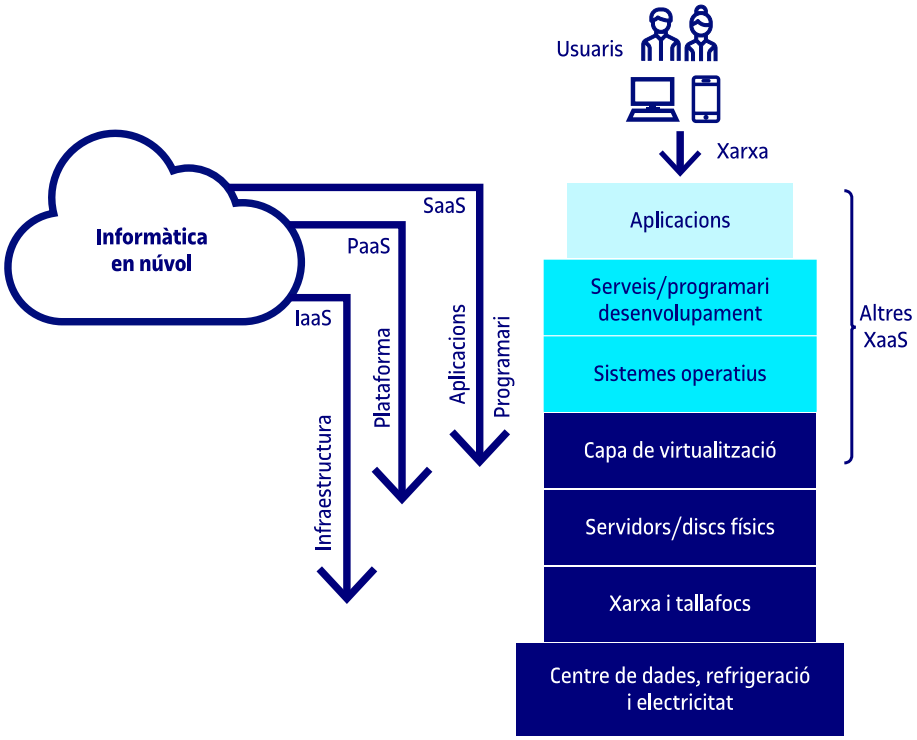
Proveïdors de SaaS

Alguns exemples representatius de SaaS són el correu web, amb Gmail (https://mail.google.com), Outlook (https://outlook.live.com), Yahoo (https://login.yahoo.com), etc.; SalesForce (https://www.salesforce.com/es/products), Google Docs (https://www.google.es/intl/es/docs/about), Office 365 (https://products.office.com/es-es/business/office) o SiteBuilder (https://www.sitebuilder.com).

Aquí el pensament de l'usuari seria una cosa així com «**executa'm això**», sense responsabilitats en la gestió del maquinari o del programari, de manera que utilitza un navegador web o evita la instal·lació de programari client basat en un servei sota demanda, amb escalabilitat gairebé immediata, accés des de qualsevol lloc i amb qualsevol dispositiu, sota un model de suport 24/7 i un model de pagament com el pagament per ús *pay as they go* i *pay as they grow*.

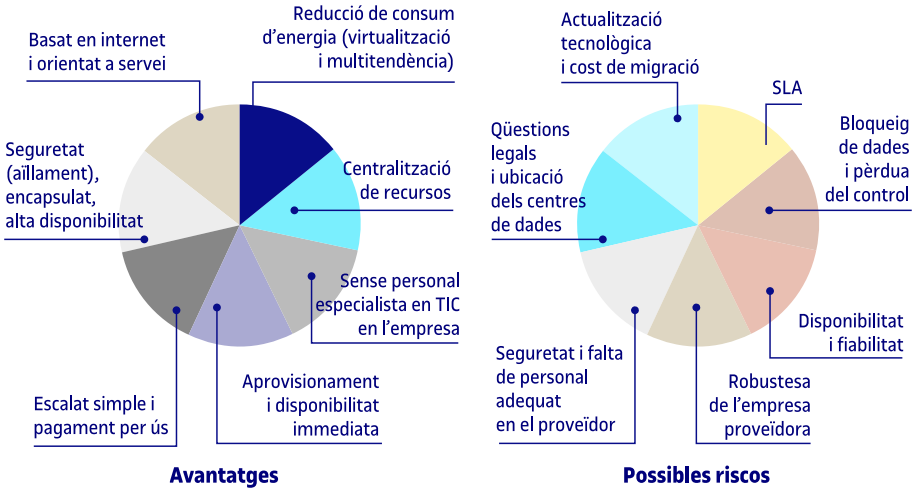
La figura 4 resumeix les diferents modalitats de servei del núvol i què implica cadascuna.

Figura 4. Modalitats de servei del núvol i les seves implicacions



En la figura 5 es mostren, de manera resumida, tant les característiques a favor de la informàtica en núvol com les que no són tan favorables, que poden representar un risc i que caldrà analitzar amb cura.

Figura 5. Avantatges i possibles riscos de la informàtica en núvol



4. El concepte de *big data*

Big data (dades massives) s'ha transformat en una expressió de moda. Tant en centres d'R+D+I com en les empreses, les indústries i les institucions cada vegada hi ha més quantitat de dades i calen noves tècniques i eines per processar-les i obtenir-ne informació. El Fòrum Econòmic Mundial va declarar les dades com un **nou actiu econòmic**, com la moneda o l'or. La revolució digital i electrònica no només ha augmentat enormement el volum de dades (com s'ha comentat anteriorment) en l'ordre d'uns quants exabytes diaris, sinó també la seva varietat (és a dir, estructurades, semiestructurades i no estructurades) i la seva velocitat de generació (milions de dispositius i d'aplicacions que generen enormes quantitats de dades cada segon).

Lectura complementària

Fòrum Econòmic Mundial (2012). *Big Data, Big Impact: New Possibilities for International Development* [en línia]. <https://www3.weforum.org/docs/WEF_TC_MFS_BigDataBigImpact_Briefing_2012.pdf>

Aquest immens creixement de les dades ha arribat al **límit del processament i de l'emmagatzematge** de les infraestructures de gestió de la informació existents, la qual cosa ha obligat les empreses a invertir en més maquinari i en actualitzacions de bases de dades. Aquesta tendència, que aparentment soluciona el problema inicial, en realitat no és una solució real, ja que el conjunt de dades continua creixent, la qual cosa condueix a un cicle de necessitar més maquinari per a més emmagatzematge, però les dades continuen creixent, i així indefinidament. A més, la infraestructura tradicional no és eficient a causa dels alts costos, de les limitacions d'escalabilitat (quan es tracta de petabytes) i de la incompatibilitat dels sistemes de bases de dades relacionals amb dades no estructurades.

Per tractar l'enorme quantitat de dades en la web, el 2004 Google va presentar un model de programació anomenat *MapReduce* (MR), amb la finalitat de fer, de manera paral·lela, tasques de cerca en els seus clústers de servidors. Per aconseguir aquest objectiu, van publicar les seves idees sobre un sistema d'arxius distribuïts (per tenir les dades a prop d'on s'haguessin de processar) el 2003 i després, el desembre del 2004, l'algorisme de processament **MapReduce**.

Basant-se en aquestes idees, dos investigadors, Doug Cutting i Mike Cafarella, van crear un sistema d'arxius i un entorn de processament, i van dur a terme les implementacions pertinents per executar Nutch (un motor de cerca) sobre aquesta infraestructura. L'any 2006, amb Doug Cutting treballant a Yahoo, van fer millores en el programari que havien provat amb Nutch i van crear un entorn anomenat *Hadoop* (http://hadoop.apache.org) com un projecte de codi obert en l'Apache Software Foundation (https://apache.org). Des d'aleshores, **Hadoop** s'ha convertit en l'estàndard *de facto* per emmagatzemar, processar

i analitzar centenars de terabytes o petabytes de dades com un projecte totalment de codi obert i disponible per a la comunitat que desitgi utilitzar-lo, fins i tot amb finalitats comercials.

