Contents

Articles

References

Article Licenses

Comparison of application virtual machines

This article lists some software virtual machines that are typically used for allowing application bytecode to be portably run on many different computer architectures and operating systems. The application is usually run on the computer using an interpreter or just-in-time compilation. There are often many implementations of a given virtual machine, each covering a different functionality footprint.

Comparison of virtual machines

The table here summarizes elements for which the virtual machine designs intended to be efficient, not the list of capabilities present in any implementation.

Virtual machine	Machine model	Memory management	Code security	Interpreter	JIT	Precompilation	Shared libraries	Common Language Object Model	Dynamic typing
CLR	stack	automatic or manual	Yes	No	Yes	Yes	Yes	Yes	Yes
Dis (Inferno)	register	automatic	Yes	Yes	Yes	Yes	Yes	No	No
DotGNU Portable.NET	stack	automatic or manual	No	No	Yes	Yes	Yes	Yes	No
JVM	stack	automatic	Yes	Yes	Yes	Yes	Yes	Yes	No
JikesRVM	stack	automatic	No	No	Yes	No	?	No	No
LLVM	register	manual	No	Yes	Yes	Yes	Yes	Yes	No
Mono	stack	automatic or manual	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parrot	register	automatic	No	Yes	Yes	No	Yes	Yes	Yes
Dalvik	register	automatic	Yes	Yes	Yes	?	?	No	No
libJIT	stack	manual	No	No	Yes	No	No	?	No
Squeak	stack	automatic	No	Yes	Yes	source to bytecode	Yes	No	Yes

Virtual machine instructions process data in local variables using a main **model of computation**, typically that of a stack machine, register machine, or random access machine often called the memory machine. Use of these three techniques is motivated by different tradeoffs in virtual machines vs physical machines, such as ease of interpretation, compilation, and verifiability for security.

Memory management in these portable virtual machines is addressed at a higher level of abstraction than in physical machines. Some virtual machines, such as the popular JVM, are involved with addresses in such a way as to require safe automatic memory management by allowing the virtual machine to trace pointer references, and disallow machine instructions from manually constructing pointers to memory. Other virtual machines, such as LLVM, are more like traditional physical machines, allowing direct use and manipulation of pointers. CIL offers a hybrid in between, offering both controlled use of memory (like the JVM, which allows safe automatic memory management), while also offering an 'unsafe' mode that allows direct manipulation of pointers in ways that can violate type boundaries and permission.

Code security generally refers to the ability of the portable virtual machine to run code while only offering it a prescribed set of capabilities. For example, the virtual machine might only allow the code access to a certain set of functions or data. The same controls over pointers which make automatic memory management possible and allow the virtual machine to ensure typesafe data access are used to assure that a code fragment is only allowed to certain elements of memory and cannot sidestep the virtual machine itself. Other security mechanisms are then layered on top as code verifiers, stack verifiers, and other techniques.

An **interpreter** allows programs made of virtual instructions to be loaded and immediately run without a potentially costly compilation into native machine instructions. Any virtual machine which can be run can be interpreted, so the column designation here refers to whether the design includes provisions for efficient interpretation (for common usage).

Just-in-time compilation or **JIT**, refers to a method of compiling to native instructions at the latest possible time, usually immediately before or during the running of the program. The challenge of JIT is more one of implementation than of virtual machine design, however, modern designs have begun to make considerations to help efficiency. The simplest JIT techniques simply perform compilation to a code-fragment similar to an offline compiler. However, more complicated techniques are often employed, which specialize compiled code-fragments to parameters that are known only at runtime (see Adaptive optimization).

Precompiling refers to the more classical technique of using an offline compiler to generate a set of native instructions which do not change during the runtime of the program. Because aggressive compilation and optimization can take time, a precompiled program may launch faster than one which relies on JIT alone for execution. JVM implementations have mitigated this startup cost by using interpretation initially to speed launch times, until native code-fragments can be generated through JIT.

Shared libraries are a facility to reuse segments of native code across multiple running programs. In modern operating systems, this generally means using virtual memory to share the memory pages containing a shared library across different processes which are protected from each other via memory protection. It is interesting that aggressive JIT techniques such as adaptive optimization often produce code-fragments unsuitable for sharing across processes or successive runs of the program, requiring a tradeoff be made between the efficiencies of precompiled and shared code and the advantages of adaptively specialized code. For example, several design provisions of CIL are present to allow for efficient shared libraries, possibly at the cost of more specialized JIT code. The JVM implementation on Mac OS X uses a Java Shared Archive (apple docs [1]) to provide some of the benefits of shared libraries.

List of application virtual machine implementations

In addition to the portable virtual machines described above, virtual machines are often used as an execution model for individual scripting languages, usually by an interpreter. This table lists specific virtual machine implementations, both of the above portable virtual machines, and of scripting language virtual machines.

Virtual machine	Languages	Comments	Interpreter	JIT	Implementation Language	SLoC
Adobe Flash Player (aka Tamarin)	ActionScript, SWF (file format)	interactive web authoring tool. bytecode is named "ActionScript Byte Code (.abc)"	Yes	Yes	C++	135k (initially released)
BEAM	Erlang, Reia, Lisp Flavoured Erlang	There exists a native-code compiler, HiPE.	Yes	No	C	247k
Clipper p-code	Clipper, Harbour	plankton, HVM	Yes	No	C	
Dis (Inferno)	Limbo	Dis Virtual Machine Specification [2]	Yes	Yes	C	15k + 2850 per JIT arch + 500 per host OS
DotGNU/Portable.NET	CLI languages including: C#	Clone of Common Language Runtime	No	Yes	C, C#	
Forth	Forth	Features are simplified, usually include assembler, compiler, text-level and binary-level interpreters, sometimes editor, debugger and OS. Compilation speeds are >20 SKLOC/S and behave much like JIT.	Yes	No	Forth, Forth Assembler	2.8K to 5.6K; advanced, professional implementations are smaller.
Glulx	Glulx, Z-code					
Icon	Icon					
JVM	Java, Groovy, JRuby, C, C++, Clojure, Scala and several others [3]	Reference implementation [4] by Sun ; OpenJDK: code under GPL ; IcedTea: code and tools under GPL	Yes	Yes	JDK, OpenJDK & IcedTea with regular JIT : Java, C, ASM ; IcedTea with the "Zero" JIT : Java, C	JVM is around 6500k lines; TCK is 80k tests and around 1000k lines
LLVM	C, C++, Objective-C, Ada, and Fortran	MSIL, C and C++ output are supported. ActionScript Byte Code output is supported by Adobe Alchemy. bytecode is named "LLVM Bytecode (.bc)". assembly is named "LLVM Assembly Language (*.ll)".	Yes	Yes	C++	
Lua	Lua		Yes	LuaJIT [5]	C	13k + 7k LuaJIT
MMIX	MMIXAL					
Mono	CLI languages including: C#, VB.NET, IronPython, IronRuby, and others	clone of Common Language Runtime.	Yes	Yes	C#, C	2332k

Oz	Oz, Alice					
NekoVM	currently Neko and haXe		Yes	x86 only	C	46k
O-code machine	BCPL					
p-code machine	Pascal	UCSD Pascal, widespread in late 70s including Apple II				
Parrot	Perl (6 & 5), NQP-rx, PIR, PASM, PBC, BASIC, bc, C, ECMAScript, Lisp, Lua, m4, Tcl, WMLScript, XML, and others		Yes	Yes	C, Perl	111k C, 240k Perl
Perl virtual machine	Perl	op-code tree walker	Yes	No	C, Perl	175k C, 9k Perl
CPython	Python		Yes	Psyco	C	387k C, 368k Python, 10k ASM, 31k Psyco
PyPy	Python	Self-hosting implementation of Python, next generation of Psyco	Yes	Yes	Python	
Rubinius	Ruby	Virtual machine for another Ruby implementation	Yes	Yes	C++, Ruby	
SEAM	Alice					
ScummVM	Scumm	Computer game engine				
SECD	ISWIM, Lispkit Lisp					
Squirrel	Squirrel		Yes	Squirrel_JIT	C++	12k
Smalltalk	Smalltalk					
SQLite	SQLite opcodes	Virtual database engine				
Squeak	Squeak Smalltalk	Self hosting implementation of Squeak virtual machine. Rich multi-media support.	Yes	Cog[6] & Exupery	Smalltalk/Slang [7]	110k Smalltalk, ~300K C
TaoGroup VP/VP2	C, Java	Proprietary embedded VM				
TraceMonkey	JavaScript	Based on Tamarin	No	Yes	C++	173k
Translator Engine	Flat File Tables/Global C++ variable declarations	IDE, programming by demonstration				
TrueType	TrueType	Font rendering engine	Yes	No	C (typically)	
Valgrind	x86/x86-64 binaries	Checking of memory accesses and leaks under Linux				

VisualWorks	Smalltalk		No	Yes	C	
VMKit [8]		JVM and CLI virtual machine based on LLVM.	No	Yes		
Vx32 virtual machine		Application-level virtualization for native code				
Waba		Virtual machine for small devices, similar to Java				
Yet Another Ruby VM (YARV)	Ruby	Virtual machine of the reference implementation for Ruby 1.9 and newer versions				
Z-machine	Z-Code					
Zend Engine	PHP		Yes	No	C	75k
libJIT Library for Just-In-Time compilation	Common Intermediate Language Java bytecode Domain-specific programming language	Virtual machine is used in Portable.NET Just-In-Time compiler, ILDJIT, HornetsEye	Yes	Yes	C, ia32, arm, amd64, alpha, low-level CPU architecture specific machine code	

References

[1] http://developer.apple.com/mac/library/documentation/Java/Conceptual/Java14Development/00-Intro/JavaDevelopment.html
[2] http://doc.cat-v.org/inferno/4th_edition/dis_VM_specification
[3] http://www.is-research.de/info/vmlanguages/
[4] http://java.sun.com/javase/
[5] http://luajit.org/
[6] http://www.mirandabanda.org/cog/
[7] http://wiki.squeak.org/squeak/2267
[8] http://vmkit.llvm.org/

- "libJIT vs LLVM discussion" Rhys Weatherley (libJIT) and Chris Lattner (LLVM) (http://lists.gnu.org/archive/html/dotgnu-libjit/2004-05/index.html)

External links

- List of Java Virtual Machines (JVMs), Java Development Kits (JDKs), Java Runtime Environments (JREs) (http://java-virtual-machine.net/other.html)

Comparison of platform virtual machines

Platform virtual machines are software packages which emulate the whole physical computer machine, often giving multiple virtual machines on one physical platform. The table below compares basic information about platform virtual machine (VM) packages.

