

CPD - Assaig i Presentació

14: Virtualització

Albert Bernal i Blai Ortuño

08/04/2021

ÍNDEX

1.- Introducció	2
2.- Història	3
2.1.- Primeres passes	3
2.2.- El canvi cap a un model comercial	4
2.3.- Java, VMware i el nou auge de la virtualització	6
3.- Definició	9
3.1.- Virtualització de <i>hardware</i>	9
3.2.- Virtualització de <i>software</i>	11
3.3.- Virtualització d'emmagatzematge	13
4.- Exemples	14
4.1.- VMware	14
4.1.1.- VMware en CPDs	14
4.2.- Microsoft Azure	18
4.3.- Xen	20
4.4.- KVM	22
5.- Conclusions	24
6.- Bibliografia	25

1.- Introducció

Avui dia i des de fa ja uns segles, utilitzar els recursos disponibles de la manera més eficient possible és un dels principals objectius per a qualsevol indústria. No és estrany, per tant, que en un món tan competitiu com el de la tecnologia, aquesta necessitat es vegi incrementada de manera substancial. Donada la seva importància, existeixen infinitat de drecceres que podem aprofitar per a millorar la productivitat del nostre sistema. En aquest treball, però, ens centrarem només en una de molt útil i interessant per al món de la computació: la virtualització.

Com totes les bones idees, neixen per a solucionar un problema. En aquest cas, la virtualització proposa una nova manera d'augmentar el percentatge d'utilització dels recursos de processament d'un servidor, molt baix degut a que cadascun només estava dedicat a una funcionalitat i, quan aquesta no estava sent executada, el servei estava inactiu. Bàsicament, la virtualització segmenta un servidor físic en n lògics, que funcionen en paral·lel i permeten aprofitar molt millor els components.

Aquesta idea mirada des d'una altra perspectiva permet també resoldre un problema potser de més rellevància, executar codi sense importar els dispositius físics específics que l'han d'interpretar. En el cas anterior, les màquines virtuals es troben en una capa superior a l'arquitectura del sistema, ja que no es comuniquen directament amb el *hardware*, sinó amb un sistema operatiu intermig destinat a administrar-les i controlar el seu ús dels recursos. Tenir només un interlocutor amb els components físics ens aporta estabilitat i suport per a aquest i, sobretot, portabilitat per al que s'executa per sobre, les màquines virtuals, que no hauran de saber comunicar-se amb el *hardware* específic i no dependran d'ell.

En aquest treball tenim la intenció de veure la història i el funcionament de la virtualització, exemples reals d'aquesta i reflexionar sobre el seu ús en centres de processament de dades, així com el que poden oferir per a un usuari individual.

2.- Història

2.1.- Primeres passes

A simple vista, el concepte de virtualització pot semblar que pertany a un passat relativament pròxim, potser als anys 90, sobretot si pensem en el fet que els serveis tan populars avui dia com Amazon Web Services o Google Cloud no van aparèixer fins a la dècada dels 2000, el 2006 i 2008, respectivament. Sorprenentment, aquest no és el cas: la idea de compartir els recursos d'una sola màquina entre diferents usuaris és present en el món de la computació des dels anys 60.

En aquella època, les poques empreses o institucions que aconseguien fer-se amb un ordinador ho feien mitjançant inversions enormes, les quals s'havia de fer tot el possible per a rendibilitzar. S'havia de buscar la màxima eficiència i, per tant, la màquina havia d'estar el màxim temps funcionant. A més, cada ordinador era compartit per centenars de persones dins l'organització, majoritàriament científics, i processar els seus programes en el menor temps possible significava un progrés més ràpid per als seus estudis.

Les tasques que feien les màquines d'aquella època eren principalment dues: entrada/sortida i processament de dades. El procés d'ús era, bàsicament, que un científic desenvolupava un programa i el portava al tècnic que s'encarregava d'administrar l'ordinador, que l'executava i retornava els resultats. En un principi, si fem una cua de científics i no perdem temps entre tasques, el percentatge d'utilització de la màquina seria del 100%. Durant uns anys es va creure així, fins que es

van adonar que les dues funcions es podien fer en paral·lel: mentre el programa d'un científic s'executava, les dades d'entrada del següent se n'anaven introduint. Per primer cop, una màquina era utilitzada simultàniament per dos usuaris, introduint el terme de “màquina de temps compartit”.



Figura 1. L'ENIAC, considerada la primera computadora d'ús general.

Havia nascut una nova idea, que va crear noves necessitats en el món de la investigació. El primer client d'un sistema amb aquestes capacitats va ser el Massachusetts Institute of Technology (MIT), que estava desenvolupant un projecte ambiciós sobre computació que requeria de nou *hardware* més potent. L'empresa que va acceptar la tasca de fabricar-lo va ser General Electric (GE), ja que International Business Machines (IBM) s'hi va negar, per por de destinar massa recursos en un producte que encara no tenia demanda suficient. Uns mesos després, adonant-se que havien perdut un client tan important com l'MIT i que potser sí es podria traduir en una oportunitat de mercat en el futur, el 7 d'abril de 1964 va anunciar el seu intent a una màquina pensada per a la virtualització a servidors, la IBM S/360 Model 40 que utilitzava el sistema operatiu CP-40¹.

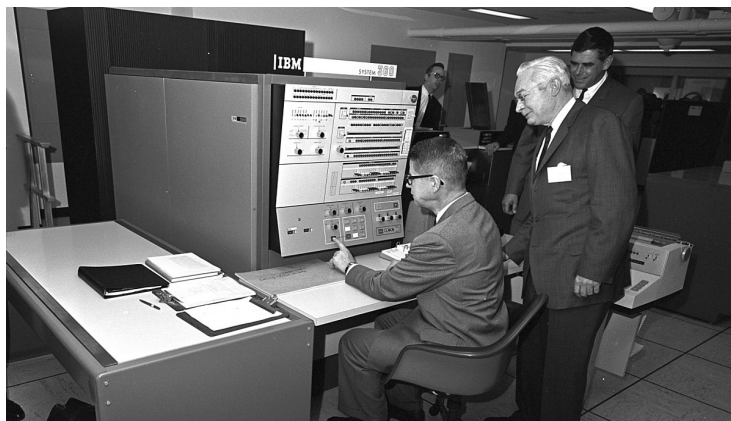


Figura 2. Sistema S/360 Model 40 instal·lat al Departament d'Agricultura dels Estats Units.

2.2.- El canvi cap a un model comercial

Les empreses tenen molts objectius diferents, però sempre n'hi ha un en el qual totes es posen d'acord: guanyar diners. Potser l'equip d'enginyers que desenvolupaven el CP-40 a la ciutat de Cambridge, Massachusetts, ho feien per passió a la tecnologia, però a IBM el que manava eren els dòlars. Fins ara, les computadores que fabricaven tenien com a destí certs clients que els hi demanaven, model de negoci que ara es quedava petit. Hi havia una demanda creixent per part d'altres empreses, i per això van produir el S/360 Model 67 amb CP-67, el primer que estaria disponible a qui pogués pagar, sense restriccions al procés de subhasta. Aquest producte comptava amb un *software* que permetia executar

¹ CP és l'acrònim de *Control Program*, un codi similar a la idea actual de sistema operatiu, és a dir, que controlava el funcionament de la màquina. Per a evitar confusió, en aquest treball s'utilitza sempre el terme "sistema operatiu" i no "programa de control".

múltiples aplicacions alhora gràcies a un sistema de particions, i és considerada la primera màquina comercial amb un sistema operatiu que suportés la virtualització.