Hadoop permet el processament distribuït d'una gran quantitat de dades en un conjunt de servidors de baix cost que emmagatzemen i processen les dades i poden escalar-les sense límits (aquest enfocament s'anomena *escalat horitzontal*).

Per exemple, segon les dades de l'any 2015, els clústers de Hadoop a Yahoo! incloïen més de 40.000 servidors i emmagatzemaven quaranta petabytes de dades d'aplicacions.

Lectures complementàries

Cade Metz (2011, 18 d'octubre). «How Yahoo Spawned Hadoop, the Future of Big Data» [en línia]. *Wired*. <https://www.wired.com/2011/10/how-yahoo-spawned-hadoop>

Sushant Gupta (2020, 2 de setembre). «What is big data analytics? Beginner guide to the world of big data» [en línia]. *Tricky Enough*. <https://www.trickyenough.com/big-data-analytics>

Després del llançament de l'entorn Hadoop, l'**analítica de dades massives** (*big data analytics*) es va transformar en el gran repte tecnològic que permetria a les empreses i a les institucions fer el salt estratègic de la retrospectiva a la prospectiva i extreure valor dels grans conjunts de dades. Com succeeix amb tota nova idea, molts escèptics estan en contra del «*big data* i del que comporta». Sostenen que durant dècades les empreses han pres decisions comercials basades en dades transaccionals emmagatzemades en bases de dades relacionals (DBRMS).

Aquestes crítiques obliden que hi ha un enorme potencial en les dades no tradicionals i no estructurades, com les dels registres web, les xarxes socials, el correu electrònic i les dades de sensors, imatges, àudio i vídeo, que poden contenir informació útil i valuosa. Aquestes opinions tampoc tenen en compte el que ocorre en el món real, on la presa de decisions basada en dades ha crescut exponencialment en els últims anys amb una reducció significativa de les decisions basades en les proves, l'instint o l'experiència.

D'altra banda, les ciutats intel·ligents (*smart cities*) i la internet de les coses (*Internet of things*), precisament gràcies a les dades massives, comencen a oferir una **gestió més eficient** de l'energia, l'aigua i els serveis de transport a les grans ciutats, la qual cosa permet que les ciutats puguin aconseguir un desenvolupament econòmic sostenible i una millor qualitat de vida.

Aquest tipus d'entorn (ciutats intel·ligents) obliga a fer anàlisis de dades per prendre decisions en temps real. Les eines de l'ecosistema de Hadoop fan possible recopilar, emmagatzemar i processar aquestes dades i aprofitar-les per a l'**anàlisi en temps real**. Tot això ha potenciat que cada vegada més empreses busquin incloure dades no estructurades, però potencialment molt valuoses, juntament a les seves dades empresarials tradicionals en les anàlisis d'intel·ligència empresarial.

A partir de les moltes definicions disponibles de dades massives podem resumir que es refereix a grans conjunts de dades diverses que creixen a un ritme cada vegada superior. Aquest concepte inclou el volum d'informació, la velocitat a què es crea i es recopila i la varietat de les dades que s'incorporen.

Les dades massives són un conjunt de dades massa gran, complex i dinàmic, de manera que no és factible per a qualsevol maquinari o programari convencional administrar i processar aquestes dades de manera eficient i escalable.

4.1. Les 5 «V» de *big data*

Per definir les característiques de les dades massives, inicialment els investigadors van plantejar les 3 «V», però avui dia es considera més adequat plantejar les 5 «V»: volum, velocitat, varietat, veracitat i valor (*volume*, *velocity*, *variety*, *veracity* i *value*).

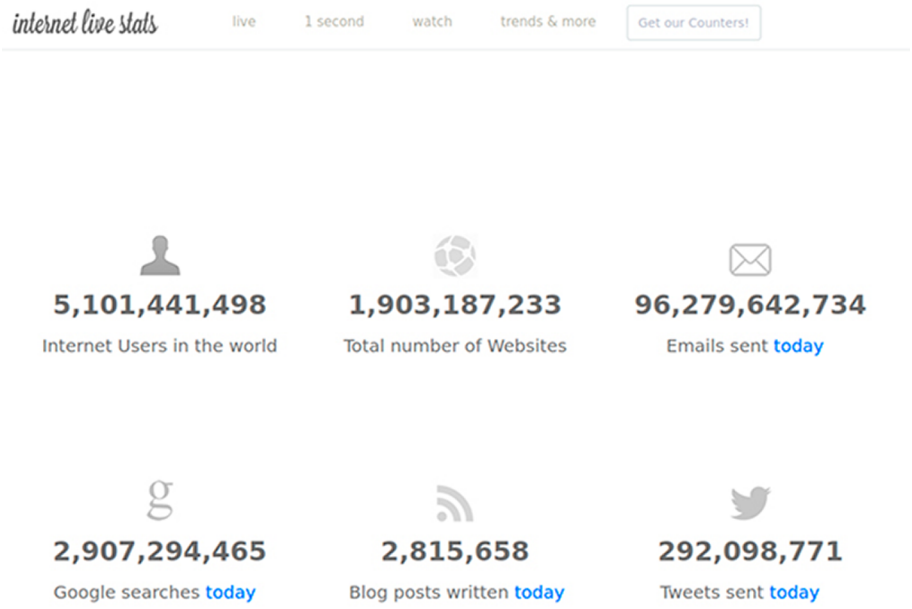
1) Volum: fa referència a la gran quantitat de dades generades en forma de correus electrònics, tuits, fotos, videoclips, dades de sensors, etc. Habitualment es parla d'exabytes diaris, zettabytes anuals i aviat de brontobytes. És interessant veure la quantitat d'informació que es genera en línia a https://www.internetlivestats.com (figura 6).

Lectures complementàries

O'Reilly Media (2012, 19 de gener). «Volume, Velocity, Variety: What You Need to Know About Big Data» [en línia]. *Forbes*. <https://www.forbes.com/sites/oreillymedia/2012/01/19/volume-velocity-variety-what-you-need-to-know-about-big-data>

Anil Jain (2016, 17 de setembre). «The 5 V's of big data» [en línia]. *IBM. Watson Health Perspectives*. <https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data>

Figura 6. Quantitat d'informació que es genera en línia



Font: <https://www.internetlivestats.com>

Figura 6

La figura mostra l'estat a les 08.20 hores del 8 de novembre de 2021 de les dades generades durant aquell dia, però és possible veure les quantitats per segon o per any.

2) Varietat: fa referència als diversos tipus de dades que es generen cada dia. En el passat, l'anàlisi se centrava en dades estructurades (bàsicament en taules de bases de dades relacionals), però en l'actualitat el 80 % de les dades mundials no estan estructurades i, per tant, no es poden processar fàcilment amb les tecnologies tradicionals de bases de dades.

3) Velocitat: fa referència a la velocitat a què es generen o processen i analitzen les noves dades. És interessant observar el tema de la velocitat en comptadors com Internet Live Stats. Per exemple, es pot considerar la velocitat a la qual es verifiquen les transaccions amb targeta de crèdit (l'any 2014, Visa processava 150 milions de transaccions per dia, 47.000 per segon) per detectar activitats fraudulentes o els mil·lisegons que triguen els sistemes de negociació a analitzar i captar moviments borsaris que desencadenen decisions per a la compravenda d'accions. És evident que aquesta enorme quantitat de dades no es pot emmagatzemar en bases de dades, per la qual cosa cal analitzar-les mentre es generen i després rebutjar-les.