General information

Name	Creator	Host CPU	Guest CPU	Host OS(s)	Guest OS(s)	License
Bochs	Kevin Lawton	Any	x86, x86-64	Linux, IRIX, AIX, FreeBSD, OpenBSD, BeOS, Mac OS X	DOS, Windows, Windows Mobile, xBSD, Linux, OS/2	LGPL
Containers, or Zones	Sun Microsystems	x86, x86-64, SPARC (portable: not tied to hardware)	Same as host	Solaris 10, Solaris 11 Express, OpenSolaris 2009.06	Solaris (8, 9, 10, 11), Linux (BrandZ)	CDDL
Cooperative Linux	Dan Aloni, other developers (1)	x86[1]	Same as host	Windows 2000, XP, 2003, Vista[1]	Linux	GPL version 2
Denali	University of Washington	x86	x86	Denali	Ilwaco, NetBSD	?
DOSBox	Peter Veenstra, Sjoerd, community help	Any	x86	Linux, Windows, Mac OS classic, Mac OS X, BeOS, FreeBSD, OpenBSD, Solaris, QNX, IRIX, MorphOS, AmigaOS, Maemo	Internally emulated DOS shell; classic PC booter games, unofficially Windows 1.0 to 3.11	GPL
DOSEMU	Community project	x86, x86-64	x86	Linux	DOS	GPL version 2
FreeBSD Jail	FreeBSD	Any running FreeBSD	Any running FreeBSD	FreeBSD	FreeBSD, Linux ABI	BSD
GXemul	Anders Gavare	Any	ARM, MIPS, M88K, PowerPC, SuperH	Unix-like	NetBSD, OpenBSD, Linux, Ultrix, Sprite	BSD
Hercules	Begun by Roger Bowler, as of 2011 maintained by Jay Maynard	Any	z/Architecture	Unix-like	Linux on zSeries, z/OS, z/VM, z/VSE, OS/360, DOS/360, DOS/VS, MVS, VM/370, TSS/370	QPL

Hyper-V Server 2008 R2	Microsoft	x86-64 + hardware-assisted virtualization (Intel VT-x or AMD-V)	x86-64, x86 (up to 8 physical CPUs)	Windows 2008 w/Hyper-V Role, Windows Hyper-V Server	supported drivers for Windows 2000, Windows 2003, Windows 2008, Windows XP, Windows Vista, Linux (SUSE 10 released, more announced)	Proprietary
iCore Virtual Accounts	iCore Software	x86	x86	Windows XP	Windows XP	Proprietary
Integrity Virtual Machines	Hewlett-Packard	IA-64	IA-64	HP-UX	HP-UX, Windows, Linux (OpenVMS announced)	Proprietary
JPC (Virtual Machine)	Oxford University	Any running the Java Virtual Machine	x86	Java Virtual Machine	DOS, Linux, Windows up to 3.0	GPL version 2
KVM	Qumranet [2]	x86, x86-64, IA-64, with x86 virtualization, s390, PowerPC (see restrictions, e.g., no PowerPC 970 support [3])	Same as host	Linux	FreeBSD, Linux, Solaris, Windows	GPL version 2
Linux-VServer	Community Project	x86, x86-64, IA-64, Alpha, PowerPC 64, PA-RISC 64, SPARC64, ARM, S/390, SH/66, MIPS	Compatible	Linux	Linux variants	GPL version 2
LynxSecure	LynuxWorks	x86, Intel VT-x, Intel VT-d	x86	No host OS	LynxOS, Linux, Windows	Proprietary
lxc	Community project lxc.sourceforge.net [4]	x86, x86-64, IA-64, PowerPC 64, SPARC64	Same as host	Linux	Linux variants	GPL version 2
Mac-on-Linux	Mac-on-Linux [5]	PowerPC	PowerPC	Linux	Mac OS X, Mac OS 7.5.2 to 9.2.2, Linux	GPL
Mac-on-Mac	Sebastian Gregorzyk	PowerPC	PowerPC	Mac OS X, up to Tiger excluded	Mac OS X, Mac OS 7.5.2 to 9.2.2, Linux	GPL
OpenVZ	Community project, supported by SWsoft, now Parallels, Inc.	x86, x86-64, IA-64, PowerPC 64, SPARC64	Same as host	Linux	Linux variants	GPL
Oracle VM for x86	Oracle Corporation	x86, x86-64, Intel VT-x, AMD-V	x86, x86-64, Intel VT-x, AMD-V	No host OS	Microsoft Windows, Oracle Enterprise Linux, Red Hat Enterprise Linux, Solaris	Oracle VM Server GPL; Manager proprietary
Oracle VM Server for SPARC (LDoms)	Oracle Corporation	UltraSPARC T1, UltraSPARC T2, UltraSPARC T2+, SPARC T3	Compatible	Solaris 10	Solaris, Linux, FreeBSD	Proprietary

OVPsim	OVP [6]	x86	OR1K, MIPS32, ARC600/700, ARM; and public API which enables users to write custom processor models, RISC, CISC, DSP, VLIW all possible	Microsoft Windows, Linux	Depends on target machine, for example includes MIPS Malta that runs Linux or SMP-Linux; and includes public API which enables users to write custom peripheral and system models	Apache 2.0
Parallels Desktop for Mac	Parallels, Inc.	x86, Intel VT-x	x86	Mac OS X x86	DOS, Windows, Linux, Mac OS X Server, FreeBSD, OS/2, eComStation, Solaris	Proprietary
Parallels Workstation	Parallels, Inc.	x86, Intel VT-x	x86	Windows, Linux	Windows, Linux, FreeBSD, OS/2, eComStation, DOS, Solaris	Proprietary
PearPC	Sebastian Biallas	x86, x86-64, PowerPC	PowerPC	Windows, Linux, Mac OS X, FreeBSD, NetBSD	Mac OS X, Darwin, Linux	GPL
PikeOS	SYSGO AG	PowerPC, x86, ARM, MIPS, SPARC, SuperH	Same as host	PikeOS	PikeOS native, Linux, RTEMS, OSEK, ARINC 653 APEX, ITRON	Proprietary
PowerVM	IBM	POWER4, POWER5, POWER6, PowerPC 970	POWER4/5/6, PowerPC 970, X86 (PowerVM-Lx86)	No host OS	Linux PowerPC, x86; AIX, i5/OS, IBM i	Proprietary
QEMU	Fabrice Bellard, other developers	x86, x86-64, IA-64, PowerPC, Alpha, SPARC 32/64, ARM, S/390, M68k	x86, x86-64, ARM, CRIS, LM32, MicroBlaze, MIPS, SPARC 32/64, PowerPC	Windows, Linux, Mac OS X, Solaris, FreeBSD, OpenBSD, BeOS	Changes regularly[7]	GPL/LGPL
QEMU w/ kqemu module	Fabrice Bellard	x86, x86-64	Same as host	Linux, FreeBSD, OpenBSD, Solaris, Windows	Changes regularly[7]	GPL/LGPL
QEMU w/ qvm86 module	Paul Brook	x86	x86	Linux, NetBSD[8], Windows	Changes regularly	GPL
QuickTransit	Transitive Corp.	x86, x86-64, IA-64, POWER	MIPS, PowerPC, SPARC, x86	Linux, Mac OS X, Solaris	Linux, Mac OS X, Irix, Solaris	Proprietary
RTS Hypervisor	Real-Time Systems	x86	x86	No host OS	Windows XP, XP Embedded, Windows CE, Linux, Android, VxWorks, OS-9, RTOS-32, QNX, proprietary	Proprietary

SIMH	Bob Supnik, The Computer History Simulation Project [9]	Alpha, ARM, HPPA, x86, IA-64, x86-64, M68K, MIPS, MIPSel, POWER, s390, SPARC	Data General Nova, Eclipse; Digital Equipment Corporation PDP-1, PDP-4, PDP-7, PDP-8, PDP-9, PDP-10, PDP-11, PDP-15, VAX; GRI Corporation GRI-909, IBM 1401, 1620, 1130, 7090/7094, System 3; Interdata (Perkin-Elmer) 16b/32b systems; Hewlett-Packard 2114, 2115, 2116, 2100, 21MX; Honeywell H316/H516; MITS Altair 8800 with 8080 and Z80; Royal-Mcbee LGP-30, LGP-21; Scientific Data Systems SDS 940	BSD, Linux, Solaris, VMS, Windows	Depends on target machine, includes NetBSD/VAX, OpenBSD/VAX, VAX/VMS, Unix v6, Unix v7, TOPS-10, TOPS-20, ITS	BSD-like, unique
Simics	Virtutech [10]	x86, x86-64, SPARC v9	Alpha, ARM, IA-64, MIPS 32/64, MSP430, POWER, PowerPC 32/64, SPARC v8/v9, x86, x86-64, TI TMS320C64xx	Windows, Linux, Solaris	Depends on target machine, VxWorks, OSE, QNX, Linux, Solaris, Windows, FreeBSD, RTEMS, TinyOS, many others	Proprietary
Sun xVM Server	Sun Microsystems	x86-64, SPARC	Same as host	No host OS	Windows XP, 2003 Server (x86-64 only), Linux, Solaris	GPL version 3
SVISTA 2004	Serenity Systems International [11]	x86	x86	Windows, OS/2, Linux	Windows, Linux, OS/2, BSD	Proprietary
TRANGO	TRANGO Virtual Processors, Grenoble, France [12]	ARM, XScale, MIPS, PowerPC	Paravirtualized ARM, MIPS, PowerPC	No host OS, Linux or Windows as dev. hosts	Linux, eCos, µC/OS-II, WindowsCE, Nucleus, VxWorks	Proprietary
User Mode Linux	Jeff Dike, other developers	x86, x86-64, PowerPC	Same as host	Linux	Linux	GPL version 2
VirtualBox	Innotek, acquired by Oracle	x86, x86-64	x86, (x86-64 only on VirtualBox 2 and later with hardware virtualization)	Windows, Linux, Mac OS X x86, Solaris, FreeBSD, eComStation	DOS, Linux, Mac OS X Server,[13] FreeBSD, Haiku, OS/2, Solaris, Syllable, Windows	GPL version 2; full version with extra enterprise features is proprietary
Virtual Iron 3.1	Virtual Iron Software, Inc., acquired by Oracle	x86 VT-x, x86-64 AMD-V	x86, x86-64	No host OS	Windows, Linux	Full product is proprietary, a few components are GPL version 2[14]

