

**МОДЕЛИ НАДЕЖНОСТИ КОМПОНЕНТОВ ОБЛАЧНОГО ДАТА-ЦЕНТРА**

**Яновская О. В, Харченко В. С.**

*Национальный аэрокосмический университет им. Н. Е. Жуковского "ХАИ" (г. Харьков)*

*Рассматривается типовая архитектура облачного дата-центра, производится расчет показателей надежности компонентов системы на основе марковских моделей с различными входными параметрами.*

**Постановка проблемы.** Типовая архитектура облачного дата-центра (ЦОД) имеет иерархическую структуру, состоящую из трех основных уровней [1]: ядра, агрегации и доступа. Высокая готовность виртуального дата-центра достигается за счет дублирования сетевых устройств на каждом уровне и использования резервных маршрутов, а также за счет методов балансировки нагрузки. Возможность применения различных вариантов аппаратной и программной инфраструктуры ЦОД обуславливают необходимость проведения оценок показателей надежности для определения методов улучшения этих показателей.

**Анализ последних исследований и публикаций.** Анализ [2] показывает, что наиболее распространенными в виртуальном компьютеринге являются следующие виды отказов:

- аппаратные отказы хостов (Hardware Faults,  $f_h$ ),
- отказы в работе операционной системы (Operating System Faults,  $f_{OS}$ ),
- отказы модуля Virtual Machine Monitor/Hypervisor (VMM Faults,  $f_{VMM}$ ),
- программные отказы гостевых операционных систем на виртуальных машинах (Guest Operating System Faults,  $f_{VM}$ ),
- отказы прикладного ПО (Application Faults,  $f_a$ ).

К физическим дефектам относятся неисправности аппаратных средств хостов  $f_h$ . Наиболее распространенными проявлениями таких неисправностей является «зависание» системы, невозможность загрузки или полная потеря работоспособности хоста. Аппаратные неисправности являются наиболее вредоносным типом неисправностей: их проявление может повлечь увеличение расходов в связи с возможной заменой отказавших аппаратных компонентов. Отказы хостовых  $f_{OS}$  и гостевых  $f_{VM}$  операционных систем, модуля VMM  $f_{VMM}$  и прикладного ПО  $f_a$  относятся к дефектам программных компонент.  $f_{OS}$  и  $f_{VM}$  проявляются в виде сбоев работы ядра ОС (ошибками BSOD, Oops, Panics),  $f_{VMM}$  – в виде прекращения работы гостевых ОС и всех запущенных на них процессов,  $f_a$  – в виде «зависания» приложения, запущенного на ВМ, с которым работает пользователь. В рамках данной работы не учитываются дефекты взаимодействия (interaction faults) вследствие атак на уязвимости или физических воздействий. Все рассмотренные типы дефектов и неисправностей приводят к нарушениям работы всей системы (errors)  $e_h, e_{OS}, e_{VM}, e_{VMM}, e_a$ , которые обусловлены соответствующими отказами [3]. В результате этого происходит переход системы в частично-работоспособное состояние, связанное с

проявлением этих отказов: неисправности аппаратных средств сервера, отказ его ОС и модуля VMM приводят к отсутствию доступа ко всем виртуальным машинам на этом сервере и отказу в обслуживании всех сервисов, которые на них запущены; отказы гостевой ОС и прикладного ПО на виртуальной машине – к потере доступа к этой ВМ для пользователя.

Специфику рассмотренных типов отказов необходимо учесть при оценке показателей надежности таких компонентов облачной архитектуры, как сервера и виртуальные машины.

**Цель статьи** – разработка и анализ марковских моделей надежности сервера и виртуальной машины облачного ЦОД.

**Основные материалы исследования. Модель сервера.** Для оценки надежности такой сложной инфраструктуры, как виртуальный дата-центр, необходимо оценить показатели надежности его основных компонентов.

Модель работы сервера виртуального дата-центра может быть представлена в графа состояний и переходов на рис. 4:

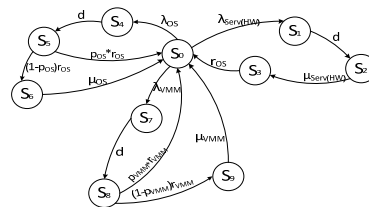


Рисунок 1 - Граф состояний и переходов модели сервера облачного ЦОД

Допущения: отказы в системе независимы; отказы разных типов не накладываются по времени; отказы типа  $i$  не возникают до устранения отказов типа  $j$  (при  $i \neq j$ ;  $i, j$  изменяются от 1 до  $n$ , где  $n$  – число типов отказов); средства диагностирования и мониторинга идеальны: средства обнаружения абсолютно безотказны; средства переключения идеальны.