Lectures complementàries

El Economista (2014, 28 de maig). «Data Center de Visa puede procesar 47,000 transacciones por segundo» [en línia]. *El Economista*. Mèxic. <https://www.eleconomista.com.mx/tecnologia/Data-Center-de-Visa-puede-procesar-47000-transacciones-por-segundo-20140528-0067.html>

4) Veracitat: fa referència a la incertesa o confiabilitat de les dades. Per a algunes dades, i sobretot quan són en quantitats significatives, la qualitat i la precisió són menys controlables.

Publicacions de Twitter

Considerem les publicacions de Twitter: dins del tuit hi ha etiquetes, abreviatures, errors tipogràfics i llenguatge col·loquial, elements que poden posar en dubte la confiabilitat o precisió del contingut perquè estan subjectes a interpretació. Les dades no tenen valor si no són precises (sobretot quan les analitzen màquines), i s'ha de tenir molta cura quan són aquestes dades les que formen el conjunt d'entrada a sistemes de presa automatitzada de decisions.

5) Valor: les dades massives són interessant per si soles, però tret que es puguin utilitzar per prendre decisions analítiques amb la intenció de millorar algun aspecte del sistema estudiat, no tenen gaire utilitat. Si bé és cert que les dades contenen una bona quantitat d'informació, el desafiament és identificar què té valor i després transformar i extreure les dades representatives per a la seva anàlisi. Amb això, les empreses i institucions poden desenvolupar una comprensió més profunda del seu negoci, els seus processos i projectes, etc., la qual cosa conduirà a un superior coneixement, productivitat i eficiència perquè les empreses generin, per exemple, una posició competitiva més sòlida. Molts experts en dades massives consideren que el «valor» és la característica més important de totes, la que determina l'èxit de qualsevol projecte que contempli les dades massives com a eina.

4.2. Desafiaments de *big data*

El creixement de les dades en diferents dimensions presenta molts desafiaments i oportunitats per a les organitzacions. Amb l'**augment del volum i de la velocitat** de generació de les dades és essencial que es pugui extreure informació útil en temps real per a la presa de decisions; en cas contrari, els negocis o els projectes correran el risc de veure's inundats per un allau de dades (*data deluge*) en entorns hipercompetitius.

Enllaç complementari

Podeu consultar la definició d'*allau de dades* de la Viquipèdia en el següent enllaç: <https://en.wikipedia.org/wiki/Information_explosion>

Per aquest motiu el desafiament és «travessar» els enormes volums de dades i accedir al **nivell de detall** necessari a alta velocitat; a mesura que n'augmenti la granularitat, això serà cada vegada més complex. Aquest tipus de situació haurà d'anar acompanyada d'un canvi en l'estratègia de negoci, ja que utilitzar dades en temps real implicarà prendre decisions en temps real, però també per a un negoci que es desenvolupi en temps real.

A aquest fet cal afegir altres elements que augmenten la complexitat: a part de l'augment de la velocitat i de la varietat de les dades, els seus **fluxos** poden ser molt inconsistents, amb pics periòdics (diaris, estacionals i desencadenats per esdeveniments com les dades de les xarxes socials) que seran molt difícils de gestionar. A més, les dades són no estructurades (correus electrònics, fotos, vídeos, dispositius de monitoratge, sensors, PDF, àudio, etc.), per la qual cosa la seva gestió és extremadament complexa.

Les varietats de **dades no estructurades** creen problemes d'emmagatzematge, de mineria i d'anàlisi. En dominis com consum, finances, seguretat, medicina, etc., la quantitat de possibles fonts és extremadament gran, la qual cosa fa inviable el processament en una sola màquina. Això obliga a buscar sistemes d'administració de dades que puguin gestionar bases de dades distribuïdes, sense esquemes i no relacionals.

D'altra banda, la **visualització** de dades pot comunicar tendències i valors atípics molt més de pressa que les taules que contenen números i text, però es torna una tasca difícil quan es tracta de quantitats extremadament grans d'informació o quan aquesta prové d'una gran varietat de categories. Generalment, els valors atípics representen entre l'1 i el 5 % de les dades, però quan es treballa amb quantitats massives de dades, veure aquest 1-5 % de les dades pot ser força difícil, i no és una tasca trivial representar aquests punts sense problemes de visualització.

Un altre desafiament important està en la **selecció** de les dades. Normalment es tracta d'un flux continu de dades entrants, gran part de les quals no tenen interès, de manera que poden filtrar-se i comprimir-se o simplement descartar-se. El repte és definir els filtres de manera que no descartin informació útil. Per exemple, es pot considerar la lectura d'un sensor que difereix substancialment de la resta: pot ser que el sensor estigui defectuós o pot ser que sigui la lectura la que necessita atenció. El repte serà plantejar-se com s'han de pren-

dre aquestes decisions i recórrer, per exemple, a tècniques de correlació, ja que generalment les dades de sensors estaran correlacionades espacialment i temporalment. Cal contemplar **tècniques per a la reducció del conjunt de dades** perquè aquestes puguin ser processades en línia perquè no hi haurà temps per emmagatzemar-les primer i després reduir-les.

A continuació veurem com es poden superar els desafiaments que sorgeixen a causa del volum, la varietat i la velocitat mitjançant ecosistemes com Hadoop, tot utilitzant les bases de dades NoSQL per resoldre els problemes que sorgeixen a causa de la varietat de les dades.

4.3. Arquitectura per a *big data*

Com en qualsevol entorn orientat a dades, una arquitectura per al processament de dades massives està formada per quatre nivells que interactuen més un global d'administració:

1) Recol·lecció de dades.
2) Emmagatzematge.
3) Processament.
4) Visualització.

Aquesta arquitectura és l'habitual en qualsevol entorn orientat a dades (per exemple, mineria de dades, intel·ligència empresarial o aprenentatge profund). No obstant, el concepte de *big data* ha fet que cadascun d'aquests nivells evolucioni i es generin noves eines, algorismes i entorns per adequar-se a les dades massives.

1) En la **recol·lecció de les dades** orientades a *big data*, han sorgit una gran quantitat d'eines orientades a aquest efecte (*data ingestion*), que permeten el processament seqüencial habitual de dades emmagatzemades (lot o *batch*) per obtenir el següent tram o seqüència (*chunk*) de dades des de l'última que s'ha llegit; en tot cas, en gran part i atès el volum d'aquestes, les eines han evolucionat per recol·lectar eficientment fluxos de dades (*streams*) en temps real.

2) Quant a l'**emmagatzematge**, també han sorgit grans canvis per adaptar-se a les característiques de les dades; això s'ha reflectit en una gran quantitat de desenvolupaments de motors de bases de dades (NoSQL, documents, columnes, clau/valor, grafs) i sistemes d'arxius distribuïts per adequar-se a la realitat de processament i, així, disposar d'escalabilitat, fiabilitat i proximitat de les dades per al seu tractament (per exemple, Apache HDFS, BeeGFS, DiscoDDFS, Google GFS, BaiduFS, GlusterFS, QuantcastFS, CephFS, GridGain-in-memoryFS, LustreFS).

3) Per al **processament**, com ja hem comentat, s'han dissenyat nous paradigmes que han caracteritzat tot l'entorn (i cadascuna de les capes per adequar-les a les seves necessitats); es coneixen com MapReduce (MR) o *massive parallel processing* (MPP).

a) MapReduce és un paradigma de programació el nom del qual prové de les dues funcions que el formen (*map* i *reduce*), i s'implementa en l'entorn Apache Hadoop (codi obert). Aquest paradigma, si bé no és la solució per a tots els problemes, treballa amb grans conjunts de dades (petabytes) i s'executa sobre sistemes d'arxius distribuïts (HDFS) i vinculats a eines com HBase, Hive, Impala o Cassandra (bases de dades).