Virtual PC 2007	Connectix	x86, x86-64	x86	Windows Vista (Business, Enterprise, Ultimate), XP Pro, XP Tablet PC Edition	DOS, Windows, OS/2, Linux (SUSE, Xubuntu), OpenSolaris (Belenix)	Proprietary
Windows Virtual PC	Connectix	x86, x86-64 with Intel VT-x or AMD-V	x86	Windows 7	Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2008	Proprietary
Virtual PC 7 for Mac	Connectix	PowerPC	x86	Mac OS X	Windows, OS/2, Linux	Proprietary
VirtualLogix VLX	VirtualLogix	ARM, TI DSP C6000, x86, Intel VT-x, Intel VT-d, PowerPC	Same as host	No host OS	Linux, Windows XP, C5, VxWorks, Nucleus, DSP/BIOS, proprietary	Proprietary
Virtual Server 2005 R2	Connectix	x86, x86-64	x86	Windows 2003, XP	Windows NT, 2000, 2003, Linux (Red Hat, SUSE)	Proprietary
CoWare Virtual Platform	CoWare	x86, x86-64, SPARC v9	Devices including (multi) cores from ARM, MIPS, PowerPC, Toshiba MeP, Renesas SH, TI, Tensilica, ZSP	Windows, Linux, Solaris	Depends on guest CPU; includes: Linux (various flavors), mITRON (various flavors), Windows CE, Symbian, more	Proprietary
Virtuozzo	SWsoft, now Parallels, Inc.	x86, IA-64, x86-64	x86, IA-64, x86-64	Linux, Windows	Linux, Windows	Proprietary
VMware ESX Server	VMware	x86, x86-64	x86, x86-64	No host OS	Windows, Linux, Solaris, FreeBSD, OSx86 (as FreeBSD), virtual appliances, Netware, OS/2, SCO, BeOS, Darwin, others: runs arbitrary OS[15]	Proprietary
VMware ESXi	VMware	x86, x86-64	x86, x86-64	No host OS	Same as VMware ESX Server	Proprietary
VMware Fusion	VMware	x86, x86-64	x86, x86-64	Mac OS X x86	Same as VMware ESX Server	Proprietary
VMware Server	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary
VMware Workstation 7.1	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary
VMware Player 3.1	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary

Wind River hypervisor	Wind River	x86, PowerPC	Same as host	No host OS	Linux, VxWorks, unmodified guests (including MS Windows and RTOSes such ach OSE, QNX and others), bare metal virtual board	Proprietary
Wind River VxWorks MILS Platform	Wind River	PowerPC	Same as host	No host OS	VxWorks, bare metal virtual board	Proprietary
Xen	Xensource	x86, x86-64, IA-64	Same as host, up to 128 physical CPUs	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP & 2003 Server (needs vers. 3.0 and an Intel VT-x (Vanderpool) or AMD-V (Pacifica)-capable CPU), Plan 9	GPL
XtratuM	Universidad Politecnica de Valencia	x86, x86; SPARC v8 LEON2/3	Same as host	No host OS	GPOS: Linux, RTOS: PartiKle, RTEMS	GPL
z/VM	IBM	z/Architecture	z/Architecture, z/VM does not run on predecessor mainframes	No host OS, itself (single or multiple levels/versions deep, e.g. VM/ESA running in z/VM 4.4 in z/VM 5.2 in z/VM 5.1.)	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, VM/CMS, MUSIC/SP, OpenSolaris for System z, predecessors	Proprietary
z LPARs	IBM	z/Architecture	z/Architecture	Integrated in firmware of System z mainframes	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, MUSIC/SP, and predecessors	Integrated in firmware of System z mainframes
Name	**Creator**	**Host CPU**	**Guest CPU**	**Host OS(s)**	**Guest OS(s)**	**License**

Features

Name	Guest OS SMP available	Runs arbitrary OS	Supported guest OS drivers	Method of operation	Typical use	Speed relative to host OS	Commercial support available
Containers, or Zones	Yes, over 500-way on current systems	No	Uses native device drivers	Operating system-level virtualization	Server consolidation with workload isolation, single workload containment, hosting, dev/test/prod	Native	Yes

Hyper-V Server 2008 R2	Yes, up to 4 VCPUs per VM	Yes	Yes	Virtualization	Server consolidation, service continuity, dev/test	Near native	Yes
OpenVZ	Yes	No	Compatible	Operating system-level virtualization	Virtualized server isolation	Native[osvirt]	?
KVM	Yes [16], up to 16 VCPUs per VM	Yes	?	AMD-V and Intel-VT-x	?	?	Yes, Look at RedHat [17] or Novell [18] for details
Linux-VServer	Yes	No	Compatible	Operating system-level virtualization	Virtualized server isolation	Native[osvirt]	Yes
Oracle VM Server for x86	Yes	Yes	Yes	Paravirtualization and hardware virtualization	Server consolidation and security, enterprise and business deployment	Near native	Yes
Oracle VM Server for SPARC	Yes	Yes	Yes	Paravirtualization and hardware virtualization	Server consolidation and security, enterprise and business deployment	Near native	Yes
OVPsim	Yes	Yes	Yes, but most of the time unmodified is the goal	Full system simulation with optional component virtualization	Software development (early, embedded), advanced debug for single and multicore software, compiler and other tool development, computer architecture research, hobbyist	Depends on target, up to 500% faster than embedded target, runs over 1,000 MIPS on desktop	Yes, with commercial license from Imperas[19]
PikeOS	Yes	Yes, but modifications required as paravirtualization is used	Yes	Paravirtualization	Safety and security critical embedded systems.	Near native	Yes

RTS Hypervisor	Yes	Yes	Compatible. OS drivers run unmodified, no special drivers required	Two modes: full virtualization and paravirtualization; both modes may be used for different operating systems concurrently	x86 based devices: vertical markets include robotics, industrial automation, medical, telecom, test and measurement; focus is on real-time uses	Native	Yes
Sun xVM Server	Yes	Yes	Yes	Paravirtualization and porting or hardware virtualization	Servers	Up to near native speed[native]	Yes
SVISTA 2004	No	?	?	?	Hobbyist, Developer, Business workstation	?	?
TRANGO	Yes	Yes[patch]	Yes	Paravirtualization and porting or hardware virtualization	Mob. phone, STB, routers, etc.	Native[native]	?
User Mode Linux	???	No	special guest kernel+modules required	Porting	used as a separate machine for a server or with X11 networking	near Native [native_speed] (Runs slow as all calls are proxied)	?
Oracle VirtualBox	Yes	Yes	Yes	Virtualization	Business workstation, server consolidation, service continuity, developer, hobbyist	Near native	Yes (with commercial license)
Virtual Iron 3.1	Yes, up to 8 way	Yes	Yes	Native virtualization	Server consolidation, service continuity, dev/test	Near Native	Yes
Virtual PC 2007	No	Yes	Yes	Virtualization, guest calls trapping where supported	Hobbyist, Developer, Business workstation	Near native with virtual machine additions	?
Windows Virtual PC	Yes	Yes	Yes	Hardware virtualization	Developer, Business workstation, support for Compatibility with Windows XP applications	Near native with virtual machine additions	No

Virtual PC 7 for Mac	No	Yes	Yes	dynamic recompilation (guest calls trapping where supported)	Hobbyist, Developer, Business workstation	Slow	?
Virtual Server 2005 R2	No	Yes	Yes	Virtualization (guest calls trapping where supported)	Server, server farm	Near native with virtual machine additions	?
CoWare Virtual Platform	Yes	Yes	Yes (Same compiled Software image as for the real device)	Full-system virtualization (Processor Core ISA + Hardware + External connections)	Early embedded software development and integration (from driver to application), Multi-core software debugging and optimization	Depending on the system characteristics and the software itself, ranges from faster than real time to slow.	Yes
Virtuozzo	Yes	No	Compatible	Operating system-level virtualization	Server consolidation, service continuity, disaster recovery, service providers	Native native_speed	Yes
VMware ESX Server 4.0 (vSphere)	Yes, add-on, up to 8 way	Yes	Yes	Virtualization	Server consolidation, service continuity, dev/test, cloud computing	Up to near native	Yes
VMware ESX Server 3.0	Yes, add-on, up to 4 way	Yes	Yes	Virtualization	Server consolidation, service continuity, dev/test	Up to near native	Yes
VMware ESX Server 2.5.3	Yes, add-on, 2 way	Yes	Yes	Virtualization	Server consolidation, service continuity, dev/test	Up to near native	Yes
VMware Fusion	Yes	Yes	Yes	Virtualization	Hobbyist, Developer, Tester, Business workstation	Near native	Yes
VMware Server	Yes (2-way)	Yes	Yes	Virtualization	Server/desktop consolidation, dev/test	Up to near native	Yes

