

**УДК 339.54.012**

**Сергеев Анатолий Васильевич**, доцент, кандидат технических наук,  
Петербургский Политехнический Университет имени Петра Великого, г.  
Санкт-Петербург

**Касьяненко Григорий Алексеевич**, магистрант, Петербургский  
Политехнический Университет имени Петра Великого, г. Санкт-Петербург

**СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ  
ГИПЕРВИЗОРОВ ПРИ ИСПОЛЬЗОВАНИИ VMWARE  
WORKSTATION И ORACLE VIRTUALBOX В ОС LINUX**

**Аннотация.** В статье предложена методика комплексной оценки производительности виртуальных машин на гипервизорах второго типа и выполнено экспериментальное сравнение VMware Workstation и Oracle VirtualBox в среде Linux. Методика основана на автоматизированном запуске синтетических нагрузочных тестов, сборе временных рядов показателей производительности и их последующей статистической обработке. Оценка включает анализ вычислительных, память-зависимых, дисковых и графических нагрузок, а также исследование масштабируемости при увеличении числа одновременно работающих виртуальных машин. Для формирования обобщенных показателей эффективности применены методы нормализации данных и многокритериальной оптимизации на основе иерархической модели Саати.

**Annotation**

This paper proposes a comprehensive methodology for evaluating the performance of virtual machines on type-2 hypervisors and presents an experimental comparison of VMware Workstation and Oracle VirtualBox in a Linux environment. The methodology is based on automated execution of synthetic workload benchmarks, collection of performance time series, and subsequent statistical analysis of the

results. The evaluation includes analysis of computational, memory-intensive, disk, and graphical workloads, as well as an investigation of scalability with an increasing number of concurrently running virtual machines. To obtain aggregated performance indicators, data normalization techniques and multi-criteria optimization based on the Analytic Hierarchy Process (AHP) are applied.

**Ключевые слова:** виртуализация; сравнение гипервизоров; VMware Workstation; Oracle VirtualBox; нагрузочное тестирование; метод анализа иерархий; Linux

**Keywords:** virtualization; hypervisors comparison; VMware Workstation; Oracle VirtualBox; load testing; analytic hierarchy process; Linux

## **Введение**

Развитие технологий виртуализации стало одним из ключевых факторов повышения эффективности использования вычислительных ресурсов и гибкости современных информационных систем. Виртуальные машины (VM) широко применяются в задачах разработки, тестирования, обучения, а также при развертывании корпоративных и научных вычислительных инфраструктур. Производительность виртуализированных систем в значительной степени определяется архитектурой гипервизора, механизмами виртуализации аппаратных ресурсов и особенностями взаимодействия с операционной системой хоста.

В научной литературе представлен широкий круг исследований, посвященных сравнительному анализу гипервизоров второго типа. В работе [1] выполнено экспериментальное сравнение VMware Workstation, Oracle VirtualBox и Microsoft Hyper-V при одинаковых конфигурациях гостевых систем, где показано, что даже гипервизоры одного класса могут демонстрировать заметные различия по вычислительной производительности и устойчивости результатов. Аналогичные выводы получены в исследованиях

[2,3], в которых подчеркивается влияние реализации виртуального планировщика и подсистемы ввода-вывода на итоговые показатели производительности.

Отдельное направление исследований связано с анализом масштабируемости виртуальных машин при увеличении числа одновременно работающих экземпляров. В ряде работ отмечается, что деградация производительности носит, как правило, сублинейный характер и зависит от типа нагрузки и степени конкуренции за ресурсы хоста. В частности, работы [4,5] показывают, что операции с оперативной памятью и вычислительные тесты обладают различной чувствительностью к конкуренции за кэш-память и пропускную способность шины памяти, что приводит к неоднородному поведению различных гипервизоров. Эти результаты подтверждаются и в сравнительных исследованиях виртуализации и контейнеризации [6,7].

Значительное внимание в современных публикациях уделяется также дисковой подсистеме и операциям ввода-вывода. В работах [8,9] показано, что при типовых сценариях ввода-вывода различия между гипервизорами могут сглаживаться за счет механизмов кэширования, однако при интенсивных I/O-нагрузках архитектурные особенности гипервизоров проявляются более отчетливо. Аналогичные выводы приводятся и в прикладных исследованиях последних лет.

Отдельную группу составляют работы, посвященные разработке методик тестирования и анализу устойчивости результатов. В публикации [10] предлагаются подходы к автоматизации нагрузочного тестирования и статистической обработке экспериментальных данных, подчеркивается необходимость учета не только средних значений, но и вариативности результатов. Вместе с тем большинство таких работ ориентировано либо на контейнерные технологии, либо на ограниченный набор метрик, что оставляет открытым вопрос комплексной многокритериальной оценки гипервизоров второго типа в среде Linux.

Таким образом, несмотря на наличие значительного числа исследований, остается актуальной задача разработки комплексной методики тестирования виртуальных машин, ориентированной на практическое сравнение гипервизоров второго типа с учетом вычислительных, память-зависимых, дисковых и графических нагрузок. Целью данной работы является разработка такой методики и экспериментальное сравнение гипервизоров VMware Workstation и Oracle VirtualBox на одинаковых конфигурациях хостовой и гостевых систем под управлением Linux с применением статистического анализа и методов многокритериальной оптимизации.

## Подготовка виртуальных машин

На данном этапе задокументируем ресурсы хоста, создадим виртуальные машины и установим гостевые операционные системы вместе с пакетами, необходимыми для запуска бенчмарков.

### 1. Конфигурация хоста

Результаты тестирования ВМ напрямую зависят от аппаратного обеспечения и операционной системы хоста. Подробные характеристики хоста приведены в табл. 1.

*Таблица 1 – Подробные характеристики хоста*

Имя устройства	Aspire-A315-42G (V1.06)
Тип устройства	Ноутбук
CPU	AMD Ryzen 5 3500U (4 ядра/8 потоков; 1400-2100 МГц)
RAM	16ГБ (2x8; DDR4-2400)
Disk	AMD R5SL1024G (SSD SATA; 938ГБ; ext4)
GPU	AMD Radeon Vega 8 Graphics (iGPU, DRM 3.57)
Host OS	CachyOS (Linux 6.18.2-2-cachyos)
VMware Workstation	VMware Workstation 25.0.0 24995812
Oracle VirtualBox	7.2.4r170995

Virtualization	AMD-V
----------------	-------

## 2. Создание VM

Для объективного сравнения важно соблюдать строгую симметрию настроек VM в двух гипервизорах. Следует выставить минимальные настройки, чтобы в будущем протестировать одновременную нагрузку нескольких VM (до 3 штук). Выбранные настройки указаны в табл. 2.

*Таблица 2 – Настройки виртуальных машин*

vCPU	2 ядра, 2 потока
vRAM	2ГБ
Диск	20ГБ (SATA, фиксированный)
Видеопамять	128МБ; 3D отключено
Сеть	Сетевой мост
Звук	Отключен
Guest OS	Lubuntu-24.04.3-desktop-amd64

При создании VM в VMware Workstation и Oracle VirtualBox были замечены некоторые дополнительные настройки. Далее разберем их.

VMware Workstation:

- создание диска: storage disk as a **single** file (точнее измеряется I/O производительность, меньше накладных расходов на файловую систему хоста);
- virtualize CPU performance counters = **False** (при включении PMU гипервизор должен перехватывать дополнительные инструкции MSR/CPUID, что добавляет накладные расходы) [11].

VirtualBox:

- материнская плата: включить I/O APIC = **True** (включается современная схема прерываний APIC: гостевая ОС видит все vCPU, может работать SMP-планировщик, корректно работают таймеры и

sleep/interrupt-события);

— процессор: включить PAE/NX = **True** (без этого часть тестов может выдать предупреждения или использовать неэффективные адресные таблицы);

— ускорение: включить Nested Paging = **True** (для минимизации накладных расходов на трансляцию адресов памяти) [12].

Эти настройки позволяют уравнивать возможности ВМ на разных гипервизорах для честного сравнения.