Es tracta d'un paradigma apte per processar les dades que pot treballar sobre tuples del tipus [clau, valor] i on la funció `map()` processa en paral·lel les tuples d'un domini i genera una llista de parells en un domini diferent que s'agruparan sota la mateixa clau utilitzant un esquema de processament *master-worker* en forma d'arbre.

Posteriorment, la funció `reduce()` s'aplica de manera paral·lela per a cada grup i genera una col·lecció de valors per a cada domini. Un exemple típic és el procediment per comptar el nombre de vegades que apareixen les paraules en un text. Les funcions serien:

```
map(string name, string document):
    foreach word w in document:
        generate-tupla(w, 1);

reduce(string word, iterator partialList):
    int total = 0;
    foreach value in partialList:
        total += int(value);
    generate(word, total);
```

Funció `map()`

La funció `map()` divideix el document per paraules i genera les tuples [paraula, valor]. Per exemple, la famosa frase d'Alan Turing «La ciència és una equació diferencial. La religió és una condició de frontera» quedarà: [La,1], [ciència,1], [és,1], [una,1], [equació,1], [diferencial,1], [La,1], [religió,1], [és,1], [una,1], [condició,1], [de,1], [frontera,1]. L'entorn agruparà totes les claus iguals ['La,1,1', 'és,1,1', 'una,1,1' i la resta 'clau,1'] com a entrada a reduce(), que generarà l'agrupació i la suma dels valors de les claus, que en el nostre exemple serà [La,2], [és,2], [una,2] i la resta [clau,1].

b) MPP ofereix una solució tradicional adaptada a les dades massives basada en la divisió dels grans conjunts de dades en seccions (*slices*) que seran més fàcils de gestionar; cadascun d'ells serà assignat a un element informàtic per al seu processament. El dispositiu informàtic les processarà i, quan acabi, el sistema combinarà els resultats parcials per donar un resultat final (equivalent a una seqüència *fork-join* en un model *master-workers*). Atès que els elements

Enllaç complementari

Podeu consultar la definició del model *fork-join* de la Viquipèdia en el següent enllaç:
<https://en.wikipedia.org/wiki/Fork–join_model>

informàtics (processadors) treballaran sobre el seu segment de dades assignat de la BD i es comunicaran per missatges quan generin els resultats, no hi ha interacció entre ells; això es coneix com a «feblement acoblats» (altres autors ho anomenen *shared nothing*).

Entre les característiques principals d'aquests sistemes es poden citar les seves **possibilitats de sintonització i d'escalat il·limitat** (en principi, al nombre de nodes disponibles), amb el conseqüent increment del seu rendiment (pràcticament de manera lineal) i sense colls d'ampolla.

Hi ha gran quantitat de bibliografia sobre la comparació entre MR i MPP. Tanmateix, moltes vegades la solució no prové d'un o altre paradigma i, atès que es complementen bé quant a les prestacions i als tipus de dades que obtenen millors rendiments, és habitual que s'utilitzin arquitectures mixtes en les quals generalment primer s'aplica MR sobre dades no estructurades i, posteriorment, MPP per processar i visualitzar les dades estructurades generades en la primera operació.

Lectura complementària

Un exemple de bibliografia sobre la comparació entre MR i MPP és el següent:

Alexey Grishchenko (2015, 13 de juliol). «Hadoop vs MPP» [en línia]. *Distributed Systems Architecture*. <https://0x0fff.com/hadoop-vs-mpp>

4) Finalment, en l'últim nivell de l'arquitectura hi ha la **visualització** de dades massives. Fa referència a la implementació de tècniques de visualització usant estructures i gràfics adequats per obtenir «informació» a diferents nivells sobre les relacions existents en les dades. Les estratègies de visualització són diverses i inclouen aplicacions que poden mostrar canvis en temps real i gràfics més il·lustratius, tot deixant de banda els típics gràfics de dues variables (circulars, de barres o de línies). Aquestes visualitzacions deriven en una representació «més atractiva, més visual» de les dades, considerada per alguns autors com «una visualització més artística i menys artesanal» de les dades.

Normalment, quan les empreses o institucions necessiten presentar relacions entre dades utilitzen gràfics de barres i diagrames amb una varietat de colors, termes i símbols. No obstant, el principal problema amb aquest tipus de visualització és que generalment no funciona quan es volen presentar dades molt grans o molt variades (des de valors molt petits fins a molt grans), o dades que no tenen la mateixa unitat o amb quatre paràmetres, per exemple.

La visualització de dades massives està orientada a utilitzar il·lustracions gràfiques més interactives que tinguin personalització i animació, per mostrar figures i establir connexions entre peces d'informació i que sigui l'usuari qui triï l'escala o el grau de detall que vol veure.

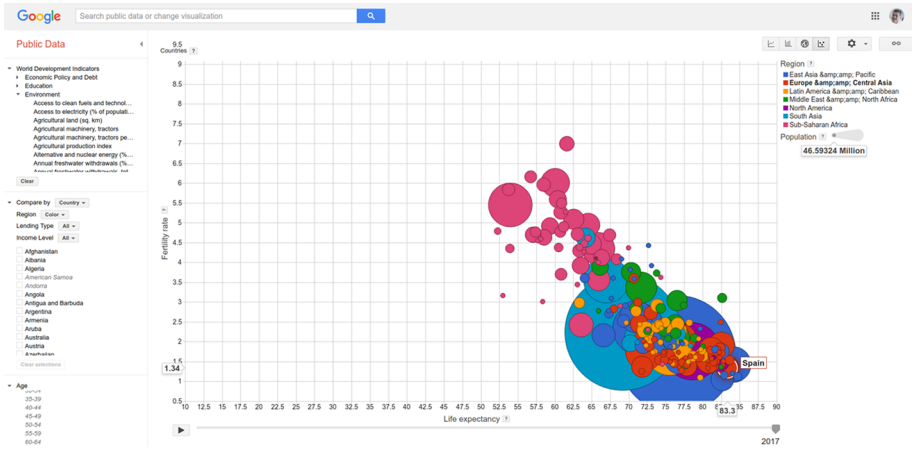
Enllaç complementari

Google Public Data Explorer: <https://www.google.com/publicdata/ directory>

Com a exemple, es pot esmentar (i es recomana interactuar amb aquest entorn de visualització de dades obertes) el **Google Public Data Explorer**. En l'exemple de la figura 7 es mostra la relació de l'esperança de vida i de la ferti-

litat per països, incloent-hi la població, i és l'usuari qui decidirà què vol veure: l'evolució des de 1960 fins a 2017 (o seleccionar qualsevol altra variable per inserir en el gràfic).

Figura 7. Exemple de visualització de dades massives amb Google Public Data Explorer



Eines per visualitzar dades massives

Hi ha una gran quantitat d'eines per visualitzar dades massives: Chartsblocks (http://www.chartblocks.com), Databox (https://www.adamenfroy.com/recommends/data-box), DataHero (https://datahero.com), Datawrapper (https://datawrapper.de), Plotly (https://plot.ly/), PowerBI (https://powerbi.microsoft.com) i Qlik (https://www.qlik.com); i per a desenvolupadors (les utilitzades per dissenyadors web o aplicacions per visualitzar *big data*): Chart.js (http://www.chartjs.org), Chartist.js (https://gionkunz.github.io/chartist-js), D3.js (http://d3js.org), Ember Charts (http://addepar.github.io/ember-charts), FusionCharts (https://www.fusioncharts.com), Google Charts (https://developers.google.com/chart), Highcharts (https://www.highcharts.com), n3-charts (https://github.com/n3-charts), NVD3.js (http://nvd3.org) i Processing.js (https://processing.org), entre altres.