Пусть изначально система (сервер) находится в полностью работоспособном состоянии  $S_0$ . С интенсивностью  $\lambda_{SerV(HW)}$  произойдет отказ одного из аппаратных компонентов сервера и переход в состояние  $S_1$ . Отказ будет обнаружен посредством, как правило, используемого в системе механизма «heart-beat» в течение времени  $T_d = 20$  секунд. Произойдет переход в состояние обнаруженного отказа  $S_2$  с интенсивностью  $d = 1/T_d$ . После этого потребуются вызов технического специалиста для восстановления (замены) отказавших

аппаратных компонент (переход в состояние  $S_3$  с интенсивностью  $\mu_{\text{Serv(HW)}}$ ) с последующим перезапуском операционной системы ( $r_{\text{OS}}$ ). Величина  $r_{\text{OS}}$  определяется как  $1/T_{\text{OS}}$ , где  $T_{\text{OS}}$  – среднее время перезагрузки ОС. Таким образом система возвратится в работоспособное состояние  $S_0$ . Аналогично, при отказе операционной системы (состояние  $S_4$ ), отказ будет обнаружен с интенсивностью  $d$  в состоянии  $S_5$ . В случае удачного рестарта операционной системы с вероятностью  $p_{\text{OS}}$  и интенсивностью  $p_{\text{OS}}*r_{\text{OS}}$  система перейдет в состояние  $S_0$ . В случае неудачного рестарта ОС (переход по дуге в  $S_6$  с интенсивностью  $(1-p_{\text{OS}})*r_{\text{OS}}$ ) потребуются дополнительные действия технического специалиста по восстановлению работоспособности ОС, после чего система восстановится из состояния  $S_6$  в состояние  $S_0$  с интенсивностью  $\mu_{\text{OS}}$ . Таким же образом можно описать поведение системы при отказе гипервизора (VMM-модуля).

Если учесть, что каждое состояние системы в очередной момент времени  $t + d$  зависит только от текущего состояния в момент времени  $t$  и принять, что все времена являются детерминированными, то граф состояний и переходов рассматриваемой модели можно представить как марковский процесс с непрерывным временем и дискретными состояниями.

В расчетах для данной модели использовалась встроенная функция ППП Mathcad - Rkadapt, реализующая решение системы СДУ методом Рунге-Кутты на временном интервале от 0 до 10000 часов. Исходные данные выбирались как усредненные значения результатов экспериментальных исследований, опубликованных в [4] и [5].

Таблица 1 – Входные параметры модели сервера

Параметры	Наборы значений		
	I	II	III
$\lambda_{\text{Serv(HW)}}$	$1/300000 \text{ ч}^{-1}$	$1/300000 \text{ ч}^{-1}$	$1/300000 \text{ ч}^{-1}$
$\lambda_{\text{OS}}$	$1/1440 \text{ ч}^{-1}$	$1/1440 \text{ ч}^{-1}$	$1/1440 \text{ ч}^{-1}$
$\lambda_{\text{VMM}}$	$1/2880 \text{ ч}^{-1}$	$1/2880 \text{ ч}^{-1}$	$1/2880 \text{ ч}^{-1}$
$d$	$1/20 \text{ с}^{-1}$	$1/20 \text{ с}^{-1}$	$1/20 \text{ с}^{-1}$
$\mu_{\text{Serv(HW)}}$	$1 \text{ ч}^{-1}$	$1 \text{ ч}^{-1}$	$1 \text{ ч}^{-1}$
$\mu_{\text{OS}}$	$1/2 \text{ ч}^{-1}$	$1/2 \text{ ч}^{-1}$	$1 \text{ ч}^{-1}$
$\mu_{\text{VMM}}$	$1/2 \text{ ч}^{-1}$	$1/2 \text{ ч}^{-1}$	$1 \text{ ч}^{-1}$
$r_{\text{OS}}$	$1/10 \text{ мин}^{-1}$	$1/163 \text{ с}^{-1}$	$1/163 \text{ с}^{-1}$
$r_{\text{VMM}}$	$1/10 \text{ мин}^{-1}$	$1/163 \text{ с}^{-1}$	$1/163 \text{ с}^{-1}$
$p_{\text{OS}}$	0.9	0.9	0.9

При изменении параметров в данном диапазоне результаты моделирования показывают, что с уменьшением времен восстановления до 1 часа можно получить увеличение функции готовности в 4 знаке:

Таблица 2 – Выходные параметры модели сервера

Наборы входных параметров	Коэффициент готовности
I	0.999609
II	0.999735
III	0.999839