VMware Workstation 6.0	Yes (2-way)	Yes	Yes	Paravirtualization (VMI) and virtualization	Technical professional, advanced dev/test, trainer	Up to near native	Yes
VMware Player 2.0	Yes (2-way [20])	Yes	Yes	Virtualization	Technical professional, advanced dev/test, trainer, end user on prebuilt machines	Up to near native	Yes
Wind River hypervisor	No	Yes	Yes	Paravirtualization, hardware assisted virtualization	Embedded, safety critical, secure	Native	Yes
Wind River VxWorks MILS Platform	No	Yes	Yes	Paravirtualization, hardware assisted virtualization	Embedded, safety critical, secure	Native	Yes
Xen	Yes, v4.0.0: up to 128 VCPUs per VM	Yes	Not required with the exception of the networking drivers where a NAT is required. A modified guest kernel or special hardware level abstraction is required for guest OSs.	Paravirtualization and porting or hardware virtualization	Server/desktop consolidation, dev/test	Up to near native speed,[native] substantial performance loss on some workloads (network and disk intensive especially)	Yes
XtratuM	Yes	No	Yes, but not required.	Paravirtualization	Embedded, safety critical, secure	Native (overhead lower than 1%)	Yes
z/VM	Yes, both real and virtual (guest perceives more CPUs than installed), incl. dynamic CPU provisioning and reassignment	Yes	Yes, but not required	Virtualization (among first systems to provide hardware assists)	Servers	Near Native[zvm_performance]	Yes

z LPARs	Yes, both real and virtual (guest perceives more CPUs than installed), incl. dynamic CPU provisioning and reassignment; up to 64 real cores	Yes	Yes, but not required	Microcode and hardware hypervisor	Servers	Native: System z machines always run with at least one LPAR	Yes
Name	**Guest OS SMP available**	**Runs arbitrary OS**	**Supported guest OS drivers**	**Method of operation**	**Typical use**	**Speed relative to host OS**	**Commercial support available**

- Providing any virtual environment usually requires some overhead of some type or another. Native usually means that the virtualization technique does not do any CPU level virtualization (like Bochs), which executes code more slowly than when it is directly executed by a CPU. Some other products such as VMWare and Virtual PC use similar approaches to Bochs and QEMU, however they use a number of advanced techniques to shortcut most of the calls directly to the CPU (similar to the process that JIT compiler uses) to bring the speed to near native in most cases. However, some products such as coLinux, Xen, z/VM (in real mode) do not suffer the cost of CPU-level slowdowns as the CPU-level instructions are not proxied or executing against an emulated architecture since the guest OS or hardware is providing the environment for the applications to run under. However access to many of the other resources on the system, such as devices and memory may be proxied or emulated in order to broker those shared services out to all the guests, which may cause some slow downs as compared to running outside of virtualization.
- OS-level virtualization is described as "native" speed, however some groups have found overhead as high as 3% for some operations, but generally figures come under 1%, so long as secondary effects do not appear.
- See [21] for a paper comparing performance of paravirtualization approaches (e.g. Xen) with OS-level virtualization
- Requires patches/recompiling.
- Exceptional for lightweight, paravirtualized, single-user VM/CMS interactive shell: largest customers run several thousand users on even single prior models. For multiprogramming OSes like Linux on zSeries and z/OS that make heavy use of native supervisor state instructions, performance will vary depending on nature of workload but is near native. Hundreds into the low thousands of Linux guests are possible on a single machine for certain workloads.

Other features

Name	Can boot an OS on another disk partition as guest	USB support	GUI	Live memory allocation	3D acceleration	Snapshots per VM	Snapshot of running system	Live migration	Shared folders	Shared clipboard
KVM	Yes	Yes	Yes[22]	Yes		?	Yes[23]	Yes[24]		
User Mode Linux	Yes	No	No	No	No			No	Yes	N/A
Oracle VirtualBox-OSE (GPLv2)	Partial (since version 1.4, but unsupported) VirtualBox-partitions	USB 1.1	Yes	Yes	OpenGL 2.0[25]	Yes branched[26]	Yes	Yes	Yes	Yes
Oracle VirtualBox-PUEL (pre-compiled, not free)	Partial (since version 1.4, but unsupported) VirtualBox-partitions	USB 2.0	Yes	Yes	OpenGL 2.0[25]	Yes branched[26]	Yes	Yes	Yes	Yes
Virtual Iron 4.2								Yes		
Virtual PC 2007	No	No	Yes	No	No			No	Yes	Yes
Windows Virtual PC	No	partially	Yes							
VirtualPC 7 for Mac	Yes	Yes	Yes	Yes	No			No		
Microsoft Virtual Server 2005 R2		No	Yes	No	No	?	Yes	No		
Microsoft Hyper-V Server 2008 R2	Yes	Partial support over remote desktop connections Hyper-V-USBRedirection	Yes	Yes	DirectX 9.0c Hyper-V-3DAcceleration	branched	Yes	Yes		
Virtuozzo	Yes	Yes	Yes	Yes	No			Yes		
VMware ESX Server 3.0 atp			Yes		No	?	Yes	Yes		
VMware ESX Server 2.5.3			Yes		No					
VMware ESX Server 4.0 (vSphere)	Yes	Yes	Yes	Yes	Yes	?	Yes	Yes		
VMware Fusion 2.0	Yes	Yes	Yes	No	DirectX 9 Shader model 2			No		
VMware Server	Yes	Yes	Yes	Yes	No	1	Yes	No	Yes	Yes
VMware Workstation 5.5	Yes	Yes	Yes	Yes	Experimental support for DirectX 8; also supported with VMGL[27]	Yes branched	Yes	No	Yes	Yes

VMware Workstation 6.0	Yes	Yes	Yes	Yes	Experimental support for DirectX 8; Also supported with VMGL[27]	Yes branched	Yes	No	Yes	Yes
VMware Workstation 7.0	Yes	Yes	Yes	Yes	Support for DirectX 9.0c Shader Model 3 and OpenGL 2.13D.[28]	Yes branched	Yes	No	Yes	Yes
VMware Player	No	Yes	Yes	Yes	supported with VMGL[27]	No	No	No	Yes	
Wind River hypervisor	Yes	Yes	Yes	Yes	Yes			No		
Wind River VxWorks MILS Platform	Yes									
Xen	Yes		Yes[22]	Yes	Supported with VMGL[27]	?	Yes	Yes		
z/VM	Yes	Not applicable	with add-ons	Yes	No			with GDPS		
z LPARs	Yes	Not applicable	Yes	Yes	No			with GDPS		
Zones	Yes	Yes	Yes	Yes	No					
Name	**Can boot an OS on another disk partition as guest**	**USB**	**GUI**	**Live memory allocation**	**3D acceleration**	**Snapshots per VM**	**Snapshot of running system**	**Live migration**	**Shared folders**	**Shared clipboard**

- VirtualBox User Manual, Chapter 9.9; requires usage of `VBoxManage internalcommands createrawvmdk` which says:*This is a development tool and shall only be used to analyse problems. It is completely unsupported and will change in incompatible ways without warning.*
- Windows Server 2008 R2 SP1 and Windows 7 SP1 have limited support for redirecting the USB protocol over RDP using RemoteFX.[29]
- Windows Server 2008 R2 SP1 adds accelerated graphics support for certain editions of Windows Server 2008 R2 SP1 and Windows 7 SP1 using RemoteFX.[30] [31]

Restrictions

This table is meant to outline restrictions in the software dictated by licensing or capabilities.

Name	Maximum host physical processors (sockets)	Maximum host cores per processor	Maximum host memory	Maximum host disk volume size	Maximum number of guest VM running	Maximum number of logical CPU per VM guest	Maximum amount of memory per VM guest, 32/64 bit	Maximum number of SCSI + IDE disks per VM guest	Maximum disk size per VM guest
Containers, or Zones	No limit (Solaris supports system with 72 CPU sockets)	No limit	No limit (Solaris supports current systems with 4TB RAM)	No limit	8,191	No limit	No limit	No limit	No limit
VMware Player 3.1[32]	?	?	No limit	N/A	?	8	8/32 GB	?	2 TB
VMware Server 2.0[33]	Varies, max is # sockets * # cores = 16	16	No limit	N/A	16 * 4 = 64	2	8/8 GB	60 + 4 = 64	950 GB
VMware vSphere Hypervisor (ESXi) 4.1[34]	Varies, max logical is 160	160	1 TB	2 TB minus 512 bytes	320 VMs	8	255/255 GB	60 + 4 = 64	2 TB minus 512 bytes
VMware vSphere ESXi 4.1[35]	Varies, max logical is 160	160	1 TB	2 TB minus 512 bytes	320 VMs	8	255/255 GB	60 + 4 = 64	2 TB minus 512 bytes
VirtualBox 4.0.x	No limit ?	No limit ?	No limit ?	No limit	No limit[36]	16	No limit	4 IDE; no limit SATA, SCSI, SAS	2 TB
Hyper-V 2008 R2	64	(same 64)	1 TB	No limit	384 VMs	4	64 GB	4 IDE; 256 SCSI	2 TB

Note: No limit means no enforced limit. For example, a VM with 1 TB of memory cannot fill a host with only 8 GB memory, and no memory swap disk, so it will have a limit of 8 GB physically.