4.4. Eines

El projecte Apache™ Hadoop® és una plataforma de codi obert per al processament distribuït, de confiança i escalable, utilitzada per una gran quantitat d'empreses i institucions que implementa l'algorisme de MapReduce.

> Apache Hadoop és un entorn per al processament distribuït de grans conjunts de dades mitjançant clústers informàtics. Utilitza models de programació senzills i té la possibilitat d'escalar des de servidors individuals a milers de processadors.

Bàsicament, Hadoop (V2.x o V3.x) està format per quatre mòduls:

1) HadoopYARN: entorn per a la planificació de tasques i la gestió de recursos del clúster.

2) Hadoop MapReduce: sistema basat en YARN per al processament paral·lel de grans conjunts de dades.

3) Hadoop Distributed FileSystem (HDFS): sistema d'arxius distribuït que proporciona accés d'alt rendiment a les dades de l'aplicació.

4) Hadoop Common: utilitats comunes que suporten els altres mòduls de Hadoop.

Tanmateix, Hadoop es pot relacionar, integrar o ampliar de manera fàcil i ràpida amb els següents projectes (només alguns dels més habituals o referenciats):

- **Ambari**: eina web per a l'aprovisionament, la gestió i el monitoratge de clústers Hadoop amb HDFS, MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig i Sqoop.

- **Avro**: serialització de dades.

- **Cassandra**: base de dades escalable sense punts d'errada únics.

- **Chukwa**: recopilació de dades en sistemes distribuïts.

- **Drill:** *SQL query engine* de baixa latència que suporta aplicacions distribuïdes per a l'anàlisi interactiva de *datasets*.

- **Flume**: sistema distribuït per recollir, agregar i moure grans quantitats de dades des de diferents fonts i també de *datastores* centralitzats.

- **HBase**: base de dades orientada a columnes distribuïda i escalable.

- **Hive**: magatzem de dades que proporciona resum de dades i consultes *ad hoc*.

- **Mahout**: entorn d'aprenentatge automàtic i mineria de dades.

- **Pig**: llenguatge de flux de dades d'alt nivell.

- **Spark**: motor de càlcul en memòria especialitzat a processar dades de fluxos de dades (*streams*).

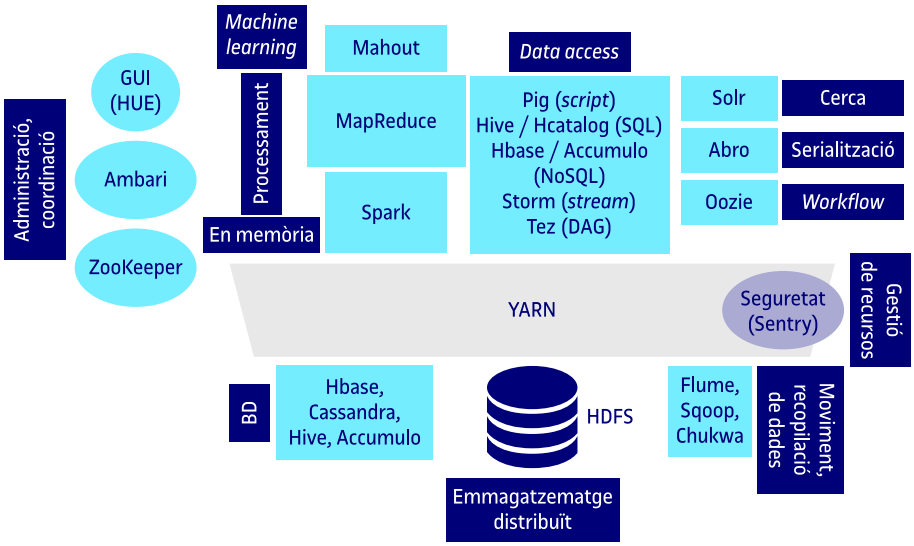
- **Sqoop**: eina per transferir dades entre Hadoop i bases de dades estructurades, com per exemple bases de dades relacionals.

- **Tez**: entorn de programació de flux de dades generalitzat per a grafs dirigits.

- **ZooKeeper**: servei de coordinació d'alt rendiment per a aplicacions distribuïdes.

La figura 8 mostra un entorn Hadoop amb els actors més importants i el seu rol (un dels possibles) dins de la plataforma.

Figura 8. Entorn Hadoop



4.5. NoSQL

Un terme important en el context de dades massives és ***NoSQL* (no només SQL)**, que abraça una àmplia varietat de tecnologies de bases de dades diferents que són **no relacionals, sense esquema, distribuïdes i escalables horitzontalment**. Estan dissenyades per fer front a les dades massives i a les aplicacions web en temps real que requereixen anàlisis de tipus de dades dispars i d'un volum extremadament alt.

Molts dels sistemes NoSQL no proporcionen garanties d'atomicitat, consistència, aïllament i durabilitat, a diferència dels sistemes de bases de dades relacionals, però això no significa un problema en el processament de les dades massives. Les bases de dades relacionals, d'altra banda, no estan dissenyades per fer front als desafiaments d'escala i d'agilitat de les aplicacions actuals ni per aprofitar l'emmagatzematge de baix cost ni la potència de processament distribuïda disponible en l'actualitat.

Atomicitat

Característica que assegura que una operació en la BD s'ha fet o no, i, per tant, davant d'una errada del sistema no pot quedar a mitges.

Avui dia hi ha més de dues-centes cinquanta bases de dades NoSQL (http://nosql-database.org), amb diferents visions i metodologies per a l'emmagatzematge de dades diferents. Hi ha diversos enfocaments per classificar les bases de dades NoSQL, cadascun amb diferents categories (i subcategories), però la classificació més utilitzada és la que es basa en el **model de dades**. En la taula 1 mostrem exemples de BD d'aquests models (encara que alguns poden situar-se en més d'una categoria).

Lectura complementària

Susan George (2013, juliol). «NOSQL - NOTONLY SQL» [en línia]. *International Journal of Enterprise Computing and Business Systems* (vol. 2, núm. 2). <http://www.ijecbs.com/July2013/3.pdf>

Taula 1. Exemples de bases de dades dels diferents models de dades

Model	Bases de dades
Columna	Hbase, Cassandra, Accumulo, Hypertable2.
Document	MongoDB, CouchDB, RavenDB, RethinkDB, Terrastore 3.
Graf	Neo4J, AllegroGraph, Infinite Graph, HyperGraphDB 5.

Model	Bases de dades
Valor clau	Dynamo, Riak, Redis, MemcacheDB, Voldemort, Scalaris, BerkeleyDB, SimpleDB 4.
Multimodel	CortexDB, AlchemyDB.

Aquests models de dades tenen les següents característiques:

1) Columna: les dades s'emmagatzemen en cel·les agrupades en columnes de dades en comptes de fileres. Les columnes s'agrupen, lògicament, en famílies de columnes que poden contenir una quantitat pràcticament il·limitada de columnes i que es poden crear en temps d'execució o en la definició de l'esquema.

La lectura i l'escriptura es fa utilitzant columnes en comptes de fileres, i en comparació amb la majoria dels sistemes de bases de dades relacionals emmagatzemen dades en columnes. El benefici d'emmagatzemar dades en columnes és que la cerca i l'accés són ràpids, i l'agregació de dades, summament eficient.

Les bases de dades relacionals emmagatzemen una sola filera com una entrada de disc contínua, i les diverses fileres s'emmagatzemen en diferents llocs del disc, mentre que les bases de dades en columnes emmagatzemen totes les cel·les corresponents a una columna com una entrada de disc contínua, la qual cosa agilita la cerca o l'accés.