El seu funcionament també va ser revolucionari perquè va ser el primer cop que IBM introduïa interacció entre usuari i computadora en temps d'execució en un producte comercial. El codi que van desenvolupar constava de dues parts, CP, el programa (també anomenat *hypervisor*) que creava les màquines virtuals, i CMS (*Console Monitor System*), el que executaven les màquines amb les quals l'usuari podia interactuar.

Vista la perspectiva d'IBM, tornem a l'altre competidor a la cursa de la computació americana, GE, els qui prèviament hem mencionat que van desenvolupar una computadora per a un projecte de l'MIT. Aquest era conegut com a *Project MAC (Mathematics and Computation)* i comptava també amb la col·laboració dels Laboratoris Bell (ara AT&T) i GE. Una de les seves branques va ser la creació d'un sistema operatiu que permetés aprofitar millor el temps anomenat Multics (*Multiplexed Information and Computing Service*). Tot i que aquest sistema tenia característiques innovadores, també presentava problemes estructurals complexos que van fer que la majoria d'investigadors deixessin el projecte. Els que s'hi van quedar, però, es van decidir a començar de nou en un treball de menor escala. Aquests eren Ken Thompson, Dennis Ritchie, Douglas McIlroy, i Joe Ossanna, que van desenvolupar en llenguatge



C un sistema anomenat irònicament Unics (*Uniplexed Information and Computing Service*), el qual més tard va canviar de nom a Unix, com es coneix actualment. Com que estava escrit en C i no en *assembler*, era molt portable, i només calia recompilar el sistema per a poder utilitzar-lo en un nou *hardware*.

Figura 3. Ken Thompson (esquerra) i Dennis Ritchie (dreta).

Aquest avantatge, juntament amb l'abaratiment dels ordinadors personals als anys 80, van portar a una època de poc progrés a la virtualització, ja que les empreses van substituir els terminals existents, que enviaven comandes al *main frame* per a que fes el processament, per PCs individuals. No va ser fins als anys 90, amb l'aparició de Java i VMware, que el món de les màquines virtuals va tornar a un paper protagonista a la computació mundial.

2.3.- Java, VMware i el nou auge de la virtualització

Java va ser creat a principis dels anys 90 per un grup d'enginyers de *Sun Microsystems* que creien que el mètode d'utilitzar Unix i anar recompilant amb C i C++ no era suficientment portable. Després d'uns anys de desenvolupament, el 1995 Java va sortir al mercat, proposant un nou nivell de facilitat a l'hora d'executar aplicacions sense importar el *hardware* del *host*. El seu funcionament era el següent: un programador escrivia en Java el seu codi font utilitzant JDK (*Java Development Kit*) i l'usuari l'executava a la seva màquina utilitzant una aplicació anomenada JRE (*Java Run-time Environment*).

A diferència de les propostes anteriors, el programa es compilava a *Java Byte Code*, un llenguatge que posteriorment interpretaria i compilaria definitivament el JRE just abans de l'execució. Aquest concepte es coneix com a *Just In Time compilation*, i permet al programador oblidar-se del *hardware* específic en el moment d'escriure el codi, i a l'usuari de compilar manualment ja que es farà de forma automàtica.

Per tant, el JRE era poc més que una capa d'abstracció, una màquina virtual que s'executava per sobre del sistema operatiu del *host*. En conclusió, el que havia de tenir suport per a la majoria de *hardware* era el JRE i no cadascuna de les aplicacions que es desenvolupaven, fet que facilitava molt el manteniment i feia disponibles els programes per a tothom, es tingués la màquina que es tingués. Aquest nou concepte, juntament amb l'auge d'Internet, va ser el que va fer ressuscitar la virtualització i portar-la a un nou nivell, primer per a ordinadors individuals i més tard en servidors.

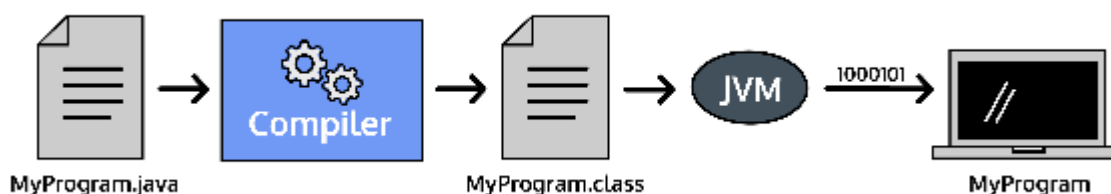


Figura 4. Pipeline de la compilació de Java, Codecademy.

Amb el pas del temps i l'augment en les capacitats de processament dels ordinadors personals, van permetre la producció de *software* per a tenir màquines virtuals com les coneixem avui dia, executant al PC de casa un sistema operatiu sobre un altre. La idea va agafar força quan va sortir al mercat un emulador d'x86 anomenat SoftPC va permetre als usuaris estalviar diners mitjançant virtualització: als anys 90, un ordinador que pogués executar MS-DOS, el sistema operatiu de Microsoft, costava més de 1.000 dòlars de l'època, mentre que un amb Unix podia estar per sota de 500. Si s'aconseguia utilitzar les

aplicacions pròpies del primer SO al segon, podries fer servir el mateix programari amb un 70% de descompte. Aquest producte va continuar evolucionant, afegint suport per a ordinadors Mac i permetent executar aplicacions de Windows, fets que el van impulsar al mercat i que van fer que companyies com Apple desenvolupessin els seus programaris de virtualització d'aplicacions, en aquest cas sota el nom de Virtual PC.

Aquesta nova manera de pensar no només va significar un canvi d'estratègia en algunes companyies, sinó que en va crear de noves. Va ser el 1998 quan va aparèixer un nou jugador que, a dia d'avui, es troba al cim del mercat: VMware, en la qual ens centrarem per a explicar el progrés de la virtualització al segle XXI ja que, tot i que hi ha altres protagonistes en aquest mercat tan competitiu, la seva història és força representativa i no creiem necessari allargar massa aquest tòpic.

Un any més tard, el 1999, van presentar la seva proposta, un programa anomenat VMware Workstation. Tot i que trigava uns 30 minuts només en posar en marxa la màquina virtual, permetia, per primer cop, no només executar aplicacions d'un SO en un altre mitjançant emuladors, sinó que era capaç de reproduir completament el sistema operatiu. A més, va estar disponible de sortida per a Windows i Linux, fet poc usual en aquells temps. El producte va ser molt ben rebut i va arribar a 1 milió de dòlars en vendes, sobretot gràcies a 2.260 universitats que van comprar-lo.