Hyper-V limit source: http://technet.microsoft.com/en-us/library/ee405267(WS.10).aspx

References

[1] "Cooperative Linux FAQ" (http://colinux.wikia.com/wiki/FAQ). Retrieved on 2009-01-27.
[2] http://kvm.qumranet.com
[3] http://www.linux-kvm.org/page/PowerPC
[4] http://lxc.sourceforge.net/
[5] http://www.maconlinux.org/
[6] http://www.ovpworld.org
[7] QEMU Official OS Support List (http://www.claunia.com/qemu)
[8] http://pkgsrc.se/wip/qemu-qvm86
[9] http://simh.trailing-edge.com/
[10] http://www.virtutech.com/
[11] http://www.serenityvirtual.com/
[12] http://www.trango-vp.com
[13] Oracle VM VirtualBox® User Manual, Chapter 3: Configuring virtual machines | Mac OS X Server guests (http://www.virtualbox.org/manual/ch03.html#intro-macosxguests)
[14] Oracle and Virtual Iron (http://www.oracle.com/us/corporate/Acquisitions/virtualiron/)

[15] Can run a guest OS without modifying it, and hence is generally able to run any OS that could run on a physical machine the VM simulates
[16] http://www.linux-kvm.com/content/running-windows-smp-guests
[17] http://www.redhat.com/virtualization/rhev/desktop/rhevm/
[18] http://www.novell.com/linux/products.html#linuxvirtualization
[19] Imperas (http://www.imperas.com)
[20] http://www.vmware.com/products/player/features.html#c6062
[21] http://www.cs.princeton.edu/~mef/research/vserver/paper.pdf
[22] "Virtual Machine Manager" (http://virt-manager.et.redhat.com/). . Retrieved 2010-02-20.
[23] "Sheepdog is a distributed storage system for KVM" (http://www.osrg.net/sheepdog/). . Retrieved 2010-05-20.
[24] "KVM Migration" (http://www.linux-kvm.org/page/Migration). . Retrieved 2010-05-20.
[25] "VirtualBox Changelog" (http://www.virtualbox.org/wiki/Changelog-3.0). . Retrieved 2009-06-30.
[26] "VirtualBox Changelog 3.1" (http://www.virtualbox.org/wiki/Changelog-3.1). . Retrieved 2010-10-01.
[27] "[[VMGL (http://www.cs.toronto.edu/~andreslc/xen-gl)] (formerly Xen-GL)"]. .
[28] http://www.vmware.com/products/workstation/new.html
[29] http://technet.microsoft.com/en-us/library/ff817581(WS.10).aspx
[30] http://technet.microsoft.com/en-us/library/ff817578(WS.10).aspx
[31] http://technet.microsoft.com/en-us/library/ff817602(WS.10).aspx
[32] (http://www.vmware.com/pdf/vmware_player310.pdf) Getting Started Guide VMware Player 3.1
[33] (http://www.vmware.com/pdf/vmserver2.pdf) VMware Server User's Guide VMware Server 2.0
[34] (http://www.vmware.com/pdf/vsphere4/r41/vsp_41_config_max.pdf) Configuration Maximums VMware® vSphere 4.1
[35] (http://www.vmware.com/pdf/vsphere4/r41/vsp_41_config_max.pdf) Configuration Maximums VMware® vSphere 4.1
[36] (http://www.virtualbox.org/manual/ch01.html) Oracle VM VirtualBox User Manual. Accessed 2011-04-07

External links

- Technical comparison of Linux virtualization technologies (http://virt.kernelnewbies.org/TechComparison).
- Unix for Windows FAQ (http://www.unix.com/answers-frequently-asked-questions/16634-unix-environments-ms-windows.html) at Unix.com

Comparison of VMware Fusion and Parallels Desktop

Represented by their respective products, VMware and Parallels are the two major commercial competitors in the Mac consumer virtualization market. Both products are based on hypervisor technology and allow users to run an additional 32- or 64-bit x86 operating system in a virtual machine alongside Mac OS X on an Intel-powered Mac. The similarity in features and functionality between **VMware Fusion** and **Parallels Desktop for Mac** has given occasion for much comparison.

Features

Feature	Product	
	VMware Fusion 3.1	Parallels Desktop 6
Installing Windows on Mac		
Easy Install with Automatic Windows Setup	Yes	Yes
Run off existing Boot Camp partition	Windows XP (32-bit), Windows Vista (32-bit and 64-bit), Windows 7 (32-bit and 64-bit)	Windows XP (32-bit), Windows Vista (32-bit and 64-bit), Windows 7 (32-bit and 64-bit)
Allows to suspend VM running off the Boot Camp partition	No	Yes
Import Physical Windows PC to VM	Yes	Yes
Import Boot Camp partition to VM	Yes	Yes
Full Support for Windows 7	Yes	Yes
Import Third Party VMs (Parallels, VMware, VirtualPC for Mac)	Yes	Yes
Mac		
Support Higher Synthetic screen resolutions in Full-Screen mode without scrolling (e.g. Run Windows in 1920x1200 mode on a 1440x900 monitor)	Yes	No
Migrate Windows 2000 PC to Mac	Yes	Yes
Migrate Windows XP to Mac	Yes	Yes
Migrate Windows 7 to Mac	Yes	Yes
Migrate Linux to Mac	No	Yes
Migrate PC over wireless network	Yes	Yes
Migrate PC over Ethernet	Yes	Yes
Migrate PC over FireWire	Yes	Yes
Migrate PC over USB	No	Yes
Windows documents migration to appropriate folders to Mac	No	Yes
Migration of the Internet bookmarks and settings from Internet Explorer, Firefox, or Chrome on PC to the default browser on the Mac.	No	Yes
Comes with Video Tutorials to help Windows Switchers on Mac	No	Switch to Mac Edition
Running Windows Apps on Mac		
Run Windows apps like Mac apps (Unity/Coherence)	Yes	Yes
Exclude Dock in Unity/Coherence	Yes	Yes
Windows application folder in Dock	No	Yes
Windows Start Menu in Dock	No	Yes
Windows application folder in menu bar	Yes	No
Windows Start Menu in menu	Yes	Yes
Always On Application Menu available to launch Windows apps at any time	Yes	Yes
Quit Individual Window applications	Yes	Yes

Use Command ` to switch between open windows in a Windows app	Yes	Yes
Assign Windows applications to a Mac OS X Space	Yes	Doesn't keep windows from app together
Shared Folders to access Mac files/folders from Windows	Yes	Yes
Access Windows tray icons in Unity	Yes	Yes
Arrow icon in the Mac menu bar for customizing Windows tray icons in Coherence.	No	Yes
Progress for downloads and other operations is displayed on the Windows 7 applications icons in the Dock.	No	Yes
Windows 7 Jump Lists are supported for Windows applications in the Dock: view your recent documents by right-clicking the application icon in the Dock.	No	Yes
Grouping of all windows of a single Windows application under the same application icon in the Dock.	No	Yes
Active screen corners	No	Yes
Launch Windows or Mac internet applications from hyperlinks	Yes	Yes
Windows applications can be set as the default applications for handling CDs and DVDs inserted into the Mac.	No	Yes
Enable Apple Gestures to work with Windows applications	No	Yes
Enable Apple Magic Mouse Gesture support with Windows applications	No	Yes
Enable Apple Remote to work with Windows applications	No	Yes
Launch Mac applications from any Windows file (Shared Applications/SmartSelect)	Yes	Yes
Automatically mount storage and network devices to guest OS	Yes	Yes
Suspend/Resume To Where You Last Left Off	Yes	Yes
Pause VM if no application running to reduce Mac OS load.	No	Yes
Mirrored/Remapped Desktop, Music, Documents, and Pictures Folders	Yes	Yes
Contextual menu items "Run on Mac" and "Show in Finder" for Shared Folders	No	Yes
Remapping Windows Keyboard Shortcuts to Mac Shortcuts	Yes, completely customizable	Yes, completely customizable
Keyboard shortcut (F6 or Fn + F6) for hiding/showing Parallels Desktop and all its windows.	No	Yes
Virtual Links - aliases to Mac OS files from Windows virtual machines	Yes	Yes
Copy/Paste Text between Windows / Mac	Yes, Plain and Rich (RTF) Text up to 4MB	Yes, Plain and Rich (RTF) Text
Drag and Drop Text between Windows / Mac	Yes, Plain and Rich (RTF) Text up to 4MB	No
Drag and Drop Files Between Windows / Mac	Yes	Yes
Copy/Paste Graphics between Windows / Mac	Yes	Yes
Passthrough "Driverless" Printing to Mac Printers	Yes	Yes
Making Windows Safer on Mac		

Single Snapshot support	Yes	Yes
Multiple Snapshot support	Yes	Yes
AutoProtect Automatic Snapshots	Yes	Yes
TimeMachine backups can be synced with SmartGuard snapshots, to reduce the space required for backups.	No	Yes
Automatically revert VM to start state upon termination	No	Yes
Virtual machine encryption with AES algorithm for better security of your data (empowered by AES-NI hardware support on i5 and i7 CPUs)	No	Yes
Mac OS Parental Controls are automatically applied to the virtual machine for managing children's computer usage	No	Yes
Windows AntiVirus & AntiSpyware Included	12-month subscription to McAfee VirusScan Plus. User Prompted to Install	3-month subscription to Kaspersky Internet Security. User Prompted to Install
Mac OS AntiVirus Included	No	3-month subscription to Kaspersky Anti-Virus for Mac. User Prompted to Install
Lock down application and virtual machine settings to prevent changes	No	Yes
Isolated Virtual Machines	No	Yes
Designed for Mac (Fit and Finish)		
Optimized for Snow Leopard	Runs on 64-bit kernel, 64-bit VMM, 64-bit helper application, 32-bit GUI	Runs on 64-bit kernel, 64-bit VMM, 64-bit helper application, 32-bit GUI
UI Built from Ground Up for Mac	Cocoa and GTK+ frameworks	Nokia Qt library with Carbon and Cocoa frameworks.
Welcome Screen to Simplify Initial Setup	Yes	Yes
Mac-like Settings and Preferences Windows	Yes	Yes
Apply different colors to virtual machines to find them in Finder or in the virtual machines list	No	Yes
Customizable Toolbars	Yes	Yes
Default virtual machine to launch on application startup	Yes	Yes
Virtual Machine Library	Yes	Yes
Live View of Running Virtual Machines in Library/Settings	Yes	Yes
Live View of Running Virtual Machine in Dock	No	Yes
Startup in Last View Format	Yes	Yes
Reduce the virtual hard disk size as its unused space grows	Yes	Yes
Apple Help (Searchable with Mac Design)	Yes	Yes
Single File Virtual Machine Packages	Yes	Yes
QuickLook/Cover Flow Integration	Yes	Yes
Mount Running Virtual Machines in Finder	No	Yes
Mount Stopped Virtual Machines in Finder	Yes	Yes
Advanced Platform		
32-bit Guest Operating Systems	Yes	Yes