Cassandra

Cassandra és capaç de gestionar aplicacions comercials que requereixen escalabilitat massiva, disponibilitat contínua, alt rendiment, seguretat i simplicitat operativa. Empreses com Netflix, Sky, SoundCloud, Healthx, GoDaddy o eBay, entre altres, utilitzen Cassandra.

2) Document: el concepte central d'un magatzem de documents és la noció de *document*, les dades del qual són una col·lecció de parells clau-valor. Les codificacions estàndard comunes es fan en formats XML, YAML i JSON (o també en formes binàries com BSON).

Una diferència important entre un magatzem de valors clau i un magatzem de documents és que aquest últim incorpora metadades d'**atributs associats** amb el contingut emmagatzemat, la qual cosa proporciona essencialment una manera de consultar les dades en funció del contingut.

MongoDB

MongoDB s'utilitza per a intel·ligència operativa i anàlisi en temps real. Diverses empreses han construït la seva *suite* d'internet de les coses (IoT) a MongoDB, la qual cosa ha portat el poder de les dades massives a una nova gamma d'aplicacions d'internet industrial, que inclouen fabricació, automoció, comerç detallista o energia, entre altres.

3) Graf: està dissenyat per a dades les relacions de les quals estan ben representades com un graf, és a dir, elements interconnectats amb un nombre indeterminat de relacions entre elles. Una base de dades de grafs és essencialment una col·lecció de nodes i de connexions.

Cada **node** representa una entitat i cada **connexió**, una relació entre dos nodes. Cada node es defineix mitjançant un identificador únic, un conjunt de connexions sortints o entrants i un conjunt de propietats expressades com a parells clau-valor. Cada connexió està definida per un identificador únic, un node de lloc d'inici o lloc de finalització i un conjunt de propietats.

Aquestes dades poden ser relacions socials, enllaços de transport públic, mapes de carreteres o topologia de xarxes o qualsevol altra relació entre el node d'entrada i el de sortida. Les bases de dades que es basen en grafs són adequades per extreure dades de les xarxes socials i molt útils per treballar amb dades en disciplines que impliquen relacions complexes i esquemes dinàmics, com ara la gestió de la cadena de subministrament, els sistemes biològics i els sistemes de recomanació.

4) Valor clau: és un dels models de dades no trivials més simple en format i sense esquema. La clau pot ser sintètica o autogenerada, mentre que el valor pot ser una cadena de caràcters (*string*), objectes JSON o BLOB (objecte gran bàsic), etc. Generalment empra una taula ***hash*** en la qual hi ha una clau única i un punter a un element de dades en particular.

Hash

Funció també anomenada *resum*, que converteix una entrada de lletres i de números de qualsevol grandària en una sortida única de longitud fixa.

Riak

Riak es pot utilitzar en aplicacions que necessiten clau de sessió en botigues de comerç electrònic; l'usen empreses com Best Buy, Copious, Ideel i Shopzilla.

5) Multimodel: les bases de dades multimodel suporten combinacions dels models abans esmentats o també incorporen el model clàssic SQL. Per exemple, MariaDB és en el seu disseny una BD SQL, però suporta taules NoSQL i emmagatzematge per columnes, essent considerada per molts experts com una BD multimodel.

4.6. Núvol públic i eines per a *big data*

Avui dia, la gran majoria de proveïdors de núvol ofereixen entorns preparats per processar dades massives. Tot seguit citarem alguns exemples (la llista no és exhaustiva):

1) Amazon EMR (https://aws.amazon.com/emr): plataforma de dades massives en el núvol que permet processar grans quantitats de dades utilitzant eines de codi obert, com ara Apache Spark/Hadoop, Apache Hive, Apache HBase, Apache Flink, Apache Hudi i Presto. Amb l'entorn EMR es poden fer anàlisis a escala de petabytes amb un cost un 50 % menor respecte a les solucions locals tradicionals (segons el proveïdor). Per a treballs d'execució curta, es poden ac-

tivar i desactivar clústers i pagar només pels segons que s'han usat les instàncies. Per a càrregues de treball de llarga durada es poden crear clústers d'alta disponibilitat que escalin automàticament per satisfer la demanda.

2) Azure HDInsight (https://azure.microsoft.com/en-us/services/hdinsight): permet executar tasques en entorns de codi obert, com ara Apache Hadoop, Spark, Kafka, Hive, Hbase, Stom i ML. Accepta tasques de diferents tipus, tant empresarials com d'investigació, de manera que manté la rendibilitat i processa sense esforç grans quantitats de dades mitjançant l'ús de les plataformes de codi obert ja esmentades. Amb HDInsight es poden generar ràpidament clústers de dades massives sota demanda, escalar (o reduir) segons les necessitats d'ús i pagar només per allò que s'ha utilitzat. La plataforma garanteix la privacitat de les dades mitjançant una xarxa virtual d'Azure i s'integra amb Azure Active Directory, Data Factory i Data Lake Storage, la qual cosa permet crear canals d'anàlisis completes encriptades.

3) Google Dataproc: és una infraestructura integrada dins de Google Cloud que permet el processament de dades massives i l'anàlisi mitjançant eines de codi obert de manera ràpida, fàcil i segura en el núvol. Igual que la majoria de proveïdors de núvol, té comptes d'ús gratuït, amb un determinat crèdit i temps limitat (a Google són tres-cents dòlars i un període de prova de noranta dies), i també ofereix l'ús gratuït (fins a límits mensuals) d'alguns productes com BigQuery. Dataproc permet crear clústers per executar Apache Spark/Hadoop o Presto, entre altres, i pagar només per ús (segons), amb xifrats per mantenir la confidencialitat de les dades i amb una seguretat unificada integrada en cada clúster, amb escalat automàtic i eliminació de clústers inactius.

Enllaç complementari

Google Dataproc: <https://cloud.google.com/dataproc/>

BigQuery

Magatzem de dades sense servidor totalment administrat que permet una anàlisi de dades escalable en petabytes i funciona com a plataforma com a servei que admet consultes mitjançant ANSI SQL.

4) Alibaba Cloud Elastic MapReduce (EMR) (https://www.alibabacloud.com/products/emapreduce): és una solució de processament de dades massives que s'executa en la plataforma Alibaba Cloud. EMR es basa en instàncies ECS d'Alibaba Cloud i en Apache Hadoop i Apache Spark, la qual cosa permet la utilització dels components de l'ecosistema de Hadoop i d'Spark, com ara Apache Hive, Apache Kafka, Flink, Druid i TensorFlow, entre altres. Amb EMR es poden analitzar i processar dades emmagatzemades en diferents serveis d'emmagatzematge de dades en el núvol d'Alibaba, com Object Storage Service (OSS), Log Service (SLS) i Relational Database Service (RDS).

5) Cloudera (https://www.cloudera.com/downloads.html): proporciona una plataforma de programari per a enginyeria de dades, emmagatzematge de dades i aprenentatge automàtic i anàlisi que s'executa en el núvol o en les instal·lacions pròpies de l'empresa o de la institució. Cloudera va començar com una distribució híbrida de codi obert d'Apache Hadoop (CDH) que facilitava l'ús de Hadoop de classe empresarial, encara que després ha anat incloent

diversos projectes de codi obert amb llicència d'Apache (Apache Spark, Hive, Avro, HBase, etc.) que es combinen per formar la plataforma Apache Hadoop. Cloudera, que també és patrocinadora de l'Apache Software Foundation, ha creat aliances amb altres empreses (Azure, EMC, etc.) o directament les ha comprat, com en el cas de HortonWorks (2019) o d'Arcadia Data, per transformar-se en una de les empreses líders del mercat en solucions *in situ* o en el núvol de plataformes de dades massives.