Tot i haver fet una important passa endavant amb aquesta idea, no ha estat la que ha portat la companyia a ser un referent a la indústria i a comptar amb prop de 25.000 treballadors a dia d'avui. El seu gran impulsor va ser un que no estava enfocat a un ordinador particular, sinó a servidors: el *hypervisor*. En realitat, eren dos productes enfocats a servidors. Un s'anomenava ESX, *Elastic Sky X*, i l'altre GSX, *Ground Storm X*. La diferència entre ells era que ESX era un producte basat en Linux que portava el seu propi entorn operatiu mentre que GSX feia servir Windows. Aquests presentaven una nova forma d'augmentar el percentatge d'utilització dels servidors. En aquell moment, s'instal·lava físicament un servidor per a cada funcionalitat que oferia el teu sistema, i per tant cadascun estava infrautilitzat, al voltant de l'1 i el 5%. Salta a la vista que aquests valors són pessims i s'estaven malbaratant recursos. Els nous productes de VMware permetien ajuntar funcionalitats en el mateix servidor, dividint-lo de forma lògica i no física. ESX i GSX van coexistir durant uns anys, fins que, amb una altra aplicació anomenada vMotion que servia per a distribuir la càrrega entre servidors, es van

ajuntar per a simplificar el sistema en un producte anomenat vCenter. Això va permetre a VMware tenir una oferta menys segmentada i més simple i fàcil d'utilitzar.

Com passa amb totes les empreses d'èxit, arriba un moment en què les fronteres no són res més que una barrera que les impedeix guanyar més diners. Al 2003, VMware crea les seves primeres oficines a l'estranger, concretament a Frimley, Regne Unit. Només un any després, adonant-se del potencial que tenia la companyia, *EMC corporation* la compra per 635 milions de dòlars. Durant anys, la seva expansió no s'atura, seguint creant noves infraestructures a mercats emergents com l'asiàtic. El progrés tecnològic no ha desaparegut, però passa a un segon pla en les prioritats de l'empresa. L'any 2007 VMware passa a cotitzar en borsa, fet que accentua el seu creixement, superant el valor de 1.000 milions de dòlars en beneficis.



Figura 5. VMware comença a cotitzar en borsa, VMware.

L'any 2013 tornen a presentar un producte revolucionari, sobretot en seguretat: NSX. El seu llançament comporta un model nou de creació de xarxes. En les paraules de VMware, “NSX Data Center és una plataforma completa de virtualització de xarxes *Layer 2 - Layer 7* i seguretat que aporta l'experiència del núvol públic al núvol privat i us permet gestionar tota la xarxa des d'un sol panell de vidre”. Bàsicament, virtualització de xarxes i administració d'aquestes.

El capítol més important de la seva història recent succeeix al 2015, quan VMware forma part de la compra més gran en el sector tecnològic, quan Dell compra *EMC corporation* i la seva participació a VMware per 67.000 milions de dòlars. Per últim, afegir que al 2016 es presenta un pacte amb Amazon, VMware Cloud on AWS, que aporta integració a AWS de les eines i els recursos de VMware, obrint la porta a un futur on hi hagi més llibertat per als clients d'operar a través de diferents núvols sense restriccions.

3.- Definició

La virtualització es pot definir com una tecnologia capaç de simular funcionalitats lligades al *hardware* en serveis de TI com servidors, emmagatzematge, xarxes... sense la qual el concepte de *cloud* que tenim avui dia no hagués sigut el mateix. Nosaltres ens quedarem amb una definició que s'apropa més al que estem treballant. Definirem virtualització com una tècnica per crear un entorn virtual enlloc d'un entorn físic, permetent "particionar" un dispositiu físic com un ordinador o un servidor en diverses màquines virtuals.

De virtualització en tenim de diferents tipus dels quals nosaltres en tractarem tres, els que pensem que són més útils en un centre de processament de dades.

3.1.- Virtualització de *hardware*

Fa referència a les tecnologies que ens permeten posar a disposició components de *hardware* a partir d'un *software* amb independència del seu suport físic, de manera que l'usuari només veu la part lògica i no en sap res de la part física. L'exemple més típic de la virtualització per *hardware* és la màquina virtual, un ordinador virtual que es comporta igual que un ordinador físic tant en el que respecte al *hardware* com al *software*.

Ara bé, per tal d'entendre com funciona, primer cal parlar sobre el *hypervisor*. El podríem definir com un *software* que s'encarrega de dividir els recursos físics dels quals disposem entre els dispositius virtuals, de tal manera que tots els dispositius virtuals tenen el que necessiten per a complir les seves funcions. Aquesta idea es pot veure clarament en la Figura 6, on podem entendre el *hypervisor* com un intermediari entre els dispositius físics i els lògics.

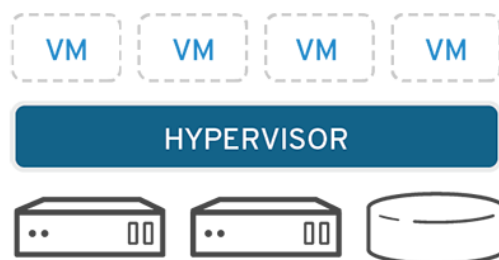


Figura 6. Exemple de la interacció entre *hypervisor*, recursos físics i màquines virtuals

Els usuaris d'aquestes màquines virtuals emetran les ordres o instruccions dins de la màquina, i en cas que alguna d'aquestes instruccions necessités més recursos físics, el *hypervisor* transmetrà la sol·licitud al sistema físic i guardarà els canvis en la memòria cache. Cal mencionar que tot aquest procés es realitza pràcticament a la mateixa velocitat que si es fes en la pròpia màquina física i per tant obtenim diverses màquines lògiques quasi igual de ràpides que l'original a canvi d'una física, això si generant una despesa de recursos més gran, el principal problema que té la virtualització per *hardware* respecte a altres opcions avui dia.

D'aquesta manera cada una d'aquestes màquines virtuals poden actuar de forma independent, de forma que podem executar aplicacions, o fins i tot sistemes operatius en diversos dispositius virtuals a la vegada. Aquest fet ens permetrà millorar moltíssim l'escalabilitat i les càrregues de treball, així com també els costos en manteniment, infraestructura, refrigeració, etc. A més reduïm l'empremta ecològica, fet molt important avui dia si volem seguir tenint un món on crear més CPDs.

Tot i això, la virtualització no només aporta eficiència i reducció dels costos, sinó que també ens proporciona confiança i seguretat com podria fer un dispositiu físic. Degut a que les màquines virtuals estan aïllades unes de les altres, en cas d'haver-hi un forat de seguretat en alguna d'elles no afectaria a la resta. A més, també permeten recuperar *backups* de manera fàcil usant instantànies de màquines virtuals de servidors existents, així com també migrar dades d'aquesta VM (*virtual machine*) a algun altre dispositiu en pocs minuts, proporcionant una fiabilitat elevada davant els possibles errors o problemes en les instal·lacions.

Aquest procés de gestió de recursos realitzat amb el *hypervisor* es pot fer de dues maneres. Primerament, trobem la virtualització completa, on el *hypervisor* simula un entorn sencer de *hardware* per a cada VM. Per tant, cada VM tindrà els seus propis recursos de *hardware* virtuals assignats pel *hypervisor* i executarà aplicacions sobre aquesta base, de manera que el sistema de la VM no té accés al *hardware* del sistema físic. Alguns exemples són Oracle Virtualbox o VMWare Workstation, ambdues molt utilitzades avui dia.