64-bit Guest Operating Systems	Yes	Yes
Multi-CPU virtual machines	Up to 8 processors per VM & Multi Core (as of VMWare Fusion 3.1)	Up to 8 processors per VM & Multi Core
Virtual CPU supports AES-NI when running on i5 and i7 CPUs	No	Yes
Maximum RAM per virtual machine	8 GB	8 GB
Support for IPv6, the next-generation version of the Internet Protocol, in shared, bridged and host-only networking.	No	Yes
Windows Aero (WDDM for Windows Vista and Windows 7)	Yes	Yes
DirectX 3D Graphics Support	DirectX 9.0c with Shader Model 3	DirectX 9.0Ex with Shader Model 3
OpenGL 3D Graphics Support for Windows Vista & Windows 7 guest OS	OpenGL 2.1 (as of VMWare Fusion 3.1)	OpenGL 2.1
OpenGL 3D Graphics Support for Windows XP guest OS	OpenGL 2.1	OpenGL 2.1
OpenGL 3d Graphics Support for Linux guest OS	No	Open GL 2.1
Multiple monitor/display support	Up to 10, each display is separate unique display to Windows	Up to 10, each display is separate unique display to Windows
Hardware Assisted Video/Movie Playback	Yes	Yes
Maximum video RAM per VM	288MB	256MB
USB 2.0 passthrough support	Yes	Yes
Support for external surround sound 5.1 USB or FireWire devices work simultaneously in Mac and the virtual machine.	No	Yes
64-bit Native Engine	64-bit VMM, 64-bit kernel extensions, 64-bit helper application, 32 GUI	64-bit VMM, 64-bit kernel extensions, 64-bit helper application, 32 GUI
Power Management/Battery Passthrough	Yes	Yes
Bluetooth support (as USB device)	Yes	Yes
Firewire passthrough support	No	No
Virtual IDE Controller	Yes	Yes
Virtual SCSI Controller	Yes	Yes
Virtual SAS Controller	Yes	Yes, using software emulation in guest OS
Shared Smart Card	Yes	Yes
Low Level Access to Network Interfaces (promiscuous mode / WiFi monitor mode etc.)	Yes	No
Advanced Tools		
Virtual Disk Management	Yes, Integrated in Settings Editor	Yes, Standalone tool
Resize Virtual Disks	Yes	Yes
Advanced Network Management	Requires modifying networking scripts	Yes
Control VMs with Scripting option	Yes	Yes
Supports Intel VT-x hardware virtualization engine	Yes	Yes
Network (PXE) Boot	Yes	Yes
Run More Operating Systems on Mac		

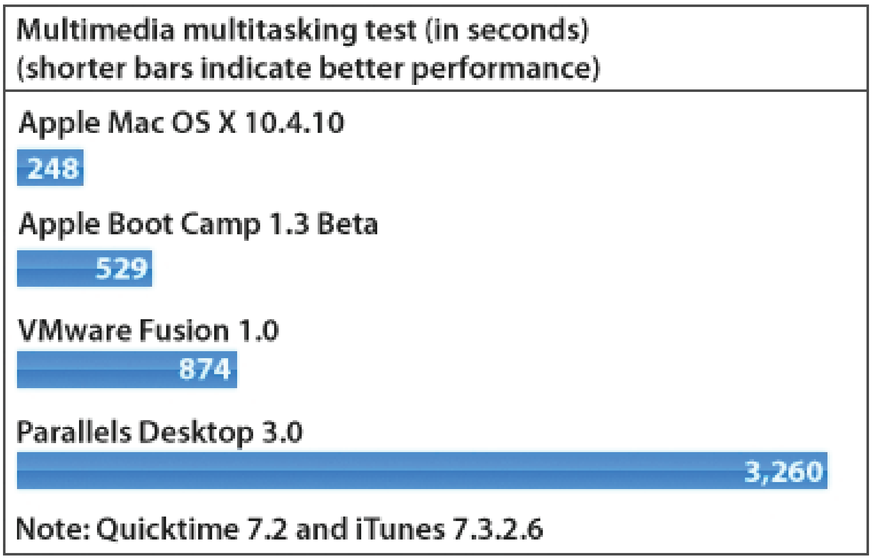
Run Multiple Operating Systems at One Time	Yes, run up to 20 with available memory	Yes
Supported x86 Operating Systems	Over 140 operating systems	Over 60 operating systems
Run Linux virtual machines	Yes	Yes
Linux Easy Install with Automatic Setup	Yes	Yes
Linux virtual machine support	Window resize, time sync, shared folders, drag and drop	Window resize, time sync, shared folders
Run Linux apps like Mac apps with Unity	Yes (Ubuntu and Red Hat with Gnome)	No
Run Mac OS X Server virtual machines	Yes	Yes
Import VHD Test Drives	Yes	Yes
Run Virtual Appliances	Over 1200 available	Fewer than 100 available
Complimentary iPhone/iPad mobile client		
Allows to work with remote VM on iPhone/iPad	No	Yes
Allows user to connect over internet	No	Yes

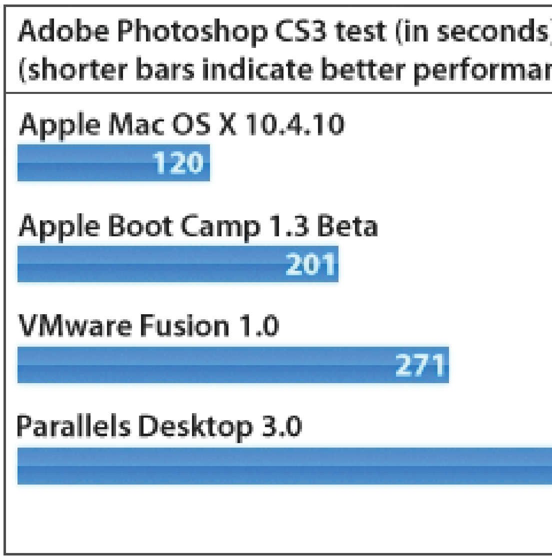
Minimum system requirements

Requirement	Product	
	VMware Fusion 3.1	**Parallels Desktop 6.0**
Host OS	Mac OS X 10.5.8 or later; Mac OS X 10.6 or later	Mac OS X 10.5.8 or later; Mac OS X 10.6 or later
RAM	1 GB (2 GB Recommended)	1 GB (2 GB Recommended)
Disk space for product	700 MB	450 MB

2007 Benchmark tests

On August 16, 2007, CNET published the results of several benchmarks[1] in which Fusion demonstrated better performance than Parallels Desktop for Mac in SMP-aware applications, which Fusion supports while Parallels does not. It should also be noted that Boot Camp is a tool for natively booting Windows XP on Intel Macintosh and is not a virtualization product. This comparison is of limited value today, as both products have had 2 major upgrades since then.





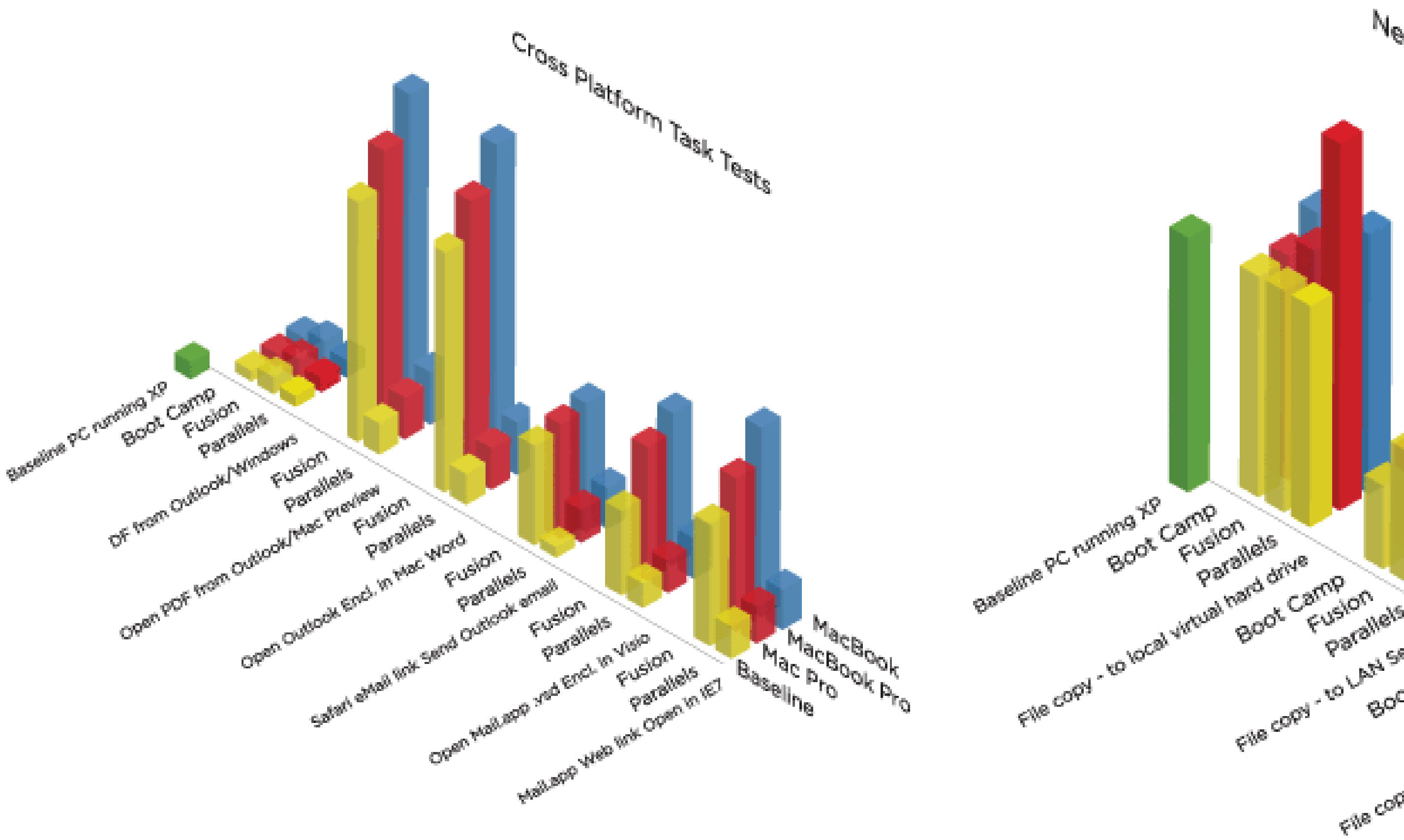
This comparison was tested on an eight-core, 2.66 GHz MacPro running Mac OS X 10.4.10, Parallels Desktop 3.0 for Mac (build 4560) and VMware Fusion 1.0 (build 51348). Fusion and Parallels were both set to 1,024 MB of system memory and a 32 GB hard disk. Fusion was set to 128 MB of graphics memory, and Parallels Desktop for Mac was set to 64 MB of graphics memory (the maximum for each at that time)[1] .

