

УДК 004.75

## ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ ОБЛАЧНЫХ СЕРВЕРОВ И ОПТИМИЗАЦИЯ ИХ СТОИМОСТИ



**И.А. Трубин**

*к.т.н., ИТ-менеджер, банк Capital One, США  
член совета директоров некоммерческой организации  
Computer Measurement Group, США  
igor@Trub.in*

*Computer Measurement Group, Capital One Bank, USA  
E-mail: Igor@Trub.in*

### **И.А. Трубин**

*Он начал карьеру в 1979 году в качестве системного инженера IBM /370. В 1986 году он получил кандидатскую степень в области робототехники в Санкт-Петербургском политехническом университете (Россия), а затем около 12 лет работал преподавателем кафедры Автоматы в области робототехники. Он опубликовал более 30 статей и сделал несколько презентаций для конференций, связанных с областями робототехники и искусственного интеллекта. В 1999 году он переехал в США и работал в банке Capital One в качестве планировщика производственных мощностей ИТ отдела. Его первая статья в [www.CMG.org](http://www.CMG.org) была написана и представлена в 2001 году. Следующая, «Система обнаружения аномалий на основе технологии MASF», получила награду «Лучший доклад» на [www.CMG.org](http://www.CMG.org) 2002 и была представлена на UKCMG 2003 в Оксфорде, Англия. Он выступил с докладами на IBM z/Series Expo, [CMG.org](http://www.CMG.org) в Южной и Центральной Европе, в Канаде (ICPE/WOSP-C 2020) и провел несколько семинаров и мастер классов, посвященных его оригинальному методу обнаружения аномалий и точек изменения. Автор идеи бесплатного профайлера [www.Perfomalist.com](http://www.Perfomalist.com). Автор курса «Обнаружение аномалий производительности» (<https://cma1.teachable.com>). Проработал 2 года руководителем группы Capacity Planning в IBM, работал в SunTrust Bank в течение 3 лет, а затем в IBM более 2 лет в качестве старшего ИТ-архитектора/консультанта. Сейчас он работает в банке Capital One в качестве ИТ-менеджера в отделе облачной инженерии, а с 2015 года он является членом совета директоров [www.CMG.org](http://www.CMG.org). Он ведет свой технический блог на [www.Trub.in](http://www.Trub.in)*

**Аннотация.** Публичное облако имеет неограниченную емкость если у вас есть неограниченный бюджет для его покупки. Но реальность такова, что бюджеты никогда не бывают действительно безграничными, и нужно делать выбор облачных серверов правильного размера (типа), чтобы перестать тратить деньги на неиспользованный или ненужный облачный ресурс. В докладе будут обсуждены методы автоматического отслеживания расходов на использование облачных серверов, оптимальной настройки их загрузки и составления отчетов об использовании облачных ресурсов приложениями и бизнесами. Будут продемонстрированы методы оптимизации облачных ресурсов с точки зрения минимизации денежных затрат без потери производительности в применении как к отдельным приложениям (аппликациям) так и ко всей ИТ инфраструктуре крупной финансовой компании.

### Постановка задачи

Публичное облако имеет неограниченную емкость если у вас есть неограниченный бюджет для его покупки. Но реальность такова, что бюджеты никогда не бывают действительно безграничными, и нужно делать выбор облачных серверов правильного размера (типа), чтобы перестать тратить деньги на неиспользованный или ненужный облачный ресурс.

Приведённые ниже ИНСТРУМЕНТЫ для анализа стоимости облачных серверов и получения рекомендаций по изменению их размера не всегда решают поставленную задачу:

#### - Amazon AWS

o Оптимизатор от AWS (Amazon EC2 resource optimization recommendations/ AWS Cost Management)