A continuació tenim l'altra alternativa: la paravirtualització. Mentre que en la virtualització completa disposem d'un entorn complet per a cada VM, en aquest cas el *hypervisor* només oferirà una API a través de la qual els sistemes de la VM podran accedir al *hardware* del sistema físic. Com a requisit s'exigirà que aquesta API integri el kernel del sistema operatiu usat pel *host* (*hardware* físic que permet

virtualitzar), és a dir, només podrem paravirtualitzar sistemes *host* modificats. Alguns exemples d'us de paravirtualització són Xen i Oracle Vm Server for Sparc.

Cal mencionar que avui dia també s'està utilitzant la virtualització per *hardware* per altres usos diferents de les VM, com per exemple la *server consolidation*, una tècnica per aconseguir un entorn de treball més eficient a partir de combinar servidors o reemplaçar-los per sistemes virtuals com les VM, el núvol, etc. Bàsicament, és la mateixa idea que hem explicat abans: enlloc de tenir 4 servidors físics, tindrem 1 únic servidor més potent que virtualitzarà 4 servidors de manera lògica. Això ens permetrà aprofitar millor el *hardware* del servidor així com també reduir els costos de manteniment, refrigeració, etc.

3.2.- Virtualització de *software*

Tipus de virtualització on virtualitzem els components de *software* i no els de *hardware*. Hi ha tres tipus principals:

Virtualització d'aplicacions: Consisteix en aïllar la component lògica de les aplicacions de la component del sistema operatiu. El seu objectiu és aconseguir que les aplicacions puguin funcionar independentment del sistema operatiu o entorn de treball on s'executin. D'aquesta manera es poden eliminar els típics problemes de compatibilitat entre aplicacions, podent per exemple executar una aplicació de Windows en un sistema Unix.

Això si, la virtualització d'aplicacions no virtualitza la presentació, és a dir, l'usuari segueix interactuant directament amb l'ordinador on s'està virtualitzant l'aplicació. Es crea un entorn específic d'execució per a cada instància d'aplicació, i qualsevol recurs que l'aplicació necessiti estarà disponible només per a ella. Per tant, a diferència d'altres tipus de virtualització com la de *hardware* on es busca resoldre problemes de compatibilitat d'aplicacions amb sistemes operatius, en la d'aplicacions buscarem solucionar problemes d'incompatibilitat entre aplicacions que es volen executar en el mateix entorn *hardware* i de sistema operatiu.

Virtualització d'escriptori: Permet als usuaris l'accés remot als seus escriptoris emmagatzemant i executant tots els seus programes, aplicacions i dades de forma central. Per tant, cada qualsevol usuari

podrà accedir al seu usuari des de qualsevol dispositiu que tingui les eines necessàries per a connectar-se remotament a l'escriptori, ja sigui un smartphone, tablet, ordinador, etc.

Aquesta virtualització pot ser de diferents tipus, però nosaltres només explicarem el cas més comú, les màquines virtuals basades en *host*. En aquest cas cada usuari es connecta al servidor a través d'un dispositiu client amb la seva pròpia VM. Aquesta virtualització podrà ser persistent si els usuaris sempre es connecten amb la mateixa VM, o no persistent si aquestes VM s'assignen de manera aleatòria. Això sí, independentment de quina opció esculli, l'usuari sempre accedirà al seu escriptori, i per tant podrà personalitzar-ho com desitgi.

Aquest tipus de virtualització pot ser molt útil per a organitzacions que vulguin una administració centralitzada. A més, també es redueixen els costos de gestió, manteniment i de les llicències de *software* que ja no hem de comprar. Tot i això, no tot són avantatges. Per a poder realitzar tantes virtualitzacions d'escriptori ens caldrà invertir una quantitat molt gran de recursos per part del servidor, i necessitarem per tant un servidor potent juntament amb un ample de banda elevat, de manera que aquest tipus de virtualització està pensada per a ser a llarg termini i així rendibilitzar els recursos utilitzats.

Virtualització de sistema operatiu: Permet executar de manera paral·lela instàncies d'espai d'usuari aïllades unes de les altres. No hem de confondre aquesta virtualització amb la de *hardware* on simulem un sistema operatiu sencer amb kernel inclòs. En aquest cas tenim aplicacions virtualitzades que comparteixen el nucli del sistema *host* a nivell de sistema operatiu. Cada una d'aquestes instàncies d'usuari tindrà un entorn de temps d'execució virtual tancat, que normalment s'anomena contenidor.

Aquest contenidor inclou una aplicació amb totes les dependències, com per exemple biblioteques o arxius de configuració necessaris per al correcte funcionament. El principal avantatge d'aquestes aplicacions és que es poden moure sense problemes d'un sistema a un altre, i és per aquest motiu que estan tenint un èxit brutal avui dia. Un bon exemple és Docker, possiblement una de les tecnologies amb més utilització avui dia, que és bàsicament la cara visible de la tecnologia de contenidors.

Cal mencionar que aquests contenidors poden ser gestionats i organitzats mitjançant altres eines com Kubernetes. Aquestes eines permetran tenir el que podríem definir com una “plataforma” de

contenidors, on es pot fer quasi qualsevol cosa amb ells, des de clonar-los fins a dividir-los jeràrquicament en diferents grups.

3.3.- Virtualització d'emmagatzematge

Virtualització que té com a principal objectiu la representació virtual dels recursos d'emmagatzematge d'una empresa, com per exemple un disc dur, una cinta, etc. Es tracta de fer disponibles un grup de recursos com 3 discs durs com a un grup interrelacionat. Per a poder realitzar-ho, trobem una capa d'abstracció entre els dispositius d'emmagatzematge físics i el nivell lògic, de manera que aquests recursos es poden gestionar de manera centralitzada a partir d'un *software* especialitzat.

Tot i això, encara que les dades s'hagin emmagatzemat mitjançant virtualització, els usuaris podran mantenir la ruta als arxius que ja tenien fins i tot si s'ha modificat l'espai físic d'emmagatzematge. Això és gràcies a una taula de mapeig que es gestiona pel *software* especialitzat que hem mencionat anteriorment.

Normalment, en l'àmbit empresarial s'utilitza la virtualització d'emmagatzematge basada en blocs, on les dades es reparteixen en blocs d'un mateix mida. Cada bloc té una direcció única que el *software* especialitzat emmagatzema en la taula de mapeig, i per tant aquesta taula conté totes les metadades necessàries per a trobar l'espai d'emmagatzematge físic dels blocs de dades. Gràcies a aquesta taula som capaços de gestionar les dades virtuals sense interferir amb el controlador del medi físic, i per tant podem moure, duplicar o reduir dades sense que afecti el medi físic.

4.- Exemples

4.1.- VMware

Si estem parlant sobre virtualització, és imperatiu parlar sobre VMware, considerada una de les empreses pioneres en aquest camp, i que avui dia segueix estant al peu del canó, arribant fins i tot a pactar al 2016 amb Amazon per ajuntar les eines disponibles en VMware amb AWS, millorant així la qualitat de vida dels enginyers de TI.