**Модель виртуальной машины.** Динамику работы виртуальной машины (ВМ) с учетом возможности ее рестарта в случае отказа сервера, на котором она запущена, на другом физическом сервере (High Availability), а также возможности ее мигрирования в случае восстановления работоспособности первого (Live Migration) также можно представить в виде марковской модели:

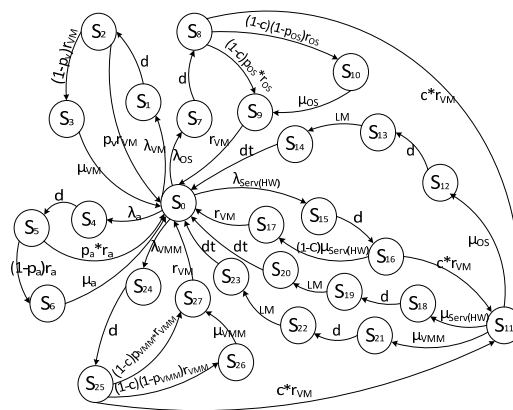


Рисунок 5 - Модель состояний и переходов работы ВМ

Таблица 3 – Входные параметры модели ВМ

Входные параметры модели:	Значение
$\lambda_{\text{Serv(HW)}}$	$1/300000 \text{ ч}^{-1}$
$\lambda_{\text{OS}}$	$1/1440 \text{ ч}^{-1}$
$\lambda_{\text{VM}}$	$1/2880 \text{ ч}^{-1}$
$\lambda_{\text{VMM}}$	$1/2880 \text{ ч}^{-1}$
$\lambda_a$	$1/336 \text{ ч}^{-1}$
$d$	$1/20 \text{ с}^{-1}$
$\mu_{\text{Serv(HW)}}$	$1 \text{ ч}^{-1}$
$\mu_{\text{OS}}$	$1/2 \text{ ч}^{-1}$
$\mu_{\text{VM}}$	$1/0.5 \text{ ч}^{-1}$
$\mu_{\text{VMM}}$	$1/2 \text{ ч}^{-1}$
$\mu_a$	$1 \text{ ч}^{-1}$
$r_{\text{OS}}$	$1/163 \text{ с}^{-1}$
$r_{\text{VM}}$	$1/80 \text{ с}^{-1}$
$r_{\text{VMM}}$	$1/163 \text{ с}^{-1}$
$r_a$	$1/20 \text{ мин}^{-1}$
$c$	0.9
$p_{\text{OS}}$	0.9
$p_{\text{VM}}$	0.95
$p_{\text{VMM}}$	0.9
$p_a$	0.9
LM	$1/49.9 \text{ с}^{-1}$
dt	$1/9.5 \text{ мс}^{-1}$

Можно принять аналогичный ряд допущений, что и в предыдущей модели. В состоянии  $S_0$  виртуальная машина запущена на полностью работоспособном сервере. Из состояния  $S_0$  произойдет переход в состояние  $S_1$  в случае отказа виртуальной машины (например, гостевой операционной системы) по дуге с интенсивностью  $\lambda_{\text{VM}}$ . Отказ будет обнаружен через среднее время  $T_d$  с интенсивностью  $d = 1/T_d$ , после чего система перейдет в состояние  $S_2$ . Из этого со-