В качестве дистрибутива был выбран `lubuntu-24.04.3-desktop-amd64`. Легковесный дистрибутив хорошо подходит для тестирования производительности. При этом для тестирования можно использовать встроенные инструменты.

Операционную систему установим стандартным образом с автоматической разметкой разделов. Все настройки оставлены по умолчанию, от предустановленных приложений отказываемся. Выбранные настройки приведены в табл. 3.

*Таблица 3 – Параметры установки гостевой ОС Lubuntu*

installation mode	minimal
таблица разделов	MBR
файловая система	ext4

Для снижения фоновой активности и исключения влияния лишних системных процессов были отключены неиспользуемые службы гостевой ОС, связанных с Bluetooth и принтерами. Это позволило уменьшить количество фоновых прерываний и контекстных переключений, обеспечив более стабильные результаты в тестах CPU и памяти.

```
sudo systemctl disable bluetooth.service cups.service avahi-daemon.service
```

Подготовим среду для тестирования. Установим недостающие пакеты во всех ВМ. Тесты производительности (далее, бенчмарки) автоматизируют

измерения метрик производительности. Таким образом, нам не обязательно писать собственную программу, нагружающую гостевую ОС. Для тестирования всех ресурсов нам понадобится:

- `sysbench` – позволяет тестировать процессор, оперативную память и файловую систему)
- `x11-apps` – позволяет тестировать 2D графику в графической оболочке Linux под названием X11.

```
sudo apt install -y sysbench x11-apps openssh-server
```

Для обеспечения корректной работы гостевой системы и воспроизводимости тестов в VMware Workstation были установлены пакеты `open-vm-tools` и `open-vm-tools-desktop`. Это позволило активировать паравиртуализированные драйверы ввода-вывода, синхронизацию времени и оптимизированный видеодрайвер `vmwgfx` для справедливого тестирования графической оболочки.

```
sudo apt install -y open-vm-tools open-vm-tools-desktop
```

Для гипервизора Oracle VirtualBox устанавливаем гостевые дополнения, аналогичные `open-vm-tools`.

```
sudo apt install -y virtualbox-guest-additions-iso virtualbox-guest-x11-hwe
```

Так как ВМ в разных гипервизорах создавались примерно в одно и тоже время, не было нужды синхронизировать версии пакетов. Однако при необходимости повторения эксперимента в будущем, следует соблюдать те же версии пакетов, чтобы избежать разночтений в вызове команд бенчмарков. Список пакетов с их версиями приведены в приложении и зафиксированы командой:

```
sudo dpkg-query -W $(apt mark showmanual)
```

Эта команда выведет список лишь родительских пакетов, которые мы сможем установить, чтобы реплицировать исходную конфигурацию ОС.

## Разработка программы для запуска бенчмарков

Будем использовать встроенные команды оболочки Bash и ранее установленные на гостевые ОС пакеты sysbench и x11-apps.

Sysbench — это многоцелевая утилита командной строки для тестирования производительности в Linux, которая позволяет оценивать производительность процессора, памяти, ввода-вывода и баз данных, например MySQL. Она представляет собой простой инструмент для оценки работы аппаратного обеспечения и может использоваться для тестирования системы под нагрузкой, в том числе для определения стабильности [13].

X11perf (из пакета x11-apps) – утилита для тестирования производительности графической подсистемы X11 (сервер, драйверы, протокол). Она измеряет, насколько быстро выполняются графические операции, такие как рисование линий, копирование областей и другие, которые обычно обрабатываются сервером X11 [14].

### 1. Скрипты для нагрузки VM по одному ресурсу

Каждый отдельный скрипт: “cpu.sh”, “ram.sh”, “disk.sh”, “graphics1.sh”, “graphics2.sh” – нагружает соответствующий ресурс VM, парсит интересующие метрики и записывает их в csv-файл. Рассмотрим, какими методами проводится нагрузка:

- “sysbench cpu --cpu-max-prime=20000”: подсчет простых чисел процессором до 20000 (берем скорость выполнения);
- ”sysbench memory --memory-total-size=2G --memory-oper=**read/write**”: прогон 2ГБ операций через оперативную память (берем пропускную способность МБ/с при чтении и при записи);
- “sysbench fileio --file-total-size=4G --file-num=16 prepare”: создание 16 файлов размером 4ГБ суммарно, размер файлов следует ставить больше оперативной памяти для корректного определения ввода;
- “sysbench fileio --file-total-size=4G --file-num=16 --file-test-mode=**seqrd/seqwr** --file-extra-flags=direct --time=60 --max-requests=0”:

последовательные чтения/записи с обходом файлового кэша (direct) длительностью 60 сек (берем пропускную способность МБ/с, число операций в секунду и среднюю задержку в мс);

— “sysbench fileio cleanup”: очистка файлов после тестирования [4];

— “x11perf -repeat 1 "\$t" 2>/dev/null”: выполнение по 1 тесту 2D-графики (фиксируем кол-во операций в секунду):

— scrollX: вертикальная прокрутка X:X пикселей;

— rectX: заполнение сплошного прямоугольника определенного размера;

— polytext/polytext16: символ в строке 20/40/20 (6x13, Times-Roman 10, 6x13) или 2-байтовый символ в строке 7/14/7 (k14, k24);

— move: движение прямоугольного окна по плоскости (фиксируем 50/100/200 ячеек) [14].

Также, чтобы протестировать эффективность многопоточности виртуальных машин, команды категории sysbench будем запускать на всех ядрах.

## 2. Скрипт для параллельной нагрузки нескольких VM

Для синхронной нагрузки одного ресурса идеально подойдет пакет *parallel-ssh* на хосте, откуда будем запускать скрипты для нагрузки VM. Нужно подготовить список IP-адресов для подключения через SSH и затем для параллельного выполнения команд (каждая строчка в виде *grigoriy@<IP address>*).

Пример команды, выполняемой параллельно:

```
pssh -h "$hosts" -t $TIMEOUT -i "bash ~/share/bench_prepare.sh $RAW",
```

где:

— hosts – текстовый файл IP-адресов всех VM (1-3 штук);

— TIMEOUT – по истечению указанного времени команда прервет свое выполнение (ставим 30 минут во избежание таких ситуаций).

В обычном случае команда будет ждать свое выполнение на всех VM

[15].

Для нагрузки 1 ВМ следует использовать тот же алгоритм для обеспечения справедливого сравнения производительности.

### 3. Скрипт для мониторинга ресурсов хоста

Качество гипервизора оценивается не только производительностью виртуальных машин, но и эффективностью потребления ресурсов хоста. Для измерения был разработан собственный скрипт, который каждую секунду записывает в файл логирования загрузку одного из ресурсов в процентах.

Скрипт мониторинга ресурсов запускался через интерфейс командной строки со следующим форматом вызова:

```
sysmonitoring.sh [category] [logfile] [freq],
```

где:

- category – тип нагружаемого ресурса (cpu/ram/disk);
- logfile – местоположение файла логирования;
- freq – частота логирования (по умолчанию – 1 секунда).

Скрипт запускался параллельно вместе с соответствующими бенчмарками. Завершение работы скрипта осуществлялось вручную посредством отправки сигнала завершения процессу.

## **Анализ полученных результатов**

В данном разделе приводится сравнение гипервизоров, подкрепленное сводными сравнительными таблицами и графическими иллюстрациями.

### **1. Сравнение производительности гипервизоров**

Ниже представлена общая сравнительная 3D-таблица («название гипервизора», «количество VM», «название метрики»). В таблице фигурирует среднее выборочное и погрешность измерения, рассчитанная по доверительному интервалу распределения Стьюдента с надежностью  $P=0,90$ . Каждый тест записан по 5 раз.

Жирным выделены лучшие значения по сравнению с конкурирующим гипервизором (табл. 4).