6) MapR (actualment, HPE Ezmeral Data Fabric) (https://www.hpe.com/us/en/software/data-fabric.html): MapR va ser un altre dels grans pilars per proveir solucions empresarials per al processament de dades massives a partir de programari de codi obert basat en Apache Hadoop i Spark, un sistema d'arxius distribuït, un sistema d'administració de base de dades de múltiples models i el processament de flux d'esdeveniments per a l'anàlisi de dades en temps real. La plataforma MapR s'executa tant en maquinari propietari de les empreses o institucions com en serveis informàtics en el núvol públic. El mes d'agost de 2019, després de les dificultats financeres, la tecnologia i la propietat intel·lectual de l'empresa van ser adquirides per Hewlett Packard Enterprise per conformar la plataforma Ezmeral Data Fabric d'HP.

Atès que es tracta d'un sector molt dinàmic, sorgeixen i desapareixen empreses o són comprades per altres; algunes de les que ofereixen solucions són les següents:

- Domino (https://www.dominodatalab.com/product/domino-data-science-platform/)
- Datameer (https://www.datameer.com)
- GreenPlum (comprada per VMWare recentment i rebatejada com a Tanzu) (https://greenplum.org)
- SAP (https://www.sap.com/index.html)
- IBM (https://www.ibm.com/analytics/hadoop)
- QuBole (https://www.qubole.com/platform/open-data-lake-platform)
- HPCC (https://hpccsystems.com/try-now)
- RapidMiner (https://rapidminer.com/get-started/)
- Teradata (https://www.teradata.com)
- Tableu (https://www.tableau.com/why-tableau)

Tanmateix, l'estudiantat o la persona usuària sempre pot anar a la font i instal·lar l'ecosistema Apache Hadoop des dels repositoris oficials o utilitzar una plataforma creada per provar tota aquesta tecnologia, anomenada *Bigtop*, que inclou (bigtop 1.4.0 stack): alluxio, ambari, apex, bigtop_groovy, bigtop_jsvc, bigtop_tomcat, bigtop_utils, flink, flume, giraph, gpdb, hadoop, hama, hbase, hive, ignite_hadoop, kafka, mahout, oozie, phoenix, qfs, solr, spark, sqoop, sqoop2, tajo, tez, ycsb, zeppelin i zookeeper.

Enllaços complementaris

Repositoris oficials: <https://projects.apache.org/projects.html>

Bigtop: <https://cwiki.apache.org/confluence/ pages/viewpage.action?pageId=70256303>

5. Models d'interacció (API)

Enllaç complementari

En la pàgina web d'IBM trobareu informació sobre què és una API. Podeu consultar el següent enllaç:

<https://www.ibm.com/cloud/learn/api>

Les interfícies de programació d'aplicacions (API, *application programming interface*) són un model d'interacció i d'intercanvi de dades entre aplicacions que simplifiquen el desenvolupament i la innovació de programari, ja que possibiliten que les aplicacions mateixes intercanviïn dades i funcionalitats de manera fàcil i segura (IBM).

Això permet a les empreses obrir les dades i la funcionalitat de les seves aplicacions a desenvolupadors externs, socis comercials o departaments interns dins de l'empresa de manera estandarditzada i segura, la qual cosa fa possible que els diferents serveis es comuniquin entre ells i es reutilitzin les dades dinàmicament mitjançant una interfície documentada. Els desenvolupadors simplement consulten la documentació de l'API i ja poden interactuar amb serveis, un recurs que cada vegada és més emprat (per exemple, moltes de les aplicacions web més populars de l'actualitat no serien possibles sense les API).

Per la seva estructura interna, una API no és res més que un conjunt de regles definides que expliquen com les aplicacions es comuniquen entre elles en un model d'interacció client-servidor.

Funcionament d'una API

Una aplicació web client inicia una crida a l'API per recuperar informació del servidor, indica en la URL els paràmetres de la informació que desitja obtenir i, després de verificar la validesa del servidor, obté la informació sol·licitada i la retorna a l'aplicació client.

Les crides a l'API generalment inclouen **credencials d'autorització** per reduir el risc d'atacs al servidor. L'accés es pot limitar per minimitzar les amenaces a la seguretat; a més, durant l'intercanvi, els encapçalaments HTTP, les galetes (*cookies*) o els paràmetres de la cadena de consulta proporcionen capes de seguretat addicionals a les dades.

Hi ha diversos **protocols** per proporcionar als usuaris un conjunt de regles definides que especifiquen els tipus de dades i les ordres acceptades, entre els quals destaquen:

- **SOAP (*simple object access protocol*)**: és un protocol per generar diferents API que utilitza XML com a format de dades i HTTP/HTTPS com a protocol de transmissió. El seu disseny data del 1998 i és un protocol flexible, ja que pot emprar altres protocols per a la transmissió d'informació si cal. Si bé el seu ús és freqüent, no és el més usat, ja que la dependència d'XML el fa molt estricte, atès que depurar codi XML és complex. Molts programadors prefereixen XML-RPC, anterior a SOAP, o la seva alternativa JSON-RPC. Tant l'un com l'altre usen el mecanismes d'RPC

(<https://en.wikipedia.org/wiki/remote_procedure_call>), però el primer utilitza XML mentre que el segon fa servir JSON per codificar les dades. Aquests protocols són molt més simples i lleugers, motiu pel qual empren menys recursos, però no tenen la versatilitat de SOAP.

- **REST (*representational state transfer*)**: aquest protocol elimina la dependència d'un llenguatge, com XML, i permet la codificació de les dades en múltiples formats, essent JSON el més utilitzat; usa HTTP/HTTPS com a protocol de transmissió. Les API que empra el protocol REST s'anomenen *API RESTful* i tenen una arquitectura client-servidor sense estat; això implica que no s'emmagatzemen dades de clients entre les sol·licituds GET (que seran diferents i estaran desconnectades entre elles), per tant, el programador obté més llibertat en la implementació i la interacció és simple i escalable. El protocol assigna a cada operació una URL única, per la qual cosa el servidor que rep una sol·licitud pot determinar quines instruccions executar per satisfer-la. Si bé les API REST són les més utilitzades avui dia, per a moltes aplicacions les API JSON-RPC, atès que tenen un abast limitat, milloren el rendiment d'una API REST i, per tant, són una alternativa quan les funcionalitats d'aquesta API poden satisfer les necessitats del disseny de l'aplicació.

- **gRPC (*Google remote procedural call*)**: API de codi obert desenvolupada per Google l'any 2015 que també utilitza RPC, amb el gran avantatge que permet als desenvolupadors definir les seves funcions per habilitar la comunicació entre serveis segons calgui. gRPC usa HTTP com a capa de transport i inclou un conjunt interessant de funcions addicionals (autenticació, control de flux, etc).

- **GraphQL**: va ser desenvolupada per Facebook l'any 2015; és un llenguatge de consulta i manipulació de dades per a API i un entorn d'execució per fer consultes de dades. En realitat es tracta d'una API per a entorns web comparable, en certs aspectes, amb REST i altres arquitectures de servei web. Permet als clients definir l'estructura de dades requerida, i la mateixa estructura de dades serà retornada pel servidor, encara que això té implicacions, segons molts experts, en l'eficiència en la memòria cau web dels resultats d'aquestes consultes. La flexibilitat i la varietat del llenguatge de consulta hi afegeix complexitat, i pot no resultar atractiva en comparació amb altres API com JSON-RPC o la mateixa REST.

