Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 11 November 2024

Case Number:	T 1428/22 - 3.5.06
Application Number:	12832494.4
Publication Number:	2756394
IPC:	G06F9/44, G06F9/455, G06F9/50, G06F9/54
Language of the proceedings:	EN

Title of invention:
APPLICATION ACCELERATION IN A VIRTUALIZED ENVIRONMENT

Applicant:
Microsoft Technology Licensing, LLC

Headword:
Hypervisor/Microsoft

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - (no)

Decisions cited:

This datasheet is not part of the Decisior
It can be changed at any time and without notic

Catchword:

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0

Case Number: T 1428/22 - 3.5.06

DECISION
of Technical Board of Appeal 3.5.06
of 11 November 2024

Appellant: (Applicant)	Microsoft Technology Licensing, LLC One Microsoft Way Redmond, WA 98052-6399 (US)
Representative:	Grünecker Patent- und Rechtsanwälte PartG mbB Leopoldstraße 4 80802 München (DE)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 5 January 2022 refusing European patent application No. 12832494.4 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	M. Müller
Members:	T. Alecu
	B. Müller

Summary of Facts and Submissions

I. The appeal is against the decision of the Examining Division. The Appellant requests that the decision of the Examining Division be set aside and that a patent be granted on the basis of the main request or one of two auxiliary requests which were subject to the decision under appeal. They were refused for lack of inventive step (Article 56 EPC) in view of D1: US 2005/246453 A1. A third auxiliary request, which was filed with the statement of grounds of appeal, was withdrawn in the oral proceedings before the Board.

II. Claim 1 of the main request defines:

In a computing environment, a method of providing direct access to hardware to virtual machines (104), the method comprising:

determining (402) that a virtual machine (104-1) of the virtual machines should have access to a piece of hardware (102);

the virtual machine requesting (404) access to the hardware from a host, wherein the host is a partition that controls the physical hardware of a computing system and manages virtual machines;

the host configuring (406) the hardware to allow access to the hardware directly by the virtual machine by a component (110) at the host allocating an overlay for the virtual machine that maps hardware resources into the virtual machine's address space, wherein the configuring further comprises the component (110) at the host creating an overlay map that maps the hardware resources to the virtual machine's address space, and wherein the configuring (406) the hardware to allow access is based on the host applying a policy that *includes access checks and wherein the policy enforces partitions; and*

the virtual machine directly accessing (408) the hardware without going through the host once the hardware has been configured by the host.

III. Claim 1 of the first auxiliary request differs from that of the main request by adding that

the configuring further comprises the component (110) at the host creating an overlay map that maps the hardware resources to the virtual machine's address space.

IV. Claim 1 of the second auxiliary request differs from that of the first auxiliary request by modifying the last feature to define that

the virtual machine directly accessing (408) the hardware <u>using the map for the virtual machine</u> without going through the host once the hardware has been configured by the host.

Reasons for the Decision

The application

1. The application relates to virtualized environments, and in particular to means for providing direct access to hardware to software components in a virtual machine. It proposes to virtualise "*drivers and APIs allowing virtual machines to directly interact with hardware devices, rather than virtualizing hardware devices*" (paragraph 14).

1.1 For instance, RDMA (remote direct memory access) devices enable applications to directly access hardware devices. By virtualising the driver components working in the application address space, applications running in virtual machines can work as if they were interfacing with the hardware directly (paragraph 18).

1.2 In the example of figure 1, a host operating system and virtual operating systems are shown on top of a hypervisor which accesses memory and processors. The operating systems can access the RDMA devices directly.

1.3 The driver virtualisation is managed by a virtual service provider (VSP), described as a component of the host. It uses the hardware driver (IHV Driver) to map physical resources to the host memory space, and, on the request of a virtual machine (a virtual service client) it provides an overlay which maps the application address in the virtual machine to the host address space corresponding to the physical resource (paragraphs 21 to 25).

The prior art

2. Document D1 also relates to providing direct access to hardware. In D1 a hypervisor and a host operating system both have direct access to the hardware by a timesharing mechanism (paragraph 103). The hypervisor provides direct access to hardware to the virtual machines by modifying the page table mappings, determining which physical memory locations are assigned to given processes. The hypervisor then allows pass-through operation for components with direct memory access (paragraphs 106, 107).

Main request

3. The Examining Division equated the claimed "*host*" with the hypervisor of D1, on the basis that the functions of the host, as defined in the claim, are the same as that of a hypervisor, and stated that the hypervisor of D1 provides a mapping "*that maps the hardware resources to the virtual machine’s address space*" (decision, reasons 12.2 and 13.2). It considered that the difference to D1 was essentially the allocation of an overlay, understood by the Examining Division to correspond to a memory allocation operation, which was an obvious measure in order to avoid conflicts (decision, reasons 12.3 to 12.7).

4. The Appellant did not contest that the allocation of an overlay to provide direct memory access was obvious.

4.1 The Appellant's central inventive step argument (see statement of grounds of appeal, pages 3 to 5) was instead that the claimed "*host*" could not be equated with a hypervisor. While historically a hypervisor could have been termed a host, that interpretation was excluded by the claim language, defining the host as a partition, whereas a hypervisor was "*a collection of code*" - as defined in D1 paragraph 105.