Таблица 4 – Сводная сравнительная таблица VMware и VirtualBox  
(P=0,90)

vmware vbox	ID теста	Название теста	1 VM	2 VM*	3 VM*
CPU	cpu	подсчет простых чисел, кол-во событий/с	<b>1 173±126</b> 1 051±107	<b>1014±120</b> 963±130	<b>567±39</b> 522±87
		чтение, МБ/с	1 272±119 <b>9 018±1 492</b>	1 087±77 <b>7 857±1 861</b>	848±103 <b>5 512±1 844</b>
RAM	mem r	чтение, МБ/с	1 272±119 <b>9 018±1 492</b>	1 087±77 <b>7 857±1 861</b>	848±103 <b>5 512±1 844</b>
	mem w	запись, МБ/с	1 393±150 <b>7 023±1 013</b>	1 097±136 <b>5 988±1 118</b>	897±152 <b>3 976±1 403</b>
DISK IO	sqrd m	последовательное чтение, МБ/с	<b>195±11</b> 122±0.669	<b>145±18</b> 73±6	<b>95±16</b> 54±7
	sqrd r	последовательное чтение, кол-во запросов/с	<b>12 479±726</b> 7 827±43	<b>9 288±1 160</b> 4 673±368	<b>6 076±1 055</b> 3 471±466
	sqrd d	последовательное чтение, задержка IO в мс	<b>0.158±0.010</b> 0.252±0.004	<b>0.216±0.023</b> 0.430±0.036	<b>0.332±0.050</b> 0.580±0.065
	sqwr m	последовательная запись, МБ/с	<b>93±8</b> 41±0.43	<b>59±5</b> 31±3	<b>40±9</b> 25±2
	sqwr r	последовательная запись, кол-во запросов/с	<b>5 961±533</b> 2 615±28	<b>3 779±329</b> 1985±179	<b>2 546±562</b> 1 568±125
	sqwr d	последовательная запись, задержка IO в мс	<b>0.288±0.025</b> 0.656±0.009	<b>0.454±0.037</b> 0.872±0.077	<b>0.698±0.143</b> 1.096±0.079
	X11-	sc10	вертикальная	<b>998 800±66 928</b>	<b>438 000±95 722</b>

2D		прокрутка 10x10, кол-во повторений/с	396 600±22 527	147 440±46 482	108 140±38 932
	sc100	вертикальная прокрутка 100x100, кол-во повторений/с	<b>198 400±21 272</b> 89 020±3 859	<b>130 200±45 054</b> 9 860±2 444	<b>84 060±22 170</b> 7 624±1 828
	sc500	вертикальная прокрутка 500x500, кол-во повторений/с	<b>18 580±3 096</b> 6 302±997	<b>8 610±1 161</b> 568±221	<b>4 478±1 479</b> 939±208
	rec10	прямоугольник размером 10, кол-во повторений/с	<b>4 748 000±504</b> <b>142</b> 358 600±48 420	<b>2 156 000±657</b> <b>258</b> 156 600±47 744	<b>710 800±193</b> <b>854</b> 130 600±30 123
	rec100	прямоугольник размером 100, кол-во повторений/с	<b>504 425±192</b> <b>730</b> 154 200±35 266	<b>284 000±58 991</b> 12 960±3 942	<b>149 400±31 292</b> 10 996±1 016
	rec500	прямоугольник размером 500, кол-во повторений/с	<b>23 020±2 129</b> 11 004±4 121	<b>12 790±4 967</b> 529±112	<b>8 302±3 997</b> 464±112
	txt8	отображение символов в строке 1 байт, кол-во повторений/с	<b>3 124 000±387</b> <b>128</b> 2 870 000±629 984	<b>2 146 000±583</b> <b>776</b> 465 400±64 276	<b>1 384 000±166</b> <b>993</b> 269 800±28 469
	txt16	отображение символов в строке 2 байта, кол-во повторений/с	<b>1 019 000±139</b> <b>819</b> 360 600±47 801	<b>718 800±30 576</b> 107 400±49 642	<b>413 200±64 744</b> 80 540±25 446
	mv50	движение окна 50 ячеек, кол-во повторений/с	<b>70 220±7 625</b> 17 700±1 846	<b>58 580±9 333</b> 9 992±3 178	<b>48 560±14 617</b> 4 392±1 398
	mv100	движение окна 100	<b>59 100±6 737</b>	<b>54 940±6 463</b>	<b>45 760±3 686</b>

00	ячеек, кол-во повторений/с	13 140±298	8 046±2 924	4 363±1 983
mv2	движение окна 200	<b>31 900±2 650</b>	<b>29 325±3 027</b>	<b>25 400±1 539</b>
00	ячеек, кол-во повторений/с	12 460±507	7 680±2 056	4 354±2 010

(\*) в расчет бралось наихудшее значение по нескольким ВМ в пределах одного теста.

Исходя из таблицы, VMware Workstation выигрывает конкурента по производительности процессора, диска и 2D-графики. Однако VirtualBox оказался эффективнее по оперативной памяти.

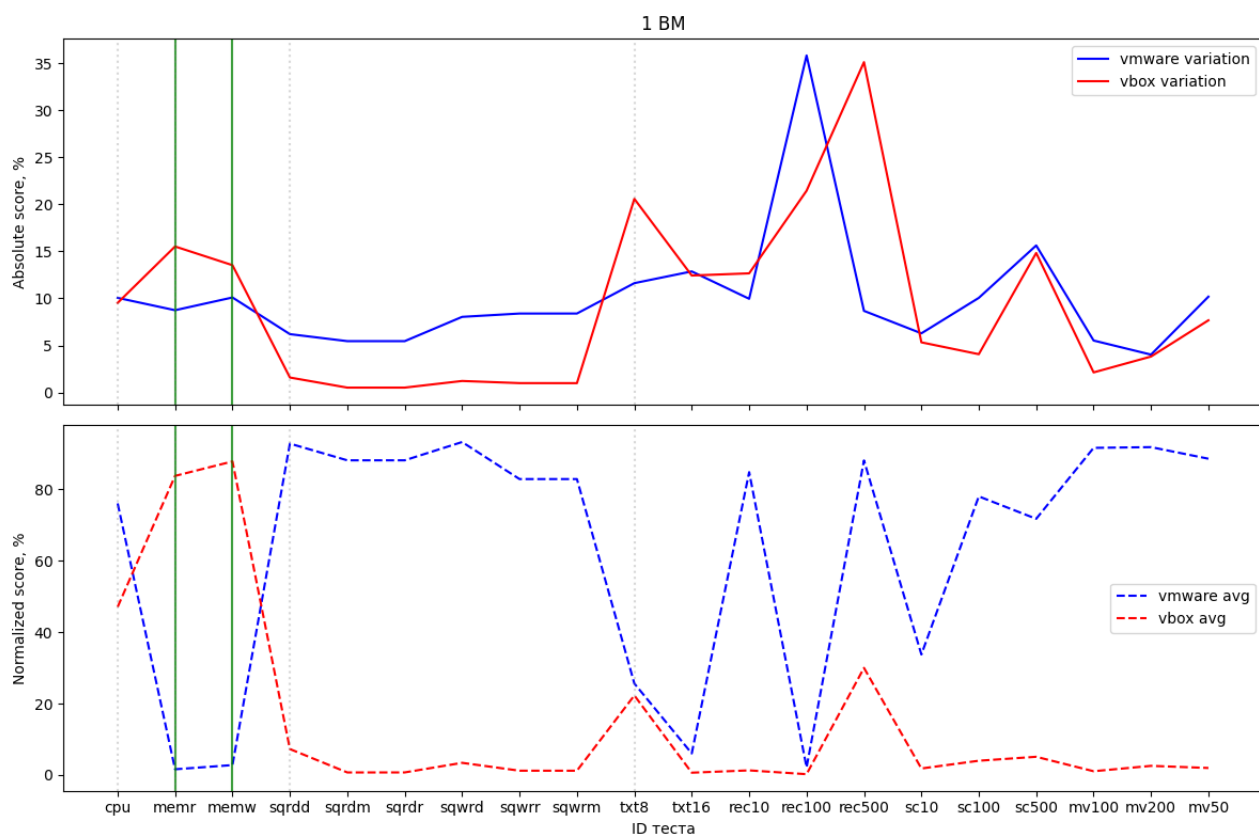
Далее рассмотрим средние значения из таблицы дополнительно вместе с коэффициентами вариации. Коэффициент вариации показывает, насколько стабильны характеристики, и рассчитывается как отношение среднеквадратического отклонения ( $\sigma_x$ ) к среднему выборочному ( $\bar{x}$ ):

$$v = \frac{\sigma_x}{\bar{x}}$$

Чем меньше коэффициент, тем стабильнее тесты.

