

Virtualization

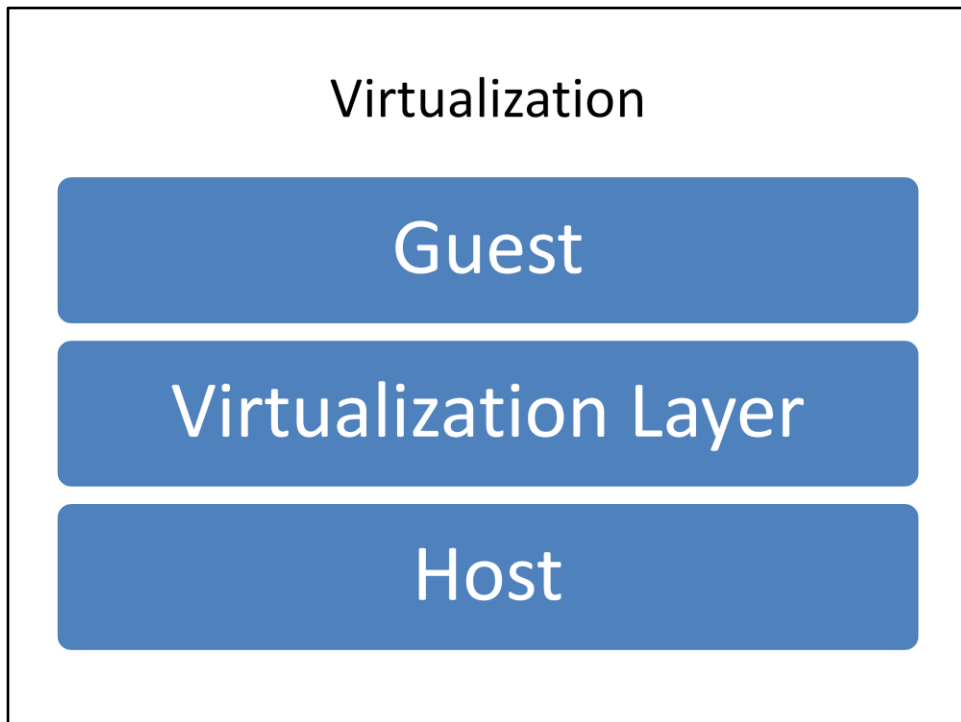
TIE-23600 Palvelupohjaiset
järjestelmät

Virtualization

- Virtualization abstracts some elements of computing
 - Hardware, storage, networking, runtime environment, ...
- Allows creating *virtual* computing environments
 - Not directly tied to the underlying hardware
- Virtualization is essential in the implementation of IaaS and PaaS providers

Virtualisoinnin avulla voidaan purkaa suora linkki suoritettavan sovelluksen (tai käyttöjärjestelmän tms.) ja sitä suorittavan laitteiston välillä. Näin saavutetaan joustavuutta laitteiston käytössä. Voidaan esimerkiksi suorittaa useampaa käyttöjärjestelmää samaan aikaan yhdessä tietokoneessa.

Virtuaalisointitekniikat ovat erittäin tärkeitä pilvipalveluiden (erityisesti IaaS) toteutuksessa. IaaS-tarjoajat eivät anna käyttäjilleen pääsyä suoraan laitteistoonsa, vaan jonkinlaiseen virtuaaliympäristöön. Tästä on useita hyötyjä, joihin tutustutaan tarkemmin myöhemmillä kalvoilla.



Virtualisointiympäristössä on kolme pääosaa: isäntä (host), virtualisointikerros ja vieras (guest).

Isäntä pitää sisällään fyysisen laitteiston sekä yleensä myös sen päällä ajettavan käyttöjärjestelmän.

Virtualisointikerros tarjoaa virtuaaliympäristön vieraan (guest) käyttöön. Se tarjoaa rajapinnan, johon tulevat kutsut se välittää isännälle muunnettuaan ne isännän ymmärtämään muotoon. Virtualisointikerros voi tarjota myös muita ominaisuuksia liittyen mm. turvallisuuteen.

Vieras (guest) on sovellus, käyttöjärjestelmä tms. jota ajetaan virtuaalikerroksen päällä. Sen sijaan, että se kommunikoi suoraan isännän kanssa (kuten normaalitilanteessa ilman virtualisointia), kommunikoi se virtualisointikerroksen kanssa. Vieraan ei tosin tarvitse tietää olevansa ajossa virtuaaliympäristössä.

Virtualization

Guest

- Application, OS, ...
- Running on top of virtualization layer

Virtualization Layer

- *Virtual* hardware, storage, networking, runtime env., ...

Host

- *Physical* hardware, storage, networking

Benefits of virtualization

- Increased security
 - Host can control and filter actions of guest
- Increased flexibility
 - Sharing
 - Aggregation
- Portability
 - The guest can be moved, duplicated, suspended, etc.
- Allows efficient utilization of resources

Virtualisoinnilla on monia hyviä puolia.