4.2 According to the Appellant (top of page 5): "*Thus, the claimed invention differs from known concepts in that no hypervisor is needed anymore for those functions that eventually provide the direct access for the virtual machines to the hardware. Because the claimed host partition is providing the direct access, the conventionally used hypervisor (as in D1) is relieved from having to performs [sic] the mapping or the modifications of the page table mappings in the common* *operating system modified (both as in D1, see [0102]-[0108] in D1 as discussed and cited in the decision)*".

4.3 This new concept increased the security of the system, by reducing the complexity of the hypervisor code, so that fewer attacks were possible due to smaller amount of code that could be attacked (a smaller "surface area") and a smaller risk for bugs.

5. In reply to the Board's preliminary opinion, which was that it tended to agree with the analysis of the Examining Division, and during the oral proceedings before the Board, the Appellant provided the following further arguments.

5.1 The invention moved functionality that was previously implemented by a hypervisor to a host partition. It was incorrect to equate a host partition with a hypervisor, which was not a partition, but code: "*[a] hypervisor could run on a host machine but also outside the host*" (reply of 30 October 2024, second page).

5.2 The term host had a specific meaning to the person skilled in the art. Claim 1 of all requests "*define[d] the host partition as 'controll[ing] the physical hardware of the computing system'*". A hypervisor did not control the hardware. A hypervisor allowed the virtual machines to access the hardware and use it. But it did not control the hardware in the sense that it could for instance not remove a piece of hardware. Hardware control was always done by the host operating system through its kernel. The application taught that the moved functionality was provided in the kernel mode of the host operating system (virtual service provider

(VSP), see figure 1 and paragraph 15) which controlled the hardware.

5.3 To carry out these functions, the host operating system did not have to communicate with the hypervisor, which could be simplified accordingly.

6. The Board notes that the claim defines the concept of "host" in the following way:

(a) it is a partition
(b) it controls the physical hardware of a computing system
(c) it manages virtual machines
(d) it provides direct memory access to virtual machines.

6.2 It is undisputed that the hypervisor of D1 provides functions (c) and (d).

6.3 Regarding (a), the Board remarks first that a "partition" normally defines a set of physical resources. As such, a partition does not control hardware or manage virtual machines. To achieve this, code installed *on a partition* is required. Property (a), in in view of properties (b) to (d), must therefore be read as stating that the host is *code installed on a partition*. The hypervisor of D1 fulfils this condition.

6.4 During the oral proceedings (see above), the Appellant emphasised property (b), in combination with the usage of the term "*host*", to argue that the skilled person would not understand a hypervisor, such as that disclosed in D1, to be a host controlling the physical hardware.

6.4.1 The Board disagrees. A hypervisor, in particular a type-I hypervisor, installed on bare metal, does control hardware, i.e. the machine which is "host" insofar as it "hosts" virtual machines. There is simply no other host system to exercise control.

6.4.2 The Board notes that the Appellant seems to conflate the terms host machine and host operating system. There is a distinction to be made. The host machine is the hardware, and any operating system installed on a host machine allowing access to hardware and providing services is a host operating system for those services.

6.4.3 This is exactly the case of the hypervisor of D1. In the words of D1, paragraph 101, the hypervisor "*can be computer executable instructions that <u>manage a virtual machine environment</u> by <u>providing limited operating system functionality</u> and by <u>providing abstracted access to underlying hardware</u>*" (emphasis by the Board). Since the hypervisor *abstracts* the hardware, and provides limited operating system functionality, the hypervisor effectively controls the hardware in the sense argued by the Appellant.

6.5 The Board is therefore of the opinion that the claim wording does not allow any distinction to be made between the claimed host and the hypervisor of D1.

7. The Appellant also argued that the claim should be read in the light of the description which made clear that the host was different from a hypervisor.

8. This argument is also not convincing. Upon reading the description the person skilled in the art may understand that in one embodiment (e.g. figure 1) the host (112) may be different from a "conventional"

hypervisor (114). However, in other embodiments, such as the method embodiment of figure 4 (cf. also paragraphs 31 to 37), upon which claim 1 is actually based, no reference is made to a hypervisor. It is noted that the description mentions the term "hypervisor" only once, namely in the description of figure 1 (see paragraph 15) and that the description of figure 4 makes no reference to the embodiment according to the system of figure 1 either (but only to the method being "*practiced in a computing environment*").

9. The Board concludes that claim 1 lacks inventive step in view of D1, Article 56 EPC.

First and second auxiliary requests

10. Claim 1 of the auxiliary requests differs from claim 1 of the main request merely by clarification of the mapping function of the overlay and the requirement that the direct access functionality is actually used by the virtual machines. These features are obvious measures for providing direct hardware access. The Appellant did not contest this. Therefore, claim 1 of the two auxiliary requests also lacks an inventive step, Article 56 EPC.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

L. Stridde Martin Müller

Decision electronically authenticated