Обычно коэффициент вариации лежит в пределах [0; 1], его можно представить в процентах без нормализации. Что касается средних значений, разные тесты допускают совершенно не пересекаемый друг с другом диапазон значений, поэтому их необходимо нормировать. Сделаем нормировку по шкале от минимума до максимума конкретного теста.

Таким образом, на рис. 3-1 представлен график абсолютных значений коэффициента вариации по каждой метрике, а ниже него расположен нормированный график средних значений. Все значения представлены в процентах. Метрики подписаны по короткому кодовому названию (см. ID теста в табл. 4).



*Рисунок 0-1 – Коэффициент вариации и среднее нормированное 1 VM*

Зеленой линией отмечены точки, в которых VirtualBox лучше VMware по среднему значению. Динамика данных сходится со значениями в таблице: лишь в тестах оперативной памяти VirtualBox лучше конкурента. Судя по коэффициенту вариации, в тестах на процессор и оперативную память VMware показывает более стабильные результаты. Что касается дисковой подсистемы, хоть VMware имеет большой выигрыш по среднему значению, лидером по стабильности остается VirtualBox.

На рис. 3-2 и рис. 3-3 представлены такие же графики для 2 VM и 3 VM соответственно.

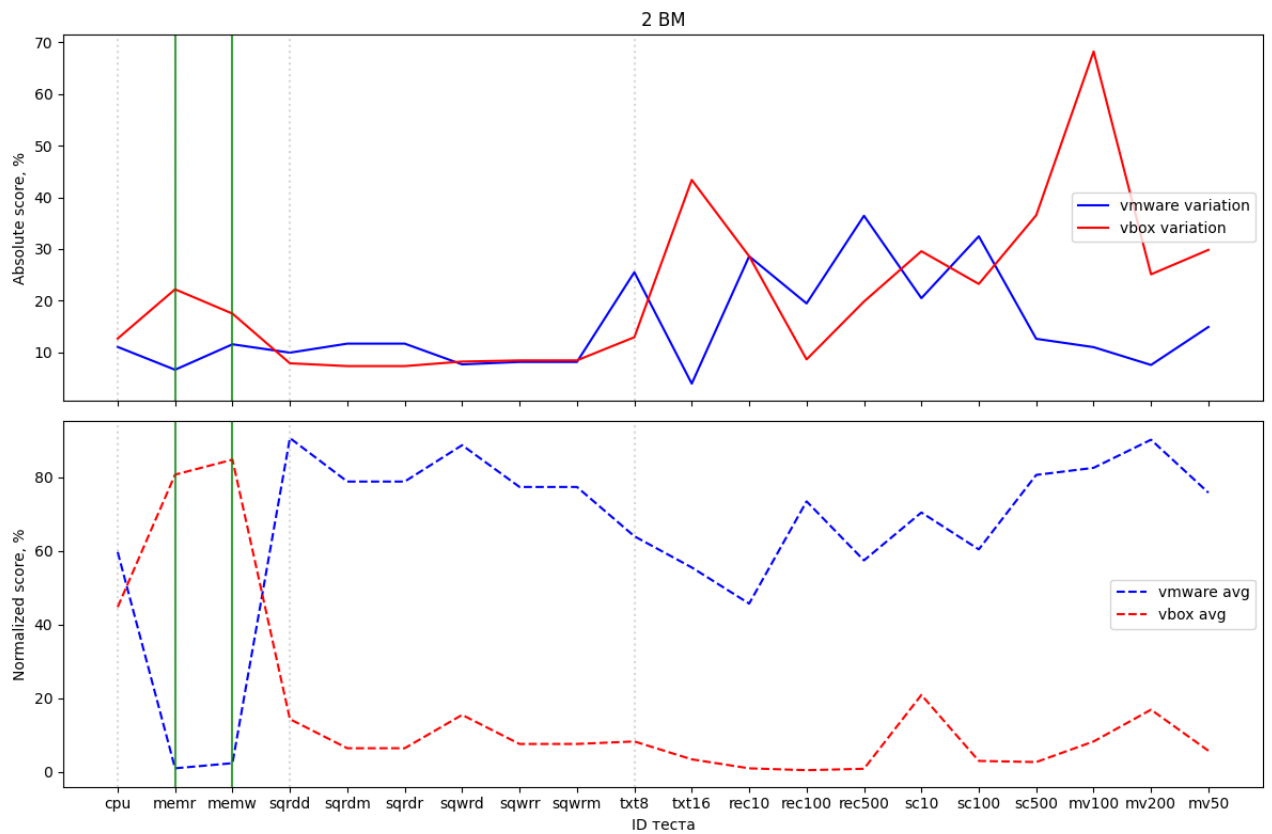


Рисунок 0-2 – Коэффициент вариации и среднее нормированное 2 VM

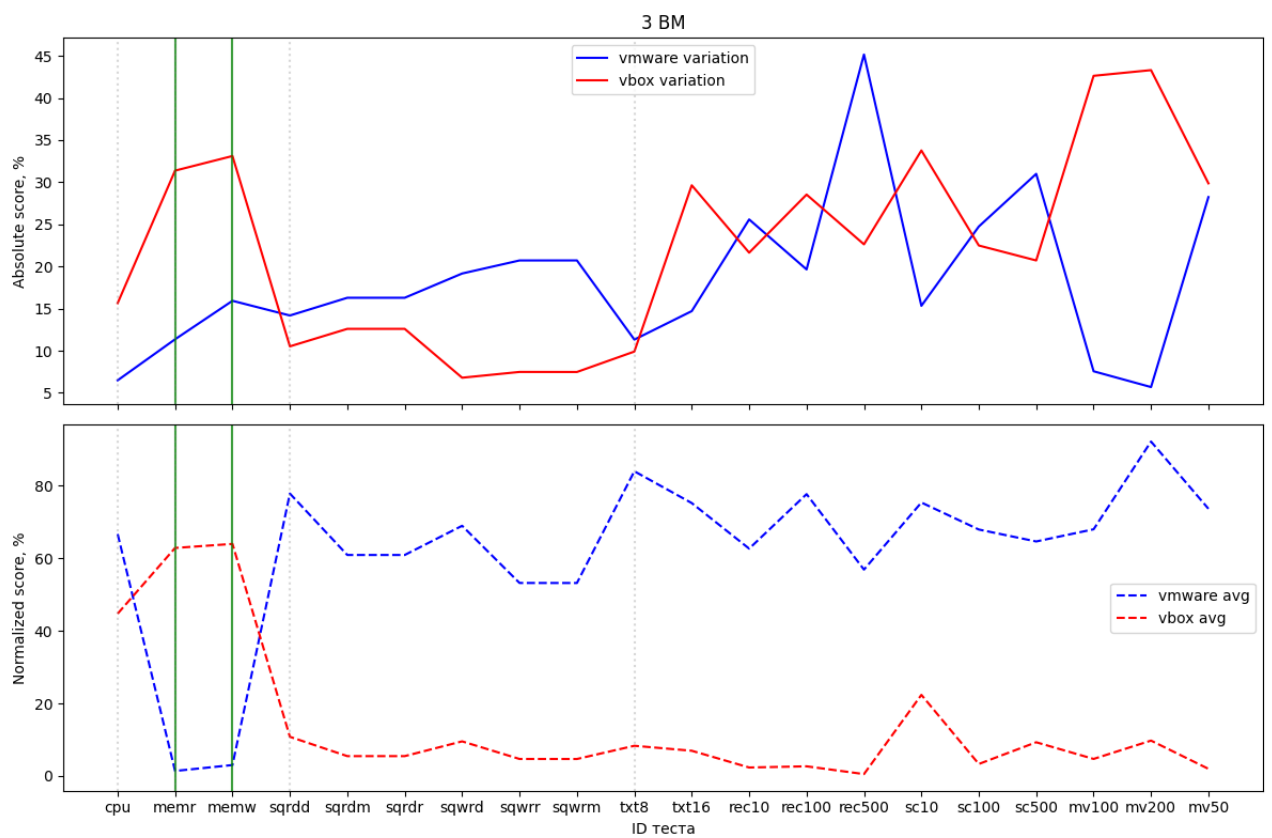


Рисунок 0-3 – Коэффициент вариации и среднее нормированное 3 VM

Для двух или трех одновременно работающих виртуальных машин динамика не меняется. Поэтому на основании рисунков 3.1-3.3 можно привести обобщенные выводы:

- VMware выигрывает по производительности процессора, дисковой подсистемы и 2D-графики;
- VirtualBox выигрывает по производительности оперативной памяти;
- Использование процессора и оперативной памяти стабильнее у VMware;
- Использование дисковой подсистемы стабильнее у VirtualBox.