Com s'ha explicat a l'apartat d'història, aquesta empresa fundada l'any 1998, va presentar el seu primer producte l'any 1999, VMware Workstation 1.0, que consistia en una plataforma que permetia executar diversos sistemes operatius com a màquines virtuals en una sola màquina. Era el primer cop que es presentava un producte capaç de fer això en una arquitectura x86 i va ser tota una revolució.

Això va permetre a l'empresa créixer i seguir investigant, aconseguint crear un dels primers sinó el primer *hypervisor* en x86 l'any 2002. A més, durant els seus 20 anys d'existència, VMware ha aconseguit tenir milers de patents en aquest àmbit, sent per tant un dels líders en el sector.

Els anys van passar i VMware Workstation es troba a la versió 16, tenint moltíssimes *features* diferents, com per exemple clonar màquines virtuals, utilitzar una API Rest per automatitzar la creació i el control d'aquestes, etc. Tot i això, Workstation no és el seu únic *software* disponible, i per tant ara en parlarem una mica de què ens ofereixen i què ens pot interessar si som un CPD.

4.1.1.- VMware en CPDs

En el cas de VMware, tenim moltíssims serveis diferents que ens poden servir per a suplir diverses necessitats, des de un entorn de *kubernetes* ja muntat fins a serveis de *cloud* treballant amb AWS. En el nostre cas hem escollit alguns que pensem que poden resultar molt útils si treballem per a un CPD, també pensant en com ajuntar aquests serveis amb la infraestructura que normalment tenim en un centre.

El servei que més ens ha agradat és el servei conjunt del *cloud* que ofereix amb AWS. Aquest servei proporciona entre altres coses el que podríem considerar com una “ruta directa” cap al núvol, ja que és el propi Amazon qui té els servidors i proporciona aquesta connexió. A més, també permet la utilització d’escriptoris i aplicacions virtuals remotes amb seguretat garantitzada, gràcies també a un altre servei anomenat vSphere.

vSphere és una plataforma de virtualització de recursos que s'utilitza entre altres coses per a virtualitzar servidors o fins i tot CPDs, bàsicament a partir de virtualitzar sistemes operatius i *hardware*. Està dividit en dues parts bastant diferenciades, VMware ESXi i VMware vCenter Server.

VMware ESXi és el sistema operatiu que s’encarrega del servidor físic que executarà les màquines virtuals, també conegut com *hypervisor*, concepte que ja hem mencionat anteriorment en aquest treball. Bàsicament, podríem entendre aquest software com un sistema operatiu que permet córrer altres sistemes operatius dins d’ell.

L’ESXi no emula el *hardware*, sinó que s’encarrega de “repartir” els recursos d’aquest entre les diferents màquines virtuals, de manera que cada màquina tindrà assignada una certa quantitat de memòria, CPU, etc. Una manera de veure el procés gràficament seria la següent imatge:



Figura 7. VMware ESXi, VMware

Per altra banda, tenim VMware vCenter Server, que és l'eina que permet administrar tots els nostres ESXi i màquines virtuals de la millor manera. Aquest vCenter està pensat com una mena de sala de control on poder controlar el nostre entorn virtual. Tot i això, no és necessari per a controlar les màquines virtuals, i per tant no tindrà sentit si només tenim un ESXi.

Està pensat per facilitar el treball del personal que s'encarrega d'aquest entorn, i és per això que un cop configurat, ens podem connectar a aquest centre ja sigui mitjançant una aplicació d'escriptori anomenada vClient o en la versió més moderna via web, i tindrem tot aquest entorn presentat d'una manera senzilla i intuïtiva.

Per acabar de lligar aquesta idea, podem visualitzar la següent imatge que reflexa molt bé com funciona aquest centre:

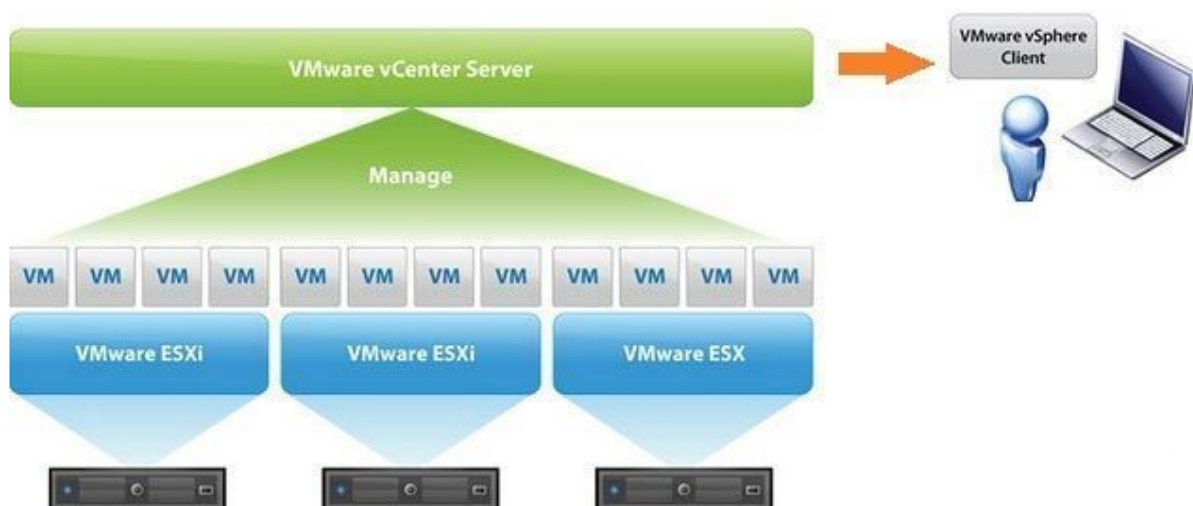


Figura 8. VMware vCenter, VMware

En definitiva, vSphere s'ha convertit en un dels *softwares* més utilitzats avui dia per les empreses per construir i mantenir un entorn virtual, estalviant una quantitat d'energia, recursos i per tant de diners molt gran, ja que on abans teníem un servidor ara en tenim 15 Windows Servers.

Per fer una mica de números, aquest mateix any 2021 en l'informe IDC que informa sobre els beneficis del consumidor, es van obtenir uns números especialment bons en relació al servei proporcionat per

AWS i VMware, que hem estant comentant anteriorment. Alguns dels números obtinguts són un 43% de reducció del cost total de les operacions, un 479% de ROI als 5 anys o uns guanys anuals de 9,6 milions de dòlars.

Tot i això, aquest no és l'únic servei que ens pot interessar. Per exemple, també podem contractar una SAN virtual de diverses maneres. Una d'elles pot ser VMware vSAN, que és un *software* de virtualització d'emmagatzematge, el qual permet reduir molt la complexitat i els costos associats a l'emmagatzematge tradicional així com també agilitar les operacions relacionades amb aquest emmagatzematge.