2008 Benchmark tests

In Volume 24, Issue 02 of MacTech, the editors published the results of one-step and task tests between VMware Fusion 1.0, Parallels Desktop 3.0 and Boot Camp and used a PC running Windows XP as a baseline comparison in a native PC environment.[2]

- One-step Test: After clicking the mouse or pressing a key, this test requires no further human action.
- Task Test: This tests the interaction between Mac OS X and the virtual environment and requires multiple tests throughout the process.

MacTech found that the faster the physical host computer, the more similarly Parallels Desktop and VMware Fusion performed. MacTech did not test multiple processor performance. The following graphs displays the results in seconds. Shorter bars indicate faster performance.



Each test was run on a MacBook (2 GB RAM; 1.83 GHz Core Duo processor), a MacBook Pro (4GB RAM; 2.16 GHz Core 2 Duo processor) and a Mac Pro (4GB RAM; Quad Core configuration with two 2.66 GHz Dual-Core Intel Xeon processors). MacTech tested Parallels Desktop 3.0 for Mac Build 5160 and VMware Fusion 1.0 Build 51348. All tests were done on clean host systems with new installations of Mac OS X 10.4.10 and Office installations and included all of the most up-to-date patches. No third party software was installed other than Mac OS X, VMware Fusion, Parallels Desktop, Windows XP, Windows Vista, Adobe Reader and Microsoft Office.

2009 Benchmark tests

In March, 2009, Volume 25, Issue 04, MacTech [3] published the results of a new series of benchmark tests that compared the performance between VMware Fusion 2.0.1 and Parallels Desktop 4.0 for Mac (build 3540), both running Mac OS X 10.5.5.

In most of MacTech's tests, Parallels Desktop performed 14-20% faster than Fusion; however, Fusion ran 10% faster than Parallels Desktop when running Windows XP 32-bit on 2 virtual processors.[4]

OS/Environment	Result
Windows XP, 32-bit, 1 Processor	Parallels Desktop runs 14% faster
Windows Vista, 32-bit, 1 Processor	Parallels Desktop runs 14% faster
Windows XP, 32-bit, 2 Processors	VMware Fusion runs 10% faster
Windows Vista, 32-bit, 2 Processors	Parallels Desktop runs 20% faster
Windows XP, 64-bit, 2 Processors	Parallels Desktop runs 15% faster

The tests were performed on the White MacBook, MacBook Pro, iMac and MacPro. Both Fusion and Parallels Desktop were optimized for virtual machine performance. MacTech's test included launch and CPU tests, File and Network IO, Footprint, Application Launch, Application Performance and 3D and HD Graphics. In many cases, tests were performed after both Adam and Successful launches and were timed using a stopwatch.

Test Suite	Performance Winner
Windows Launch Performance	Parallels Desktop for Mac
CPU	Parallels Desktop for Mac, except for 2 of the 14 tests
Footprint on Mac	Parallels Desktop for Mac
Application Launch	VMware Fusion
Application Performance	Both products did well, except for IE where Parallels Desktop is 80-91% faster[5]
3D and HD Performance	Dependent on game[1], video and Windows environment.

1 3D Games tested were Civilization IV: Colonization and Portal. In Civilization, Parallels Desktop has faster FPS (Frames Per Second) and performed better on slower machines while Fusion has better, more detailed graphics. Fusion has difficulty showing the startup video, but Parallels Desktop's graphics are not as rich. When running Portal, Fusion is faster but its graphics are visibly lighter, while Parallels Desktop has better graphics and visual details.[6]

Cross-platform task tests

MacTech's cross platform tests timed how long it took users to perform multi-step tasks that moved data between Mac OS X and Windows. VMware Fusion, which is designed for increased isolation from the host, requires more manual steps to move data between the host and the virtual environment. Parallels Desktop, which is designed to run transparently with the Mac OS X host, requires fewer steps to perform the same tasks. Therefore, Parallels Desktop was faster.

Networking and file I/O tests

Parallels Desktop occasionally displayed lag anomalies while VMware Fusion's virtual drive performance was very close to that of a physical drive. VMware Fusion preferred a bridged connection for reliable performance, and Parallels Desktop was consistent regardless of the type of virtual network adaptor used.

Simultaneous use of VM and host OS

Parallel Desktop 5 always uses wired memory for hosted OS, while VMWare Fusion 3.0 uses active memory that can be swapped. Giving better performance to hosted VM, this leaves less memory to host OS programs and causes more swapping if you use VM and host OS programs at the same time.

2010 Benchmark tests

In 2010 MacTech[7] , Volume 26, Issue 01, published the results of a new series of benchmark tests showing a performance advantage for Parallels Desktop 5 across all subcategories, with an average of 30% faster.

References

[1] Begun, Daniel (2007-08-17). "Inside CNET Labs: Windows virtual machine performance on the Mac" (http://crave.cnet.com/8301-1_105-9760910-1.html). CNET. . Retrieved 2007-10-04.

[2] Ticktin, Neil. "Virtualization Benchmarking How do Boot Camp, Parallels Desktop, and VMware Fusion stack up?" (http://www.mactech.com/articles/mactech/Vol.24/24.02/VirtualizationBenchmark/). MacTech. . Retrieved February 2008.

[3] Ticktin, Neil. "Head-to-Head: Parallels Desktop for Mac vs. VMware Fusion" (http://www.mactech.com/articles/mactech/Vol.25/25.04/VMBenchmarks/). MacTech. . Retrieved March 2009.

[4] http://www.mactech.com:16080/articles/mactech/Vol.25/25.04/VMBenchmarks/index-001.html

[5] http://www.mactech.com/articles/mactech/Vol.25/25.04/VMBenchmarks/index-002.html

[6] http://www.mactech.com/articles/mactech/Vol.25/25.04/VMBenchmarks/index-003.html

[7] Ticktin, Neil. "Head-to-Head: Parallels Desktop for Mac vs. VMware Fusion" (http://www.mactech.com/articles/special/1002-VirtualizationHeadToHead/index-001.html). MacTech. . Retrieved March 2010.

Adaptive Domain Environment for Operating Systems

Adeos (**Adaptive Domain Environment for Operating Systems**) is a nanokernel hardware abstraction layer (HAL) that operates between computer hardware and the operating system that runs on it.[1] It is distinct from other nanokernels, in that it is not just a low level layer for an outer kernel. Instead it is intended to run several kernels together, which makes it similar to virtualization technologies.

Adeos provides a flexible environment for sharing hardware resources among multiple operating systems, or among multiple instances of a single OS, thereby enabling multiple prioritized domains to exist simultaneously on the same hardware.

Adeos has been successfully inserted beneath the Linux kernel, opening a range of possibilities, such as SMP clustering, more efficient virtualization, patchless kernel debugging and real-time systems for Linux.

Unusually among HALs, Adeos can be loaded as a Linux loadable kernel module to allow another OS to run along with it. In fact Adeos was developed in the context of RTAI (Real-Time Application Interface) to modularize it and to separate the HAL from the real-time kernel.

Architecture

Adeos implements a queue of signals. Each time that a peripheral sends a signal, the different operating systems that are running in the machine are awakened, in turn, and must decide if they will accept (handle), ignore, discard, or terminate the signal. Signals not handled (or discarded) by an OS are passed to the next OS in the chain. Signals that are terminated are not propagated to latter stages.

External links

- Adeos Home Page [2]
- Adeos Workspace [3]

References

[1] "Adaptive Domain Environment for Operating Systems" (http://whitepapers.zdnet.co.uk/0,1000000651,260088515p,00.htm). whitepapers.zdnet.co.uk. February 15, 2001. . Retrieved 2009-09-02.
[2] http://home.gna.org/adeos/
[3] https://gna.org/projects/adeos/

ALGOL 68C

The **ALGOL68C** computer programming language compiler was developed for the CHAOS OS for the CAP capability computer at Cambridge University in 1971 by Stephen Bourne and Mike Guy as a dialect of ALGOL 68. Other early contributors were Andrew D. Birrell[1] and Ian Walker.

The initial compiler was written in PSYCO (the Princeton SYntax COmpiler by Edgar T. Irons) and implemented by J.H. Mathewman at Cambridge. The language was called Z was subsequently morphed into ALGOL 68. ALGOL68C was built to develop the CAMbridge ALgebra system called CAMAL.

Subsequent work was done on the compiler after Bourne left Cambridge University in 1975. Garbage collection was added and the code base is still running on an emulated OS/MVT using Hercules.