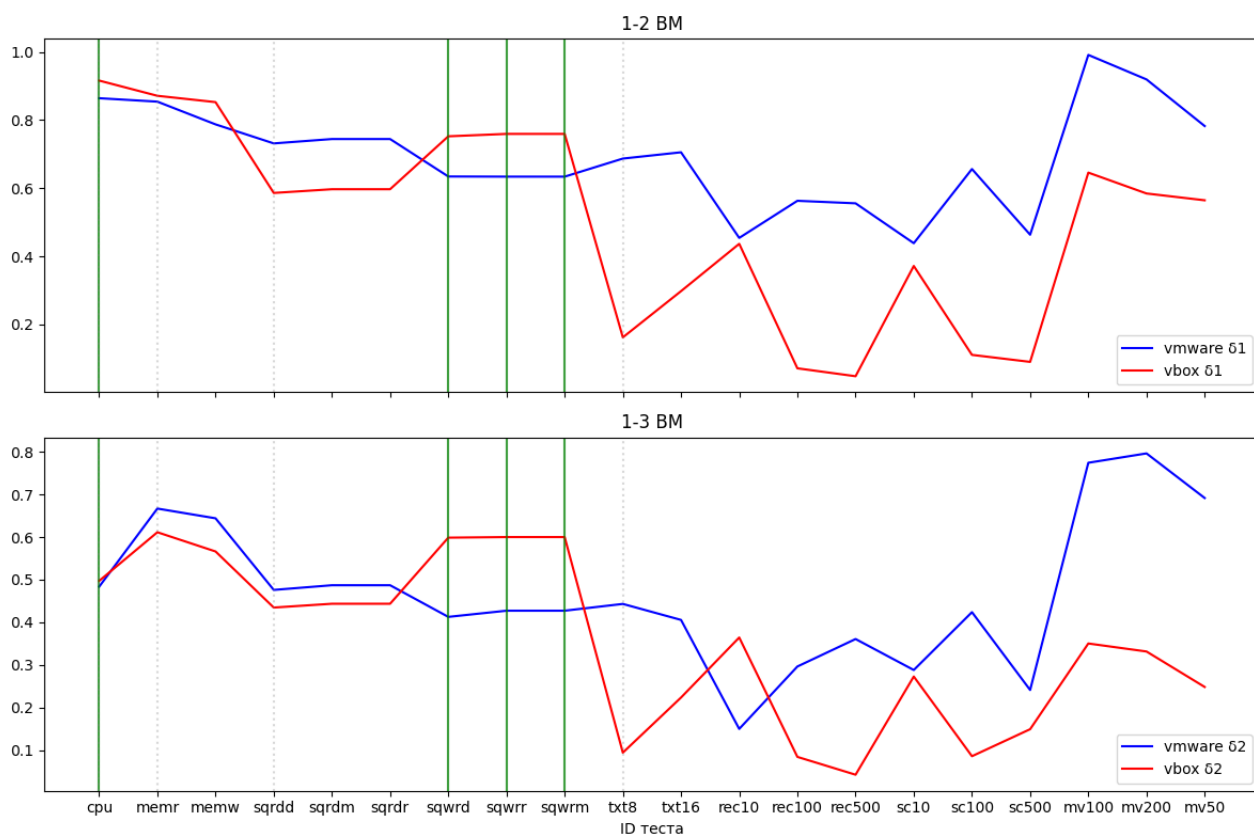
Про стабильность 2D-графики ничего сказать нельзя: оба гипервизора держат одну позицию.

Далее рассмотрим падение производительности при увеличении количества одновременно работающих виртуальных машин. Пусть  $x_1$ ,  $x_2$ ,  $x_3$  – средние значения результатов конкретного теста, которые мы уже рассматривали и в таблице, и на графиках. Тогда о величине падения производительности можно судить по относительным характеристикам ( $\delta_1$ ,  $\delta_2$ ):

$$\delta_1 = \frac{x_2}{x_1}; \quad \delta_2 = \frac{x_3}{x_1}$$

Очевидно, что производительность падает с увеличением числа ВМ, поэтому справедливо неравенство:  $\delta_2 < \delta_1 < 1$ .

На графике (рис. 3-4) отображены относительные характеристики. Зеленой линией помечены точки, в которых VirtualBox лучше конкурента по всем двум метрикам.



*Рисунок 0-4 – Величина падения производительности при нескольких VM относительно 1 VM*

На рисунке видно, что у VirtualBox есть несколько выигрышных позиций: по эффективности процессора и последовательной записи в файловой системе. Однако в основном стабильность системы при увеличении числа VM лучше реализована у VMware.

Наблюдаемое преимущество у гипервизора VirtualBox в тестах оперативной памяти при общем превосходстве VMware по другим категориям может быть обусловлено фоновой нагрузкой окружения хоста. В проведенной серии измерений гипервизоры работали при разных графических стеках. Дело в том, что даже современная версия VMware Workstation поддерживается только в графической подсистеме X11, в то время как Oracle VirtualBox запускается на новом стандарте Wayland, который эффективно показывает себя в потреблении ресурсов. Этот нюанс необходимо учитывать системным администраторам, разворачивающим виртуальные машины на Linux-серверах.

## 2. Сравнение эффективности гипервизоров по потреблению ресурсов хоста

В табл. 5 представлено медианное значение загрузки ресурсов хоста при использовании гипервизоров. Жирным выделены лучшие значения по сравнению с конкурирующим гипервизором. Все метрики представлены в процентах.

Так как при экспериментальном тестировании 3D-ускорение было отключено, нагрузка при 2D-тестах графики распределялась на процессор хоста.

*Таблица 5 – Медианная загрузка ресурсов хоста при использовании VMware и VirtualBox*

vmware / vbox, %	1 VM	2 VM	3 VM
CPU	30 / <b>26</b>	55 / <b>51</b>	82 / <b>78</b>
RAM	32 / <b>29</b>	<b>42</b> / 46	<b>58</b> / 65
DISK IO	17 / <b>5</b>	17 / <b>10</b>	17 / <b>10</b>
X11-2D	<b>21</b> / 29	<b>40</b> / 60	<b>60</b> / 73

Повышенная производительность VMware у процессора и дисковой подсистемы не берется из ниоткуда: заметно повышенное потребление ресурсов. Тем не менее производительность графической подсистемы X11 вместе с более экономной нагрузкой на процессор остается в лидирующей позиции у гипервизора VMware.