A més, vSAN es pot combinar amb vSphere de manera que podem gestionar tots els recursos informàtics, ja siguin màquines virtuals, servidors... i l'emmagatzematge amb una sola plataforma. Aquest fet és un gran avantatge que permet estalviar diners, temps i un guany de salut per part dels treballadors que podran integrar molt fàcilment aquest *software* amb el *hypervisor* o el que necessitin.

Per acabar, posarem un últim exemple relacionat amb la recuperació de dades, i aquest és *Site Recovery Manager*. Es tracta d'un *software* de recuperació davant possibles problemes o desastres que permet preparar l'entorn de treball per a la prevenció de desastres i per a les possibles auditories. També afegeix proves que no provoquen cap interrupció, simplifiquen les auditories i garanteixen que es compleixin els objectius de RTO.

També cal mencionar que aquest *software* és compatible amb vSphere i per tant ens trobem en la mateixa situació que abans, on tot és compatible i per tant podríem tenir prevenció d'errors amb *Site Recovery Manager*, amb un entorn controlat per vSphere.

4.2.- Microsoft Azure

Si estem parlant sobre les millors opcions de virtualització, no podem obviar l'opció que ens presenta Microsoft, Azure. Aquest servei té unes opcions reduïdes si el comparem amb altres com VMware, i està pensat principalment per ser utilitzat per empreses que vulguin tenir un conjunt de màquines virtuals amb les seves dades emmagatzemades en el núvol.

Això sí, Azure té un avantatge respecte a altres com VMware, i és que ofereix diversos productes imprescindibles avui dia com *kubernetes* de manera gratuïta. Podem trobar altres com un servei especial per crear aplicacions, tant per mòbil com per web així com també APIS en nodeJS. També tenim un entorn virtual per a poder provar i fer *testing* de les nostres aplicacions o una *database* NoSQL, això sí, amb un número limitat de peticions.

Si en parlem sobre els serveis no gratuïts, podem trobar alguns dels més importants com la creació de màquines virtuals, tant en Linux com en Windows en el núvol. Cal dir que aquestes VMs no han de ser d'un tipus determinat que marqui Azure, sinó que podem fer deploy de les nostres pròpies màquines virtuals. Cal mencionar que aquestes màquines tindran connexió a la xarxa amb balanceig de càrrega incorporat, de forma que la velocitat d'aquesta davant peticions d'un número de màquines molt gran hauria de ser constant per a totes.

A més, per acompanyar aquestes VMs podem tenir un sistema d'emmagatzematge format per discos SSD d'alt rendiment, de forma que les dades que tenim en les màquines virtuals es mouran molt ràpidament. Aquests SSD també tindran un sistema de redundància en les dades que podrà ser de dos tipus: LDR o ZRS.

Però abans d'explicar què és cada una i quines diferències tenen, cal explicar com defineix Azure la seva arquitectura en aplicacions de n nivells. Per simplificar, trobarem bàsicament dues zones principals: primària i secundària. La regió primària serà la zona usada per a treballar i operar, usant la secundària com a zona redundant i de commutació per error, de manera que si la primària cau sempre podem seguir treballant en la secundària sense cap tall de cara a l'usuari. Aquesta idea es veu reflexada de manera molt clara en la següent imatge:

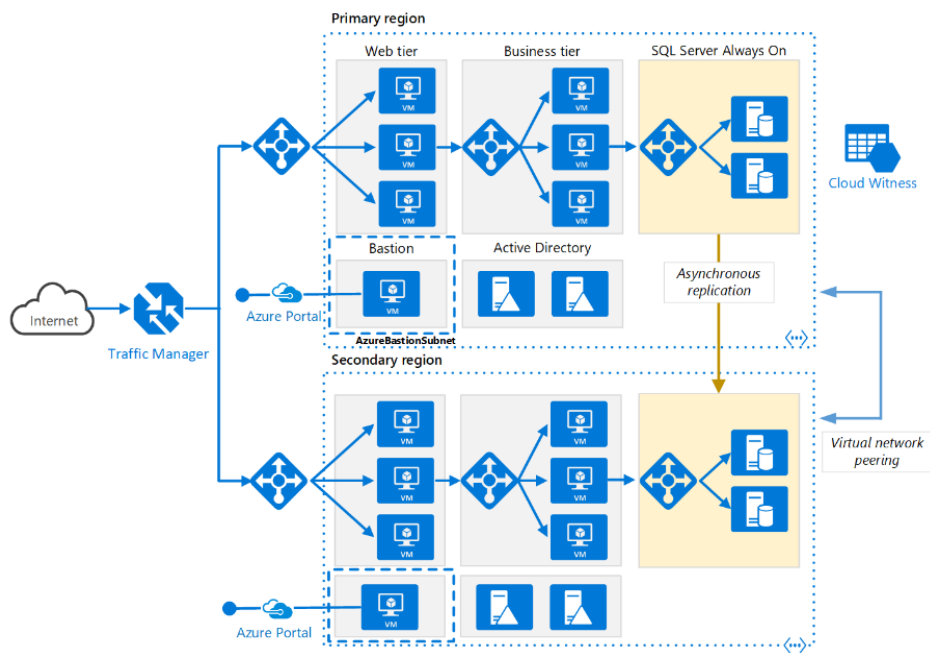


Figura 9. Esquema simplificat de l'arquitectura de Microsoft Azure

El LDR és un emmagatzematge amb redundància local que copia les dades de forma síncrona tres cops dins d'una única ubicació física en la regió primària. Aquesta opció és menys costosa ja que només estem utilitzant una zona de la regió primària, però no es recomana utilitzar-la per a aplicacions que necessitin alta disponibilitat.

En canvi, el ZRS és un emmagatzematge amb redundància de zona que copia les dades de forma síncrona en tres zones de disponibilitat d'Azure en la regió primària. A diferència del LDR, aquí copiem les dades en tres zones diferents, i per tant és la opció ideal per a les aplicacions d'alta disponibilitat. Azure recomana utilitzar ZRS en la regió primària, i a més replicació en la zona secundària.

Continuant amb altres productes que ofereix Azure, trobem alguns com diverses opcions de bases de dades, ja siguin SQL o NoSQL. Aquestes bases de dades tindran ja incorporat un sistema intel·ligent que ens ajudarà a l'hora d'emmagatzemar les nostres dades. També cal mencionar que aquestes bases de dades tindran disponibles un conjunt d'APIs ja creades que es poden utilitzar casi directament amb uns petits canvis, això com també la opció d'utilitzar les nostres pròpies APIs.

En conclusió, Microsoft Azure és una bona opció si el que busquem és tenir una xarxa de màquines virtuals amb diversos serveis. Tot i això, si volem una xarxa més avançada amb més possibilitats, hi han altres millors alternatives, com pot ser VMware que té tot el que ofereix Azure i més.

4.3.- Xen

No es pot parlar de virtualització sense esmentar un projecte originalment desenvolupat per la



Universitat de Cambridge l'any 2003 i que actualment està a càrrec de la Fundació Linux amb el suport d'Intel: Xen Project, una solució amb nombrosos contribuïdors de renom com Alibaba, AWS, AMD, Arm, Bitdefender, Cavium, Citrix, Huawei, Intel, Oracle i Qualcomm.