The **ALGOL68C** compiler generated *ZCODE* output, that could then be either compiled into the local machine code by a *ZCODE* translator or run interpreted. *ZCODE* is a register-based intermediate language. This ability to interpret or compile *ZCODE* encouraged the porting of ALGOL 68C to numerous different computer platforms. Aside from the *CAP capability computer* the compiler was ported to systems including CMS, TOPS-10 and Z80.

Popular Culture

A very early predecessor of this compiler was used by Guy and Bourne to write the first life game programs on the PDP-7 with a DEC 340 [2] display (see Scientific American article) "For long-lived populations such as this one Conway sometimes uses a *PDP-7* computer with a screen on which he can observe the changes. The program was written by M. J. T. Guy and S. R. Bourne. Without its help some discoveries about the game would have been difficult to make." Scientific American 223 (October 1970): 120-123.

Various Liverpool Software Gazette issues detail the Z80 implementation. The compiler required about 120Kb of memory to run, hence the Z80's 64Kb memory is actually too small to run the compiler. So ALGOL 68C programs for the Z80 had to be cross compiled from ALGOL 68C running on the larger *CAP capability computer* or an IBM 370 mainframe.

Algol 68C and Unix

Stephen Bourne subsequently reused ALGOL 68's revered **if** ~ then ~ else ~ **fi**, **case** ~ in ~ out ~ **esac** and for ~ while ~ **do** ~ **od** clauses in the common Unix Bourne shell, but with in's syntax changed, out removed, and od replaced with done (to avoid conflict with the od utility).

After Cambridge, Bourne spent nine years at Bell Labs with the Seventh Edition Unix team. As well as developing the Bourne shell, he ported ALGOL 68C to Unix on the DEC PDP-11-45 and included a special option in his Unix debugger "adb" to obtain a stack backtrace for programs written in ALGOL68C. Here is an extract from the Unix 7th edition adb [3] manual pages:

```
NAME
      adb - debugger
SYNOPSIS
      adb [-w] [ objfil [ corfil ] ]
[...]
COMMANDS
[...]
       $modifier
             Miscellaneous  commands.   The  available modifiers
             are:
```

```
            [...]
            a      ALGOL 68 stack  backtrace.   If  address  is
                   given  then it is taken to be the address of
                   the current frame (instead of r4).  If count
                   is  given  then  only the first count frames
                   are printed.
```

ALGOL 68C extensions to Algol 68

Below is a sampling of some notable extensions:

- Automatic *op*:= for any operator, e.g. *:= and +:=
- UPTO, DOWNTO and UNTIL in loop-clauses;
- displacement operator (:=:=)
- ANDF, ORF and THEF *syntactic elements*.
- separate compilation - ENVIRON clause and USING clause
- scopes not checked
- bounds in formal-declarers
- CODE ... EDOC clause - for embedding ZCODE

The ENVIRON and USING clauses.

Separate compilation in ALGOL 68C is done using the ENVIRON and USING clauses. The ENVIRON saves the complete environment at the point it appears. A separate module written starting with a USING clause is effectively inserted into the first module at the point the ENVIRON clause appears.

ENVIRON and USING are useful for a *top-down* style of programming, in contrast to the *bottom-up* style implied by traditional library mechanisms.

These clauses are kind of the *inverse* of the **#include** found in the C programming language, or **import** found in Python. The purpose of the ENVIRON mechanism is to allow a program source to be broken into manageable sized pieces. Note that it is only necessary to parse the shared source file once, unlike a **#include** found in the C programming language where the include file needs to be parsed for each source file that includes it.

Example of ENVIRON clause

A file called *mylib.a68*:

```
BEGIN
   INT dim = 3; # a constant #
   INT a number := 120; # a variable #
   ENVIRON EXAMPLE1;
   MODE MATRIX = [dim, dim]REAL; # a type definition #
   MATRIX m1;
   a number := ENVIRON EXAMPLE2;
   print((a number))
END
```

Example of USING clause

A file called *usemylib.a68*:

```
USING EXAMPLE2 FROM "mylib"
BEGIN
  MATRIX m2; # example only #
  print((a number)); # declared in mylib.a68 #
  print((2 UPB m1)); # also declared in mylib.a68 #
  ENVIRON EXAMPLE3;  # ENVIRONs can be nested #
  666
END
```

Restrictions to the language from the standard ALGOL 68

- no algol68 FLEX and variable length arrays.
- MODE STRING implemented without FLEX.
- The PAR parallel clause was not implemented.
- nonstandard transput.
- others...

A translator/compiler for ALGOL 68C was available for the PDP-10 and System/360 as well as a number of other computers.

References

[1] Andrew D Birrell (December 1977). "System Programming in a High Level Language" (http://birrell.org/andrew/papers/thesis.pdf) (PDF). *Dissertation submitted for the degree of Doctor of Philosophy*. University of Cambridge. . Retrieved 04-22 2007.
[2] http://www.aceware.iinet.net.au/acms/ItemDetail.asp?lngItemId=175&
[3] http://modman.unixdev.net/?sektion=1&page=adb&manpath=v7man

Notes

- S.R. Bourne, A.D. Birrell and I. Walker, *Algol68C reference manual*, Cambridge University Computer Laboratory, 1975

External links

- Cambridge Algol 68: on the historical roster of computer languages (http://hopl.murdoch.edu.au/showlanguage.prx?exp=667) - includes 10+ publication references.
- A TRANSPORTATION OF ALGOL68C - PJ Gardner, University of Essex (http://portal.acm.org/ft_gateway.cfm?id=807148&type=pdf) - March 1977 (From 370 to DECsystem-10)

Amazon Machine Image

An **Amazon Machine Image (AMI)** is a special type of virtual appliance which is used to instantiate (create) a virtual machine within the Amazon Elastic Compute Cloud. It serves as the basic unit of deployment for services delivered using EC2.[1]

Contents

Like all virtual appliances, the main component of an AMI is a read-only filesystem image which includes an operating system (e.g., Linux, UNIX, or Windows) and any additional software required to deliver a service or a portion of it.[2]

The AMI filesystem is compressed, encrypted, signed, split into a series of 10MB chunks and uploaded into Amazon S3 for storage. An XML manifest file stores information about the AMI, including name, version, architecture, default kernel id, decryption key and digests for all of the filesystem chunks.

An AMI does not include a kernel image, only a pointer to the default kernel id, which can be chosen from an approved list of safe kernels maintained by Amazon and its partners (e.g., RedHat, Canonical, Microsoft). Users may choose kernels other than the default when booting an AMI.[3]

Types of images

- **Public**: an AMI image that can be used by anyone.
- **Paid**: a for-pay AMI image that is registered with Amazon DevPay and can be used by any one who subscribes for it. DevPay allows developers to mark-up Amazon's usage fees and optionally add monthly subscription fees.
- **Shared**: a private AMI that can only be used by Amazon EC2 users who are allowed access to it by the developer.

References

[1] Amazon. "Amazon EC2 Functionality" (http://aws.amazon.com/ec2/#functionality). .
[2] Amazon. "Creating an Image" (http://docs.amazonwebservices.com/AmazonEC2/gsg/2006-06-26/creating-an-image.html). .
[3] Feature Guide: Amazon EC2 User Selectable Kernels (http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1345)

External links

- Creating and preparing AMIs (http://docs.amazonwebservices.com/AWSEC2/2008-02-01/DeveloperGuide/CreatingAndBundlingAMIs.html)
- Amazon Web Services Developer Community : Amazon Machine Images (AMIs) (http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171)

Application virtualization

Application virtualization is an umbrella term that describes software technologies that improve portability, manageability and compatibility of applications by encapsulating them from the underlying operating system on which they are executed. A fully virtualized application is not installed in the traditional sense[1] , although it is still executed as if it were. The application is fooled at runtime into believing that it is directly interfacing with the original operating system and all the resources managed by it, when in reality it is not. In this context, the term "virtualization" refers to the artifact being encapsulated (application), which is quite different to its meaning in hardware virtualization, where it refers to the artifact being abstracted (physical hardware).

Description

Limited application virtualization is used in modern operating systems such a Microsoft Windows and Linux. For example, *INI file mappings* were introduced with Windows NT to virtualize, into the registry, the legacy INI files of applications originally written for Windows 3.1.[2] Similarly, Windows Vista implements a shim that applies limited file and registry virtualization so that legacy applications that try to save user data in a readonly system location that was writable by anyone in early Windows, can still work.[3]

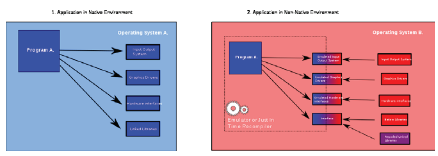


Illustration of an application running in a native environment and running in an application virtualization environment

Full application virtualization requires a virtualization layer.[4] Application virtualization layers replace part of the runtime environment normally provided by the operating system. The layer intercepts all file and Registry operations of virtualized applications and transparently redirects them to a virtualized location, often a single file.[5] The application never knows that it's accessing a virtual resource instead of a physical one. Since the application is now working with one file instead of many files and registry entries spread throughout the system, it becomes easy to run the application on a different computer and previously incompatible applications can be run side-by-side. Examples of this technology for the Windows platform are BoxedApp, Cameyo, Ceedo, Evalaze, InstallFree, Citrix XenApp, Novell ZENworks Application VIrtualization, Endeavors Technologies Application Jukebox, Microsoft Application Virtualization, Software Virtualization Solution, VMware ThinApp and InstallAware Virtualization.

Related Technologies

Technology categories that fall under application virtualization include:

- Application Streaming. Pieces of the application's code, data, and settings are delivered when they're first needed, instead of the entire application being delivered before startup. Running the packaged application may require the installation of a lightweight client application. Packages are usually delivered over a protocol such as HTTP, CIFS or RTSP.[6]
- Desktop Virtualization/Virtual Desktop Infrastructure (VDI). The application is hosted in a VM or blade PC that also includes the operating system (OS). These solutions include a management infrastructure for automating the creation of virtual desktops, and providing for access control to target virtual desktop. VDI solutions can usually fill the gaps where application streaming falls short.