Ensinnäkin sillä saadaan aikaan turvallinen ajoympäristö, jossa virtualisointikerros voi hallita sitä, mitä vieras saa tehdä. Koska kaikki käskyt kulkevat virtualisointikerroksen läpi, voi se esim. rajoittaa tiedostojärjestelmän tai verkon käyttöä.

Toisekseen virtualisoinnilla saadaan paremmin hyödynnettyä resursseja. Yhdessä fyysisessä tietokoneessa voi olla ajossa useampia käyttöjärjestelmiä samaan aikaan. Tai päinvastoin: useampi fyysinen kone voi toteuttaa virtuaaliympäristön, joka näkyy yhtenä suurena tietokoneena.

Näin, jos laaS-tarjoaja haluaa tarjota ympäristöä, jossa on 2 suoritinydintä, 4 gigatavua muistia, 1 Mbit/s verkkoyhteys ja 100 GB levytilaa, ei tarjoajan tarvitse hankkia tietokonetta täsmälleen näillä spekseillä vaan voi luoda tätä vastaavan virtuaaliympäristön.

Virtualisoinnissa on myös se hyvä puoli, että vierassovelluksia voidaan siirrellä fyysiseltä laitteelta toiselle ilman että vieras välttämättä huomaa mitään.

Jos vieras on vaikkapa web-palvelin, jonne ei ole tullut yhtään pyyntöä pitkään aikaan, voidaan se laittaa "nukkumaan" jolloin resursseja voidaan käyttää muuhun tarkoitukseen. Tarvittaessa vieras voidaan käynnistää myöhemmin vaikkapa eri

isäntäkoneella täsmälleen samasta tilasta kuin missä se oli aiemmin.

Sharing

Guest

Guest

Guest

Virtualization Layer

Host

Aggregation

Guest

Virtualization Layer

Host

Host

Host

Disadvantages of virtualization

- Performance degradation
 - Virtualization layer brings some performance penalty
 - However, with current technology, the penalty is quite small
- Lack of features
 - Virtualization environment may not provide all the features available in the host

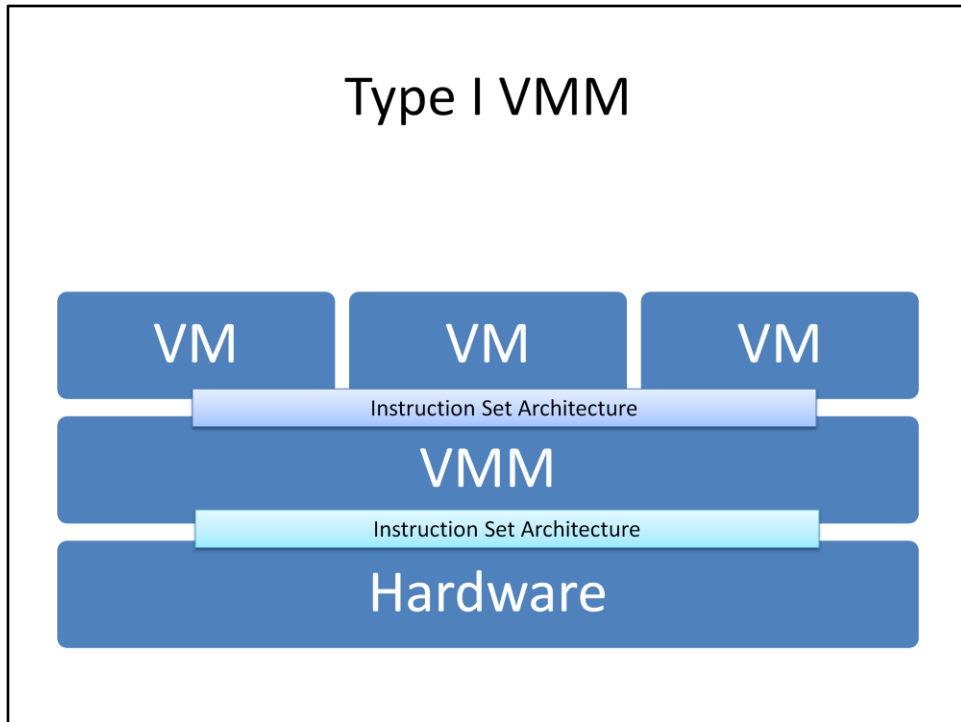
Virtualization techniques

- Execution virtualization
 - Hardware-level virtualization
 - Programming language-level virtualization
 - Application-level virtualization
- Storage virtualization
- Network virtualization