Figura 10. Logotip de Xen, Xen Project

Com hem mencionat a l'apartat 3, Xen és un *hypervisor* de tipus 1 i que, per tant, és una capa de *software* que s'instal·la directament al servidor físic i al seu *hardware*. Aquests, com que no s'executen dins d'un sistema operatiu, ofereixen un rendiment i estabilitat excel·lents. Cal mencionar que Xen és l'únic *hypervisor open source* (de codi obert) d'aquest tipus. Suporta dos tipus de virtualització, paravirtualització i HVM (*hardware virtualized machine*) també coneguda com a "virtualització completa". Tot i que estan explicades a l'apartat 3, en farem un breu recordatori. Bàsicament, la primera fa servir sistemes operatius de *guest* modificats coneguts com a *enlightened*. Aquests sistemes operatius són conscients que s'estan virtualitzant i, per tant, no requereixen dispositius virtuals. En el seu lloc, fan *syscalls* especials al *hypervisor* que els permetrà accedir als recursos com CPU, emmagatzematge i xarxa.

En canvi, els *guest* que utilitzen HVM no han de ser modificats, ja que el *hypervisor* crearà un conjunt de dispositius completament virtuals per a que els interpretin com si fos un ordinador físic. Aquesta emulació requereix més despeses que la paravirtualització, però permet que els sistemes operatius *guest* no modificats com Windows s'executin a la part superior del *hypervisor*. Les HVM funcionen mitjançant extensions de virtualització a la CPU. S'han introduït diverses iteracions d'aquestes durant l'última dècada aproximadament, conegudes col·lectivament com Intel VT i AMD-V, i el

desenvolupament continua. Aquesta és la tecnologia predominant; tots els servidors recents, molts ordinadors d'escriptori i alguns sistemes mòbils estan equipats amb almenys algunes extensions.

Un cop vist què és, passem a la seva arquitectura. Com podem comprovar a la figura 10, el *hypervisor* és de tipus 1, ja que situa directament sobre els components físics. Per sobre d'aquesta capa trobem les màquines virtuals que executarem i *dom0*, el controlador de domini, una VM que tindrà dues funcionalitats afegides: serà l'encarregat d'administrar les màquines virtuals, donant ordres al *hypervisor*; i haurà de comptar amb els drivers dels dispositius físics per a que es garanteixi la compatibilitat de les màquines anomenades VM_0, VM_1, \dots, VM_a a la figura amb el *hardware* del *host*.

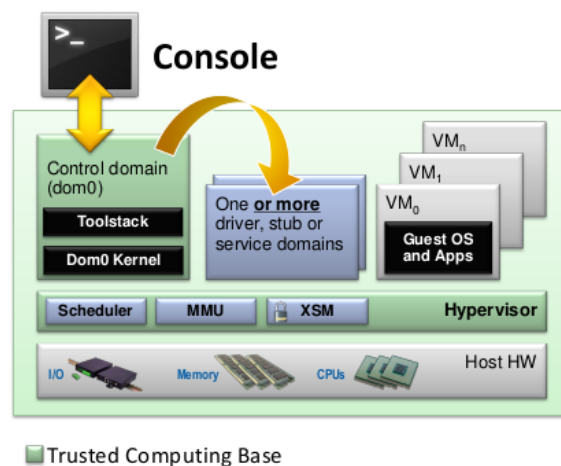


Figura 11. Esquema simplificat de l'arquitectura de funcionament de Xen Project

Com hem esmentat anteriorment, Xen Project implementa dos tipus de virtualització. En el cas de la paravirtualització, tenim dos camins de comunicació per on circulen les dades, la xarxa i l'emmagatzematge, anomenats *net-back/netfront* i *blk-back/blk-front*, respectivament. Cadascuna consta de dues parts: un *backend* a *dom0* que proporciona el dispositiu virtual, i un *frontend* al *guest*, que el permetrà accedir a aquest. A més, aquesta capa compta amb una capa d'abstracció de la paravirtualització per a interrupcions, comptadors i taules de pàgines, entre molts d'altres.

En el cas d'HVM, *dom0* utilitza les extensions de virtualització de maquinari proporcionades pel fabricant de la CPU (Intel VT i AMD-V). En un principi només es tenia suport per a la pròpia CPU, però amb el temps d'han afegit funcionalitats com virtualització per als dispositius d'entrada/sortida i gestió de la taula de pàgines.

Per a finalitzar, farem esment de les virtuts de les que presumeix la pàgina oficial de Xen Project, ja que, tot i que en alguns casos siguin paraules buides, ens sembla interessant veure com es descriu una companyia. Són les següents: codi obert, fiabilitat, flexibilitat, suport, seguretat i modularitat.

4.4.- KVM

Per a concloure aquest apartat d'exemples, pensem que és una bona idea explicar un cas especial i interessant de tecnologia de virtualització: *Kernel-based Virtual Machine*, més conegut com KVM. Qumranet va començar el seu desenvolupament de 2006 i va ser publicat per primer cop amb la versió 2.6.20 del *kernel* de Linux, el 5 de febrer de 2007. Al 2008, la companyia va ser adquirida per Red Hat Inc, qui encara manté el codi.

Tot i funcionar com a un *hypervisor* corrent, la particularitat de KVM ve donada pel seu origen, ja que és un mòdul del *kernel* de Linux que permet a aquest *kernel* funcionar com a *hypervisor*, el qual ens permetrà crear màquines virtuals i administrar-les, entre d'altres funcionalitats habituals. Com que és un recurs inherent a Linux, hereda els seus avantatges, com el fet de ser *open source* o tenir un molt bon manteniment i suport ràpid i eficient. En altres paraules, tot el que té un ho té l'altre, i les millores en funcionalitats i estabilitat que s'apliquen a Linux beneficien directa i immediatament a KVM.

Un altre aspecte que hereda de Linux i que és essencial per a un *hypervisor* són alguns components com els gestors de memòria, entrada/sortida, xarxa i processos o controladors de dispositius. El funcionament del sistema consisteix en que les màquines virtuals són un processos de Linux, gestionats pel planificador de processos com normalment, i heredant els dispositius que interpreta i utilitza el *hypervisor*, és a dir, KVM.

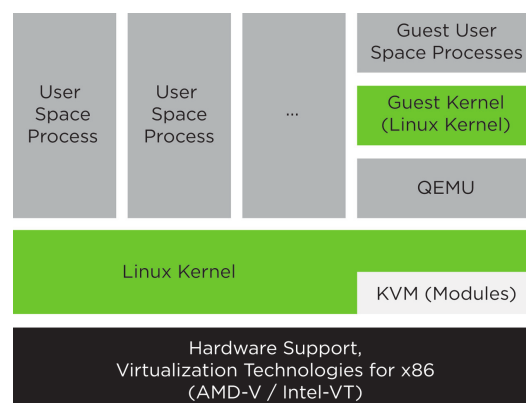


Figura 12. Arquitectura de KVM, SUSE

Com en la secció de Xen Project, farem un repàs dels aspectes positius que s'esmenten a la web oficial de Red Hat per a després fer una breu comparativa entre els dos, entre un *hypervisor* més convencional i l'integrat del *kernel* de Linux. Trobem els següents: seguretat, emmagatzematge, suport *hardware*, gestió de memòria, migració en temps real, rendiment, escalabilitat, planificació, control de recursos i baixa latència.