- **Apache Thrift**: va ser desenvolupada per Facebook amb l'objectiu d'habilitar la comunicació amb serveis escrits en diferents llenguatges i l'escalabilitat.[3] Thrift és un llenguatge de definició d'interfície i un protocol de comunicació utilitzat per definir i crear serveis en diversos llenguatges de programació. En essència, és una implementació d'un *framework* RPC que usa un motor de generació de codi combinat amb una pila de programari (*software stack*) per habilitar una API, en la que la pila ajuda a escriure codi per definir la banda del client i del servidor. La sintaxi del

(3) Actualment és un entorn de codi obert gestionat per la fundació Apache: <https://thrift.apache.org>.

codi (en arxius Thrift) és flexible i intuïtiva, i serà posteriorment l'entrada al motor de generació que crearà el codi requerit en qualsevol llenguatge de programació especificat pel desenvolupador.

Per exemple, interactuar amb l'API de GitHub és summament fàcil mitjançant l'ordre `curl`:

```
curl https://api.github.com/users/rsuppi
```

Que ens retornarà:

```
{
  "login": "rsuppi",
  "id": ............,
  "node_id": ".................",
  "avatar_url": "https://avatars.githubusercontent.com/u/.............?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/rsuppi",
  "html_url": "https://github.com/rsuppi",
  "followers_url": "https://api.github.com/users/rsuppi/followers",
  "following_url": "https://api.github.com/users/rsuppi/following{/other_user}",
  "gists_url": "https://api.github.com/users/rsuppi/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/rsuppi/starred{/owner}{/repo}",
  "subscriptions_url": "https://api.github.com/users/rsuppi/subscriptions",
  "organizations_url": "https://api.github.com/users/rsuppi/orgs",
  "repos_url": "https://api.github.com/users/rsuppi/repos",
  "events_url": "https://api.github.com/users/rsuppi/events{/privacy}",
  "received_events_url": "https://api.github.com/users/rsuppi/received_events",
  "type": "User",
  "site_admin": false,
  "name": null,
  "company": null,
  "blog": "",
  "location": null,
  "email": null,
  "hireable": null,
  "bio": null,
  "twitter_username": null,
  "public_repos": 5,
  "public_gists": 0,
  "followers": 0,
  "following": 0,
  "created_at": "2016-12-15T12:29:58Z",
  "updated_at": "2021-02-10T08:59:21Z"
```

```
}
```

Aquest és el resultat de tota la informació que, sense autenticar, es pot extreure de l'usuari rsuppi.

Enllaços complementaris

API de GitHub: <https://docs.github.com/en/rest/guides/ getting-started-with-the-rest-api>

Introducció a `curl` usant l'API de GitHub: <https://gist.github.com/btoone/2288960>

Bibliografia

Bibliografia bàsica

Fahad Akhtar, Syed Muhammad (2018). *Big Data Architect's Handbook: A guide to building proficiency in tools and systems used by leading big data experts*. Birmingham: Packt Publishing.

Gupta, Sumit; Saxena, Shilpi (2016). *Real-Time Big Data Analytics*. Birmingham: Packt Publishing.

Salvador, Jaime; Ruiz, Zoila; García-Rodríguez, José (2017). «Big Data Infrastructure: A Survey» [en línia]. A: José Manuel Ferrández Vicente; José Ramón Álvarez-Sánchez; Félix de la Paz López; Javier Toledo Moreo; Hojjat Adeli (ed.). *Biomedical Applications Based on Natural and Artificial Computing* (pàg. 249-258). International Work-Conference on the Interplay Between Natural and Artificial Computation. La Corunya, 19-23 de juny. <https://link.springer.com/chapter/10.1007%2F978-3-319-59773-7_26> i <http://www.dspace.uce.edu.ec/bitstream/25000/13653/1/Big%20data%20infrastructure%20a%20survey.pdf>

Referències bibliogràfiques

Burt, Jeff (2014, 29 de gener). «Cisco Moving Apps to the Network Edge for Internet of Things» [en línia]. *eWeek*. <https://www.eweek.com/networking/cisco-moving-apps-to-the-network-edge-for-internet-of-things>

El Economista (2014, 28 de maig). «Data Center de Visa puede procesar 47,000 transacciones por segundo» [en línia]. *El Economista*. Mèxic. <https://www.eleconomista.com.mx/tecnologia/Data-Center-de-Visa-puede-procesar-47000-transacciones-por-segundo-20140528-0067.html>

Fisher, David Edward; Yang, Shuhui (2016). «Doing More with the Dew: A New Approach to Cloud-Dew Architecture» [en línia]. *Open Journal of Cloud Computing* (vol. 3, núm. 1). <http://www.ronpub.com/publications/OJCC_2016v3i1n02_Fisher.pdf>

Fòrum Econòmic Mundial (2012). *Big Data, Big Impact: New Possibilities for International Development* [en línia]. <https://www3.weforum.org/docs/WEF_TC_MFS_BigDataBigImpact_Briefing_2012.pdf>

Foster, Ian; Kesselman, Carl (2014). *The History of the Grid* [en línia]. <http://www.ianfoster.org/wordpress/wp-content/uploads/2014/01/History-of-the-Grid-numbered.pdf>

García Lopez, Pedro i altres (2015, octubre). «Edge-centric Computing: Vision and Challenges» [en línia]. *ACM SIGCOMM Computer Communication Review* (vol. 45, núm. 5, pàg. 37-42). <https://doi.org/10.1145/2831347.2831354>

George, Susan (2013, juliol). «NOSQL - NOTONLY SQL» [en línia]. *International Journal of Enterprise Computing and Business Systems* (vol. 2, núm. 2). <http://www.ijecbs.com/July2013/3.pdf>

Gilder, George (2013). *Knowledge and Power: The Information Theory of Capitalism and How It is Revolutionizing Our World*. Washington, D. C.: Regnery Gateway.

Grishchenko, Alexey (2015, 13 de juliol). «Hadoop vs MPP» [en línia]. *Distributed Systems Architecture*. <https://0x0fff.com/hadoop-vs-mpp>

Gupta, Sushant (2020, 2 de setembre). «What is big data analytics? Beginner guide to the world of big data» [en línia]. *Tricky Enough*. <https://www.trickyenough.com/big-data-analytics>

Haefele, Paul (2012). «EC2 is 380% more expensive than internal cluster» [en línia]. *Deep Value*. <http://deepvalue.net/ec2-is-380-more-expensive-than-internal-cluster>

Jain, Anil (2016, 17 de setembre). «The 5 V's of big data» [en línia]. *IBM. Watson Health Perspectives*. <https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data>

Khan, Atta ur Rehman; Othman, Mazliza; Madani, Sajjad Ahmad; Khan, Samee Ullah (2014). «A Survey of Mobile Cloud Computing Application Models» [en línia]. *IEEE Communications Surveys & Tutorials* (vol. 16, núm. 1). <http://ieeexplore.ieee.org/document/6553297>

Mell, Peter; Grance, Timothy (2011, setembre). «The NIST Definition of Cloud Computing» [en línia]. *National Institute of Standards and Technology*. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

Metz, Cade (2011, 18 d'octubre). «How Yahoo Spawned Hadoop, the Future of Big Data» [en línia]. *Wired*. <https://www.wired.com/2011/10/how-yahoo-spawned-hadoop>

O'Reilly Media (2012, 19 de gener). «Volume, Velocity, Variety: What You Need to Know About Big Data» [en línia]. *Forbes*. <https://www.forbes.com/sites/oreillymedia/2012/01/19/volume-velocity-variety-what-you-need-to-know-about-big-data>

Sarmenta, Luis F. G. (2001). *Volunteer computing* [en línia]. Cambridge: Massachusetts Institute of Technology. <http://people.csail.mit.edu/lfgs/papers/sarmenta-phd-mit2001.pdf>

Schollmeier, Rüdiger (2001). «A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications» [en línia]. *Proceedings First International Conference on Peer-to-Peer Computing* (pàg. 0101). <https://doi.org/10.1109/P2P.2001.990434>