При запуске двух и трех виртуальных машин потребление ресурсов, очевидно, увеличивается, однако рост носит сублинейный характер и бывает меньше, чем «в 2 или 3 раза». Это может быть обусловлено особенностями работы процессорных планировщиков гипервизора и гостевой ОС, а также:

- I/O-ожидания, приводящие к неполной нагрузке ресурсов;
- разной степенью конкуренции за память и дисковые ресурсы;

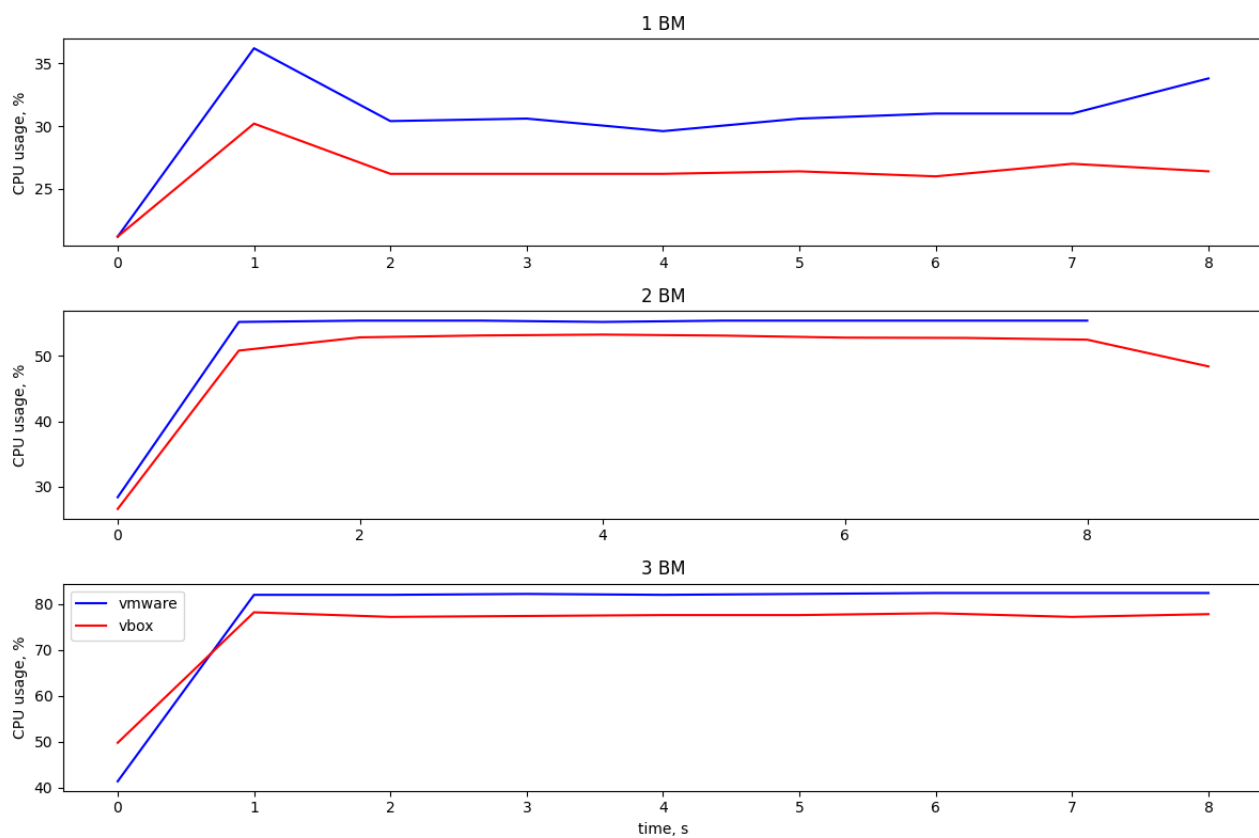
— асинхронностью VM-exit (выход из изолированного мира гостевой системы в реальное окружение хост-системы) и системных обращений [17].

Из-за этого гипервизор всегда имеет возможность перераспределять небольшие интервалы простоя между VM, сглаживая конкуренцию за ресурсы. Особенно данная тенденция наблюдается в тестах на оперативную память, что может быть связано с высокой долей разделяемых страниц (кэш, общие библиотеки), а также с тем, что часть памяти фактически используется как файловый кэш и перераспределяется динамически.

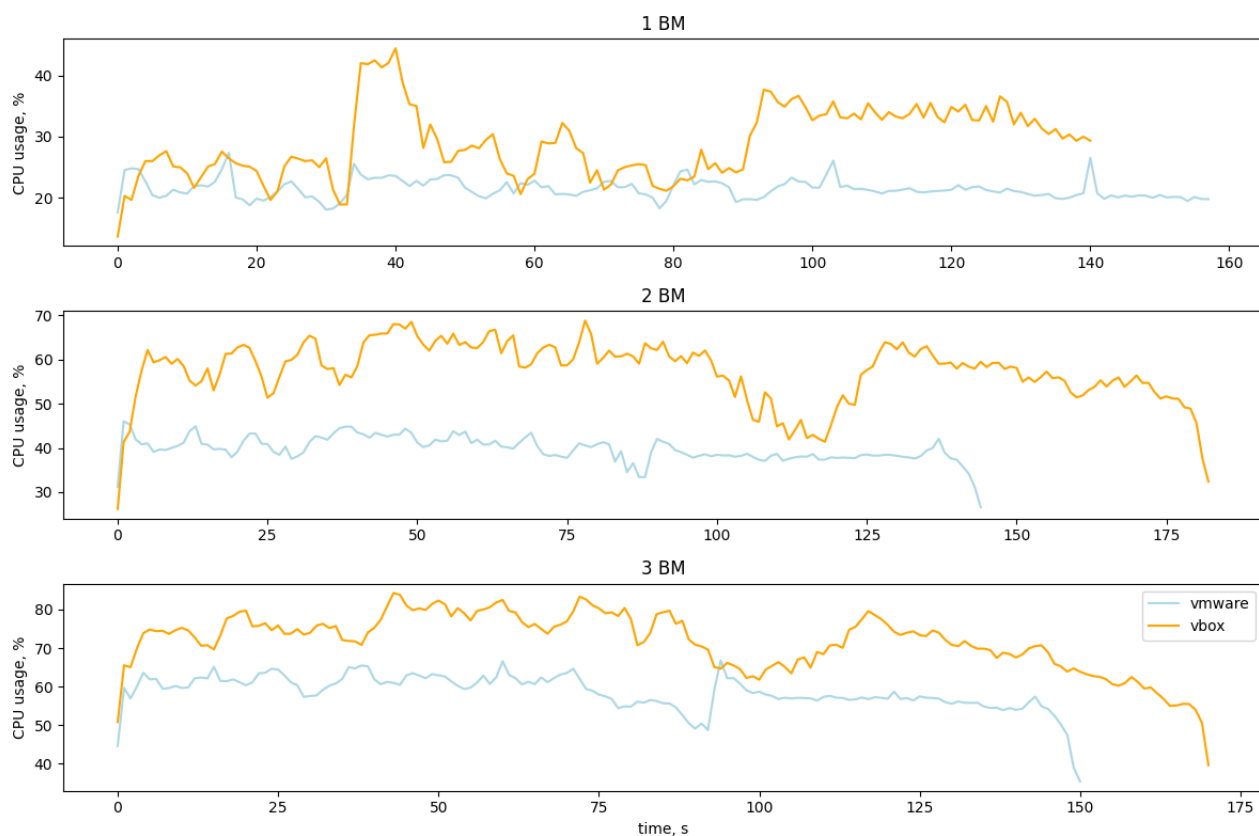
Для X11-2D наблюдается линейный рост нагрузки, поскольку отрисовка и обработка графических операций масштабируются с количеством активных GUI-сеансов и числом графических событий. Вычислительные тесты на CPU показывают рост немного меньше, чем линейный (у VMware). Это свидетельствует о грамотном использовании простоев, то есть сведении их к минимуму, когда в тот момент задача передается другой виртуальной машине.

В отличие от этого, дисковая нагрузка остается примерно постоянной, что указывает на доминирование кэширования и буферизации, а также на то, что синтетические I/O-операции в выбранной конфигурации ограничиваются не числом VM, а пропускной способностью подсистемы хранения и политиками планирования ввода-вывода.

Ниже рассматривается динамика загрузки процессора хост-системы при CPU-тестах (рис. 3-5) и при графических тестах (рис. 3-6). Значения в конкретный промежуток времени усреднены между повторами тестов.