Com es pot apreciar, molts dels punts forts són compartits, fet que pot portar a la conclusió de que triant indistintament no ens equivocaríem. Fent una mica més de recerca, però, trobem que comparteixen beneficis però no sempre en un grau similar. Per exemple, tot i que ambdós ofereixen bons rendiment, estabilitat i flexibilitat, la majoria d'articles consideren superior a Xen en aquests aspectes. Una altra característica a tenir en compte és el suport per a la paravirtualització, no disponible per a KVM, que utilitza HVM.

Tot i aquestes diferències, l'elecció d'un *hypervisor* sempre s'acaba reduint a cada cas particular, ja que en depèn molt de la infraestructura del nostre sistema. Empreses com Oracle o Citrix impulsen Xen, mentre que Red Hat i últimament Amazon estan optant per KVM, i si el nostre escenari té a veure amb algun dels serveis que aquestes companyies ofereixen, probablement ens serà més fàcil i còmode escollir el *software* que proposen ja que tindrà una millor integració.

5.- Conclusions

Un cop hem vist què és la virtualització, com funciona i alguns exemples de com la podem utilitzar, podem afirmar que és una eina molt potent i que encara té potencial per esprémer. Només cal estudiar els números aconseguits amb l'ajut de la virtualització en grans companyies, on abans teníem un servidor i ara en tenim vuit, estalviant electricitat, recursos i per tant diners, podent invertir en altres aspectes com una bona xarxa que complementi a aquestes màquines virtuals.

A més, la virtualització no només serveix per a produir màquines virtuals, sinó que també pot ajudar en altres camps, ja sigui virtualitzant emmagatzematge, xarxes, sistemes operatius o aplicacions. I és quan combinem totes aquestes opcions quan una empresa realment pot treure tot el potencial que actualment aquesta tecnologia ens pot proporcionar, sobretot si el *software* és compatible entre ell.

Com hem vist al llarg del treball, podríem tenir un cas com el de VMware, on a partir d'un centre de control podem administrar tots els recursos dels que disposem, ja siguin dades, màquines, servidors, VLANs o el que necessitem. És aquí on creiem que la virtualització pot ser de més utilitat en un CPD, oferint tot l'estalvi de recursos, així com també l'eficiència i l'organització que també ens proporciona, aconseguint tenir un millor centre en tots els sentits.

En definitiva, la virtualització ha sigut una revolució a la informàtica i sobretot en les empreses que la utilitzen, ja siguin un CPD, una multinacional o una petita empresa que ven alumini. El fet de poder estalviar recursos i per tant diners aconseguint el mateix objectiu en diferents camps és simplement l'objectiu principal d'una companyia, i és per això que la virtualització està tan present en el món empresarial avui dia.

Per últim però no menys important, tampoc oblidar que el fet de tenir menys recursos físics actius i tenir-ne més virtuals està ajudant a reduir les emissions produïdes en el món per culpa de la informàtica, fet molt important si volem seguir tenint un planeta on construir CPDs.

6.- Bibliografía

Apartat 2

Introduction to virtualization, Oracle, 2012

https://docs.oracle.com/cd/E26996_01/E18549/html/BHCCJFC.html

A brief history of virtualization, or why do we divide something at all, Victoria Fedoseenko, 2019

<https://www.ispsystem.com/news/brief-history-of-virtualization>

The History Of Virtualization Information Technology Essay, UKEssays, 2018

<https://www.ukessays.com/essays/information-technology/the-history-of-virtualization-information-technology-essay.php>

With long history of virtualization behind it, IBM looks to the future, Jon Brodtkin, 2009

<https://www.networkworld.com/article/2254433/with-long-history-of-virtualization-behind-it-ibm-looks-to-the-future.html>

VMware History and Interactive Timeline, VMware, 2021

<https://www.vmware.com/timeline.html>

Apartat 3

Virtualización: definición y enfoques, IONOS, 2019

<https://www.ionos.es/digitalguide/servidores/configuracion/virtualizacion/>

¿Qué es la virtualización?, Red Hat, 2018

<https://www.redhat.com/es/topics/virtualization/what-is-virtualization>

Virtualization Essentials, VMware, 2014

<https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/ebook/gated-vmw-ebook-virtualization-essentials.pdf>

What is Virtualization? - Virtualization Definition, Citrix

<https://www.citrix.com/es-es/glossary/what-is-virtualization.html>

Qué es virtualización - Definición, Azure

<https://azure.microsoft.com/es-es/overview/what-is-virtualization/>

Virtualización, Wikipedia

<https://es.wikipedia.org/wiki/Virtualizaci%C3%B3n>

What is Server Consolidation and Why Should You Do It?, John Hurley, Smartfile, 2020

<https://www.smartfile.com/blog/what-is-server-consolidation-and-why-should-you-do-it/>

Virtualización de aplicaciones, inLab FIB, 2010

<https://inlab.fib.upc.edu/es/virtualizacion-de-aplicaciones>

Apartat 4

VMware history, VMware, 2018

<https://www.vmware.com/latam/timeline.html>

VMware products, VMware 2021

<https://www.vmware.com/es/products.html>

VMware Cloud on AWS, Amazon, 2020

<https://aws.amazon.com/es/vmware/>

¿Qué es vSphere 7? | Software de virtualización de servidores, VMware, 2021

<https://www.vmware.com/es/products/vsphere.html>

¿Qué es Vmware vSphere? ¿Qué es virtualizar?, Manuel Serrano, 2017

<https://virtualizadesdezero.com/que-es-vmware-vsphere/>

¿Qué es VMware vSAN? | Virtual SAN, VMware, 2021

<https://www.vmware.com/es/products/vsan.html>

VMware Site Recovery, VMware, 2021

<https://cloud.vmware.com/es/vmware-site-recovery>

Qué es VMware Site Recovery, VMware, 2021

<https://www.vmware.com/es/products/site-recovery-manager.html>

Explore Microsoft Azure, Microsoft Azure, 2021

<https://bit.ly/3rMRGVr>

Redundancia de Azure Storage, Microsoft Azure, 2021

<https://docs.microsoft.com/es-es/azure/storage/common/storage-redundancy>

Azure Architecture, Microsoft Azure, 2021

<https://docs.microsoft.com/es-es/azure/architecture/reference-architectures/n-tier/multi-region-sql-server>

Xen Project: Home, 2021

<https://xenproject.org/>

Dive into the Xen *hypervisor* – IBM Developer, Bhanu P Tholeti, 2011

<https://developer.ibm.com/depmodels/cloud/articles/cl-hypervisorcompare-xen/>

What is KVM?, RedHat, 2021

<https://www.redhat.com/en/topics/virtualization/what-is-KVM>

Introduction to KVM Virtualization | Virtualization Guide, SUSE, 2020

<https://documentation.suse.com/es-es/sles/15-SP2/html/SLES-all/cha-kvm-intro.html>