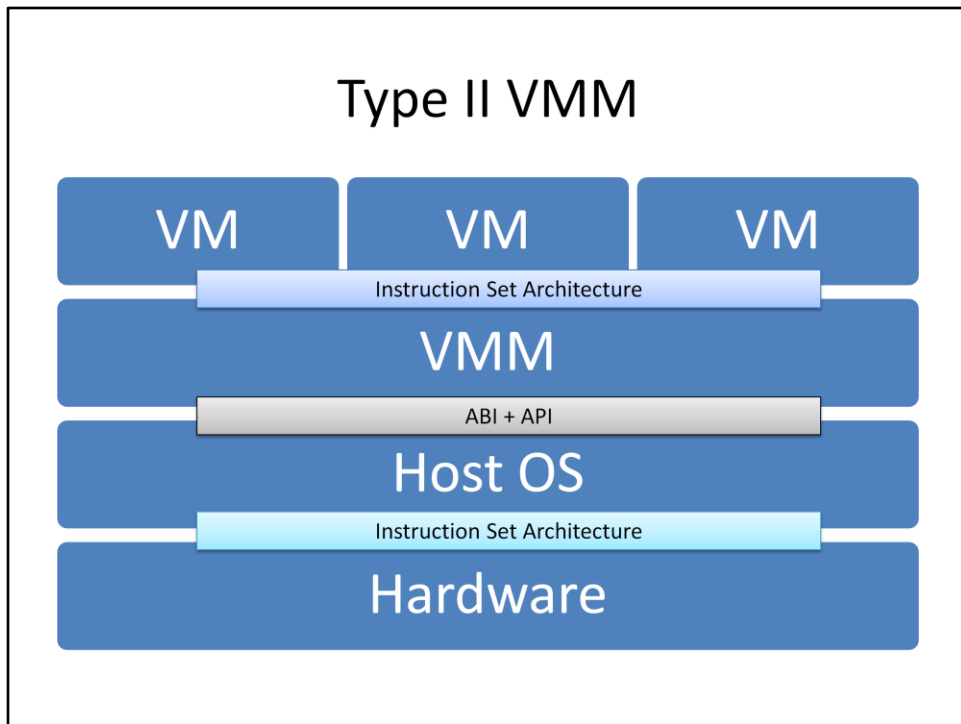
Hardware-level virtualization

- Virtual Machine Monitor (VMM) creates a virtual hardware environment for guest operating system
 - Type I and Type II VMMs
- The guest OS interacts via an Instruction Set Architecture (ISA) provided by the virtualization layer

Type I VMM



Instruction Set Architecture (ISA) on laitteiston ja ohjelmiston välinen rajapinta.



Instruction Set Architecture (ISA) on laitteiston ja ohjelmiston välinen rajapinta.

ABI (Application Binary Interface) ja API (Application Programming Interface) ovat käyttöjärjestelmän sovelluksille tarjoamia rajapintoja.

Hardware virtualization techniques

- Hardware-assisted virtualization
 - Hardware provides support for virtualization
 - Reduces performance penalty
- Full virtualization
 - Provides complete emulation of hardware
 - As opposed to partial virtualization
 - Advantage: full isolation
 - Usually achieved by a combination of software and hardware-assisted virtualization
- Paravirtualization
 - Non-transparent: requires modification to the guest operating system

Programming language-level virtualization

- Some programming languages run on top of a *high-level virtual machine* (HLVM)
 - Eg. Java, Python
- Source code is compiled into intermediate presentation called *bytecode*
- Bytecode is executed on a HLVM
 - The HLVM converts bytecode instructions into native instructions
- Same bytecode can be run in multiple environments
 - Only the HLVM implementation is environment-specific
- Useful in PaaS implementations

Virtualisoinnissa ei tarvitse välttämättä emuloida laitteistoa, myös korkeamman tason virtualisointitekniikoita on olemassa. Yksi tällainen on ohjelmointikielitason virtualisointi, jota jotkut ohjelmointikieliset (mm. Java ja Python) hyödyntävät. Sen sijaan, että lähdekoodi käännettäisiin suoraan natiiveiksi konekäskyiksi, käännetäänkin se ”tavukoodiksi” (bytecode), jonka suorittamiseen tarvitaan korkean tason virtuaalikone (HLVM). Tavukoodi sisältää käskyjä, jotka HLVM:n on kohtuullisen suoraviivaista muuntaa ajonaikaisesti (tai ns. JIT-käännöksellä) natiiveiksi konekäskyiksi.

PaaS-tarjoajat käyttävät usein tämän tason virtualisointia. PaaS:n käyttäjä voi kirjoittaa ohjelman esimerkiksi Pythonilla ja lähettää tavukoodin PaaS-tarjoajan ajettavaksi. PaaS-tarjoajalla on oma virtuaalikoneensa tavukoodin ajamiseen.

Application-level virtualization

- Allows applications to be run in environments that don't directly support all the features of required by the application
- Eg. Wine – an emulator for running Windows applications in Linux
- Not that relevant in cloud computing

Virtualization and cloud computing

- Hardware virtualization is important in implementing IaaS providers
 - Efficient use of resources
 - More flexibility
- Programming language virtualization is important in implementing PaaS providers
- Virtualization allows user applications to be *sandboxed*
 - Run multiple applications in complete isolation even on the same physical machine