*Рисунок 0-5 – Временная зависимость загрузки процессора в течение CPU-тестов*



*Рисунок 0-6 – Временная зависимость загрузки процессора в течение 2D-тестов*

Графики временной зависимости еще раз подтверждают превосходство VirtualBox в категории CPU и превосходство VMware в категории X11-2D. Заметно, что гипервизору VirtualBox чаще всего требуется больше времени на обработку бенчмарков.

### 3. Многокритериальная оценка гипервизоров

Для выбора предпочтительного гипервизора в условиях многокритериальности применен метод анализа иерархий Саати [18]. Метод основан на декомпозиции задачи принятия решения на иерархические уровни и последующей агрегации локальных приоритетов альтернатив в глобальные веса.

Цель — сравнение альтернатив (гипервизоров) по совокупности показателей производительности и эффективности при одинаковых конфигурациях виртуальных машин.

Иерархия включает два уровня критериев:

Критерии первого уровня (показатели качества выполнения тестов):

- $K_1$  — критерий производительности: нормированное среднее значение результата теста ВМ;
- $K_2$  — критерий стабильности: коэффициент вариации результата теста ВМ;
- $K_3$  — критерий эффективности: медианное значение потребления ресурсов хоста.

Веса первого уровня определялись на основе матрицы парных сравнений критериев:

$$\begin{pmatrix} 1 & 3 & 4,5 \\ 0,33 & 1 & 1,5 \\ 0,22 & 0,67 & 1 \end{pmatrix}$$

Наибольший вклад отводится производительности как основному показателю качества, при этом стабильность и эффективность учитываются как корректирующие характеристики предсказуемости и стоимости достижения производительности. Веса критериев приведены в табл. 6.

*Таблица 6 – Веса критериев второго уровня*

<b>Критерий</b>	<b>Направление оптимизации</b>	<b>Вес</b>
Среднее значение теста	максимизация*	0,64
Коэффициент вариации теста	минимизация	0,22
Медиана нагрузки ресурса хоста	минимизация	0,14

(\*) исключением являются метрики, измеряемые в мс – минимизация.

Критерии первого уровня (ресурсные категории тестирования):

- $C_1$  — CPU,
- $C_2$  — RAM,
- $C_3$  — DISK IO,
- $C_4$  — X11-2D.

Матрица парных сравнений критериев второго уровня следующая:

$$\begin{pmatrix} 1 & 1,5 & 2 & 3 \\ 0,67 & 1 & 1,33 & 2 \\ 0,5 & 0,75 & 1 & 1,5 \\ 0,33 & 0,5 & 0,67 & 1 \end{pmatrix}$$

В режиме нормальной нагрузки гостевой ОС процессор – один из самых нагружаемых ресурсов, далее – оперативная память и т.д. Полученные веса критериев занесены в табл. 7.

Таблица 7 – Веса критериев первого уровня

Критерий	Направление оптимизации	Вес
CPU	максимизация	0,400
RAM	максимизация	0,267
DISK IO	максимизация	0,200
X11-2D	максимизация	0,133

Отметим, что обе матрицы парных сравнений имеют допустимый индекс согласованности (ИС=0) [18, с. 25], поэтому рассчитываемые веса имеют смысл.

Для каждой категории первого уровня рассчитываются локальные приоритеты альтернатив по трем критериям второго уровня  $K_i$ . Далее полученные локальные оценки агрегируются по весам второго уровня:

$$S_{C_j}(A) = \sum_{i=1}^3 w_i^{(2)} \cdot p_{i,j}(A), \quad [16]$$

где  $p_{i,j}(A)$  — локальный приоритет альтернативы  $A$  по критерию  $K_i$  в категории  $C_j$ .

Итоговая (глобальная) оценка альтернативы определяется сверткой по критериям первого уровня:

$$S(A) = \sum_{j=1}^4 w_j^{(1)} \cdot S_{C_j}(A), \quad [16]$$

Интегральная оценка гипервизоров была получена путем агрегации локальных приоритетов взвешенной суммой (WSM), вычисленных в рамках иерархической модели Саати. Так как альтернативы сравниваются через экспериментальные данные, нет нужды в составлении матриц парных сравнений альтернатив.

Столбчатая диаграмма имеет агрегированную полезность по 4 ресурсам в модели первого уровня и итоговую стоимость (cost) в модели второго уровня (рис. 3-7).

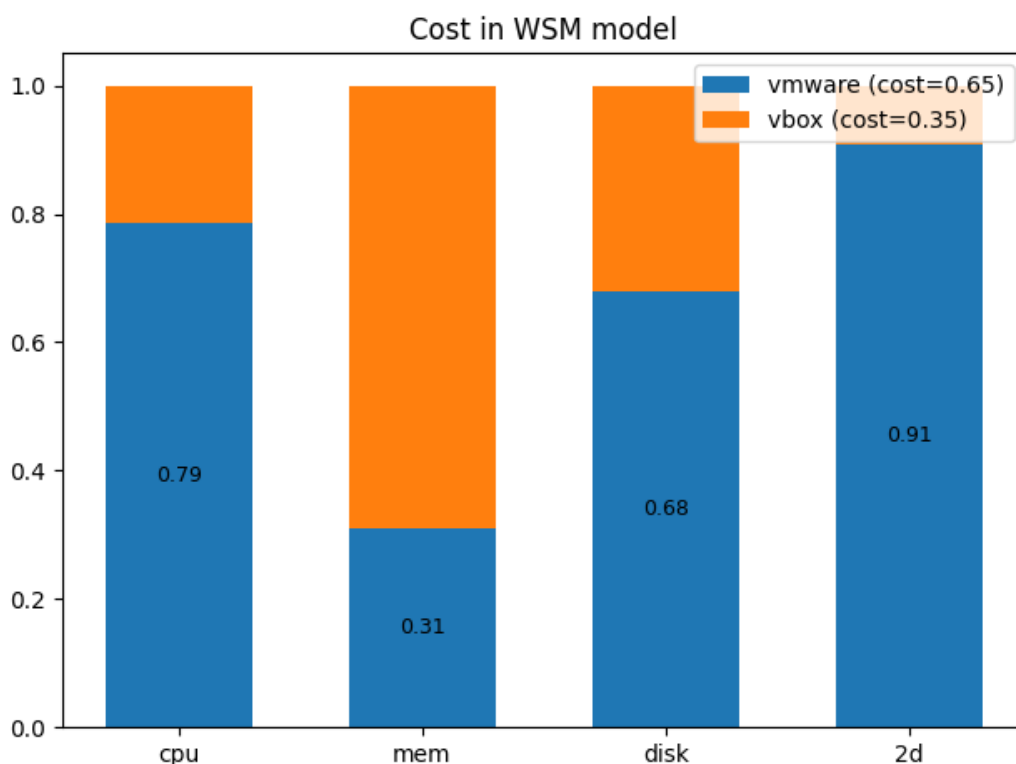


Рисунок 0-7 – Оценка полезности гипервизоров по каждому ресурсу

Наиболее доминирующее положение VMware имеет в категориях X11-2D и CPU. Гипервизор обеспечивает хорошую производительность дисковой подсистемы, хотя взамен требует больше ресурсов. Отрыв VirtualBox в категории RAM может быть нивелирован переходом приложения VMware Workstation на новый протокол для организации графического сервера Wayland.

По результатам экспериментального тестирования производительности VMware демонстрирует более высокие показатели производительности по

большинству критериев. Преимущество VMware объясняется особенностями его архитектуры виртуализации, использованием низкоуровневых модулей взаимодействия с хостовой ОС (vmmon, vmnet) и более эффективной работой диспетчера виртуальной памяти при обработке аппаратных инструкций виртуализации (VT-x/AMD-V) [17].

VirtualBox показывает повышенные задержки в сценариях, насыщенных I/O-операциями и 2D-графическими вызовами. Хотя для учебного пользования вполне подойдет VirtualBox, VMware незаменим в крупном бизнесе, в котором критически важна производительность гипервизора.

## Заключение

В ходе исследования была разработана и апробирована методика комплексного тестирования производительности виртуальных машин, включающая автоматизированный запуск синтетических нагрузок, сбор временных рядов показателей, их статистическую обработку и интегральную оценку на основе иерархической модели Саати. Предложенный подход позволяет учитывать не только среднюю производительность, но и устойчивость результатов, а также эффективность использования ресурсов хостовой системы.