стояния в случае удачного рестарта виртуальной машины (переход с интенсивностью  $p_{VM} * r_{VM}$ ) система возвратится в состояние  $S_0$ , в случае неудачного рестарта (переход с интенсивностью  $(1-p_{VM}) * r_{VM}$ ) система окажется в состоянии  $S_3$ . Требуется вызов специалиста для полного восстановления виртуальной машины и возврата в работоспособное состояние  $S_0$  с интенсивностью  $\mu_{VM}$ , где  $\mu_{VM}$  – величина, обратная времени восстановления ВМ. Аналогично описывается поведение ВМ при отказе прикладного ПО (состояния  $S_4, S_5, S_6$  с соответствующими интенсивностями переходов  $\lambda_a, p_a * r_a, (1-p_a) * r_a, \mu_a$ ). Переходы из  $S_0$  в  $S_7, S_{15}, S_{24}$  с интенсивностями  $\lambda_{OS}, \lambda_{Serv(HW)}, \lambda_{VMM}$  соответственно обуславливают отказы родительской ОС, аппаратных компонент сервера и модуля VMM. Любой отказов из этой группы будет равнозначен отказу физического хоста, на котором запущена виртуальная. Обнаружение системой одного из перечисленных отказов определяет переход по дуге с интенсивностью  $d$ . Рассмотрим поведение системы после обнаружения отказа операционной системы на сервере в состоянии  $S_8$ . Пусть возможность рестарта виртуальной машины на другом сервере оценивается вероятностью  $s$ . Тогда с интенсивностью  $s * r_{vm}$  произойдет переход в  $S_{11}$ , где  $r_{vm}$  – интенсивность, обратная времени рестарта ВМ. В  $S_{11}$  виртуальная машина запущена на другом хосте, который будем считать безотказным. Если после этого события произойдет восстановление ОС на первом хосте с интенсивностью  $\mu_{OS}$  (переход  $S_{11} - S_{12}$ ) и обнаружение этого восстановления системой с интенсивностью  $d$  (переход  $S_{11} - S_{13}$ ). Тогда ВМ может мигрировать на первый сервер. Процесс миграции ВМ (Live Migration) обычно состоит из нескольких этапов. Первые два этапа (Pre-migration, Pre-copy) подразумевают подготовку копирования файлов ВМ на другой физический хост, в течение которых сервисы ВМ остаются доступными для пользователя. Если среднее время выполнения этих двух процессов равно  $T_{LM}$ , то интенсивность перехода LM в следующее состояние  $S_{14}$  будет определяться как  $1 / T_{LM}$ . Последних 2 этапа (Stop-and-copy, Post-migration Overhead) определяют процедуру остановки ВМ, копирования ее файлов на другой хост и активацию сервисов ВМ на новом физическом сервере. В течение времени  $T_{Dt}$  сервисы ВМ остаются недоступными. Затем с интенсивностью  $Dt = 1 / T_{Dt}$  система перейдет в работоспособное состояние  $S_0$ , в котором ВМ запущена на первом хосте. Рассмотрим ситуацию, когда ВМ не имеет возможности (свободных ресурсов в пуле) рестарта на другом физическом сервере, что определяется вероятностью  $(1-s)$ . Тогда из состояния  $S_8$ , в случае удачного рестарта операционной системы с вероятностью  $p_{OS}$  и интенсивностью  $p_{OS} * r_{OS}$  система перейдет в состояние  $S_9$ , после чего потребуется перезапуск ВМ с интенсивностью перехода  $r_{VM}$  в  $S_0$ . В случае неудачного рестарта ОС (переход по дуге в  $S_{10}$  с интенсивностью  $(1-p_{OS}) * r_{OS}$ ) потребуются действия по восстановлению работоспособности ОС, после чего система восстановится до состояния  $S_9$  с интенсивностью  $\mu_{OS}$ . После перегрузки ВМ система возвратится в состояние  $S_0$ .

Для отказов аппаратных компонент сервера и модуля VMM поведение системы описывается аналогичным образом. В расчетах использовалась встроенная функция ППП Mathcad, реализующая метод интегрирования "жестких" систем дифференциальных уравнений методом Розенброка. Коэффициент готовности системы для промежутка времени 1 год составляет 0.971583.

**Выводы.** Выполненный анализ показывает, что уровень готовности ЦОД зависит не только от надежных аппаратных средств, но и от специфики отказов программных средств платформ виртуализации.

#### Список использованных источников

1. Abts D. A guided tour of data-center networking / D. Abts, F. Bob. // Magazine Communications of the ACM. – 2012. – №55. – С. 44–51.
2. Akber F. Tips for Troubleshooting VMware ESX Server Faults [Электронный ресурс] / Faisal Akber // VMWORLD. – 2006. – Режим доступа до ресурсу: <http://www.vmware.com/vmtn/vmworld/sessions/>.
3. Технологии высокой готовности для программно-технических комплексов космических систем / [В. С. Харченко, О. Н. Одарущенко, Ю. Л. Поночовный та ін.]. – "Харьк. авиац. ин-т", нац. аэрокос. ун-т, 2010. – 372 с. – (Гос. центр регулирования качества поставок и услуг).
4. Dong S. K. Availability Modeling and Analysis of a Virtualized System / S. K. Dong, F. Machida, K. S. Trivedi. // 15th IEEE Pacific Rim International Symposium on Dependable Computing PRDC '09. – 2009. – С. 365–371
5. Fenn M. An Evaluation of KVM for Use in Cloud Computing / M. Fenn, M. A. Murphy, J. Martin. // ACM Transactions on Computational Logic. – 2011. – С. 1–7

#### Анотація

#### МОДЕЛІ НАДІЙНОСТІ КОМПОНЕНТІВ ХМАРНОГО ДАТА-ЦЕНТРУ

Яновська О. В., Харченко В. С.

*Розглядається типова архітектура хмарного дата-центру, проводиться розрахунок показників надійності компонентів на основі марковських моделей з різними вхідними параметрами.*

#### Abstract

#### RELIABILITY MODEL OF CLOUD DATACENTERS' COMPONENTS

O. Yanovskaya, V. Kharchenko

*The typical architecture of virtual data center is considered, the reliability of the system components based on the Markov's chain models is estimated with different input parameter.*