Экспериментальные результаты показали, что гипервизор VMware Workstation в большинстве случаев демонстрирует более высокие показатели производительности по CPU-, дисковым и 2D-графическим тестам, а также более предсказуемое поведение при увеличении числа одновременно работающих виртуальных машин. Эти результаты хорошо согласуются с выводами сравнительных исследований гипервизоров второго типа, представленными в работах [1,2,8].

В то же время в тестах оперативной памяти в отдельных конфигурациях зафиксировано преимущество Oracle VirtualBox, что подтверждает выводы предыдущих работ о высокой чувствительности бенчмарков памяти к особенностям планирования виртуальных процессоров и конкуренции за кэш-ресурсы хоста [4,5]. Наблюдаемое сублинейное масштабирование нагрузки по памяти и практически линейный рост нагрузки для 2D-графики также соответствуют результатам, представленным в современных публикациях по анализу масштабируемости виртуализированных систем [6,7].

В целом полученные результаты находятся в хорошем соответствии с опубликованными данными и дополняют их за счет применения многокритериального подхода и анализа динамики нагрузки во времени. Разработанная методика может применяться для оценки производительности других гипервизоров и версий программных платформ, а также при выборе

оптимальной среды виртуализации для конкретных задач — от учебных лабораторий до корпоративных инфраструктур.

### Литература

1. **Djordjević V., Jovičić I., Kraljević N., Timčenko V.** Comparison of VMware Workstation, VirtualBox and MS Hyper-V hypervisor performance with MS Windows OS based guests // *Proceedings of IcETRAN*. 2023. С. 1–3.
2. **Revels M., Ciampa M.** Benchmarking comparison of VMware Workstation and Sun VirtualBox OSE // *ACET Journal*. 2011.
3. **Reddy R., Banerjee A., Padmanabhuni S.** Performance evaluation of virtualization technologies // *International Journal of Computer Applications*. 2013.
4. **Khaji M. и др.** Performance analysis of memory-intensive workloads in virtualized environments // *Journal of Cloud Computing*. 2019.
5. **Chae S. и др.** Analysis of memory and CPU performance in virtual machines // *IEEE Access*. 2020.
6. **Morabito R. и др.** Evaluating performance of containers and virtual machines // *IEEE Cloud Computing*. 2015.
7. **Abraham S. и др.** Comparative study of container-based and hypervisor-based virtualization // *Future Generation Computer Systems*. 2018.
8. **Djordjević V. и др.** Performance evaluation of VMware and VirtualBox under I/O workloads // *Theoretical and Applied Science*. 2021.
9. **Altaro Ltd.** VMware vs VirtualBox: performance and architectural differences : technical white paper. 2022.
10. **Sharma S., Poddar R., Bisen D.** Performance comparison of the type-2 hypervisors VirtualBox and VMware Workstation // *International Journal of Computer Science and Information Technologies*. 2015. Т. 6, № 2. С. 1150–1154.

11. **Broadcom Inc.** VMware Workstation Pro 17. Configuring Virtual Machine Hardware Settings : [Электронный ресурс]. URL:  
<https://techdocs.broadcom.com/us/en/vmware-cis/desktop-hypervisors/workstation-pro/17-0/using-vmware-workstation-pro/configuring-virtual-machine-hardware-settings.html>  
(дата обращения: 09.11.2025).
12. **Oracle Corporation.** VirtualBox User Manual. Chapter 3. Configuring Virtual Machines : [Электронный ресурс]. URL:  
<https://www.virtualbox.org/manual/ch03.html>  
(дата обращения: 09.11.2025).
13. **Тестирование производительности Linux сервера с Sysbench** : [Электронный ресурс]. URL:  
<https://serveradmin.ru/testirovanie-proizvoditelnosti-linux-servera-s-sysbench/>  
(дата обращения: 11.11.2025).
14. **X11PERF(1).** Manual page : [Электронный ресурс]. URL:  
<https://www.x.org/archive/X11R7.5/doc/man/man1/x11perf.1.html>  
(дата обращения: 11.11.2025).
15. **PSSH(1).** Linux manual page : [Электронный ресурс]. URL:  
<https://linux.die.net/man/1/pssh>  
(дата обращения: 11.11.2025).
16. **REST Journal on Data Analytics and Artificial Intelligence.** Optimization techniques: strategies for enhanced performance using WSM method. 2025. С. 16–20.
17. **Choosing the best benchmarks: A comparison of VirtualBox vs. VMware virtual machines** : [Электронный ресурс]. URL:  
<https://raywoodcockslatest.wordpress.com/2018/03/06/vm-performance/>  
(дата обращения: 14.11.2025).
18. **Саати Т.** Принятие решений. Метод анализа иерархий : пер. с англ. / Р. Г. Вачнадзе. М. : Радио и связь, 1993. 278 с.

## Literature

1. Djordjević B., Jovičić I., Kraljević N., Timčenko V. Comparison of VMware Workstation, VirtualBox and MS Hyper-V hypervisor performance with MS Windows OS based guests. Proceedings of IcETRAN, 2023, pp. 1–3.
2. Revels M., Ciampa M. Benchmarking comparison of VMware Workstation and Sun VirtualBox OSE. ACET Journal, 2011.
3. Reddy R., Banerjee A., Padmanabhuni S. Performance evaluation of virtualization technologies. International Journal of Computer Applications, 2013.
4. Khaji M. et al. Performance analysis of memory-intensive workloads in virtualized environments. Journal of Cloud Computing, 2019.
5. Chae S. et al. Analysis of memory and CPU performance in virtual machines. IEEE Access, 2020.
6. Morabito R. et al. Evaluating performance of containers and virtual machines. IEEE Cloud Computing, 2015.
7. Abraham S. et al. Comparative study of container-based and hypervisor-based virtualization. Future Generation Computer Systems, 2018.
8. Djordjević B. et al. Performance evaluation of VMware and VirtualBox under I/O workloads. Theoretical and Applied Science, 2021.
9. Altaro Ltd. VMware vs VirtualBox: performance and architectural differences. Technical white paper, 2022.
10. Sharma S., Poddar R., Bisen D. Performance comparison of the type-2 hypervisors VirtualBox and VMware Workstation. International Journal of Computer Science and Information Technologies, 2015, vol. 6, no. 2, pp. 1150–1154.
11. Broadcom Inc. VMware Workstation Pro 17. Configuring Virtual Machine Hardware Settings. Available at:  
<https://techdocs.broadcom.com/us/en/vmware-cis/desktop-hypervisors/workstation-pro/17-0/using-vmware-workstation->

pro/configuring-virtual-machine-hardware-settings.html

(accessed 09 Nov 2025).

12. Oracle Corporation. VirtualBox User Manual. Chapter 3. Configuring Virtual Machines. Available at:  
<https://www.virtualbox.org/manual/ch03.html>  
(accessed 09 Nov 2025).
13. Linux server performance testing with Sysbench. Available at:  
<https://serveradmin.ru/testirovanie-proizvoditelnosti-linux-servera-s-sysbench/>  
(accessed 11 Nov 2025).
14. X11PERF(1). Manual page. Available at:  
<https://www.x.org/archive/X11R7.5/doc/man/man1/x11perf.1.html>  
(accessed 11 Nov 2025).
15. PSSH(1). Linux manual page. Available at:  
<https://linux.die.net/man/1/pssh>  
(accessed 11 Nov 2025).
16. REST Journal on Data Analytics and Artificial Intelligence. Optimization techniques: strategies for enhanced performance using the WSM method, 2025, pp. 16–20.
17. Choosing the best benchmarks: A comparison of VirtualBox vs. VMware virtual machines. Available at:  
<https://raywoodcockslatest.wordpress.com/2018/03/06/vm-performance/>  
(accessed 14 Nov 2025).
18. Saaty T. Decision Making: The Analytic Hierarchy Process. Moscow: Radio i Svyaz, 1993. 278 p. (in Russian).