Instance ID	Instance name	Finding	Current instance type	Current On-Demand price	Recommended instance type	Recommended On-Demand price
i-0fb9323080785de1e	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large	\$0.0832 per hour
i-0f4f4c06ad8afe81a	-	Over-provisioned	m5.2xlarge	\$0.384 per hour	r5.xlarge	\$0.252 per hour
i-0f277818dfef522e9	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large	\$0.0832 per hour
i-0ceb95ed248026d24	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large	\$0.126 per hour
i-0af9322ff627d7e8f	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large	\$0.126 per hour
i-07084b94d1bcf391b	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large	\$0.0832 per hour
i-069f6e837890db127	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large	\$0.0832 per hour
i-0218a45abd8b53658	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large	\$0.126 per hour

Рисунок 1. – Пример рекомендации по размеру EC2 от AWS Compute Optimizer о Советчик от AWS (*Trusted Advisor*)

#### - VMWARE CloudHealth

o Выбор правильного размера объекта (*Rightsizing*)

o Управление стоимостью (*Cost management*)

ПРОБЛЕМА: для растущего бизнеса инструменты управления затратами в большинстве случаев демонстрируют рост расходов независимо от усилий по оптимизации ресурсов. Типичная тенденция показана на рисунке ниже:

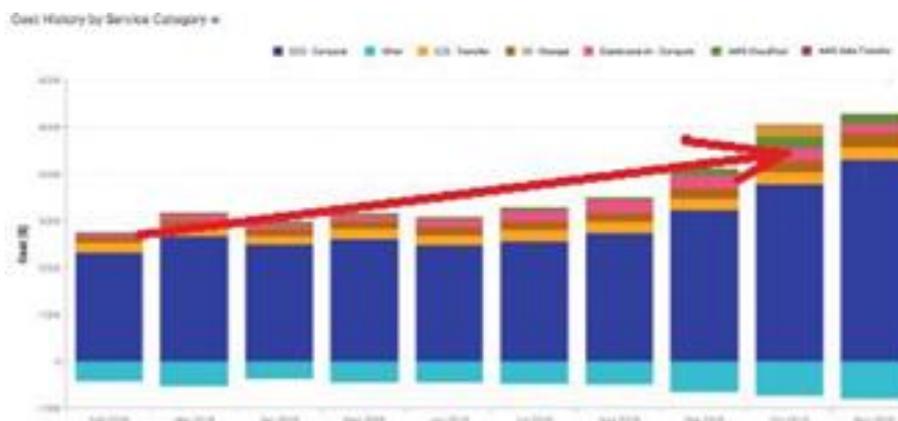


Рисунок 2. – Типичная тенденция

### Решение проблемы

Чтобы показать, насколько эффективно используется облако, предлагается другой, нормализованный подход.

Во-первых, отчет об использовании загрузки сервера в облаке должен охватывать четыре основных подсистемы (а не только процессор)

- вычислительная мощность
- оперативная память (RAM)
- пропускная способность дискового ввода-вывода
- полоса пропускания сети

Затем, чтобы показать, насколько эффективно используется вычислительная мощность, необходимо провести нормализацию и агрегирование всех типов (размеров) виртуальных серверов (EC2).

Один из подходов заключается в использовании AWS Эластичных Вычислительных Единиц (Elastic Compute Units - ECU). Это сопоставимый показатель «лошадиных сил» серверов, который можно получить из прайс-листа AWS EC2 (фрагмент списка показан ниже):

server type	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
m5.large	2	8	8 GiB	EBS Only	\$0.096 per Hour
m5.xlarge	4	16	16 GiB	EBS Only	\$0.192 per Hour
m5.2xlarge	8	31	32 GiB	EBS Only	\$0.384 per Hour
m5.4xlarge	16	60	64 GiB	EBS Only	\$0.768 per Hour
c5.large	2	9	4 GiB	EBS Only	\$0.085 per Hour
c5.xlarge	4	17	8 GiB	EBS Only	\$0.17 per Hour
c5.2xlarge	8	34	16 GiB	EBS Only	\$0.34 per Hour
c5.4xlarge	16	68	32 GiB	EBS Only	\$0.68 per Hour
c5.9xlarge	36	141	72 GiB	EBS Only	\$1.53 per Hour
c5.18xlarge	72	281	144 GiB	EBS Only	\$3.06 per Hour
c5d.large	2	9	4 GiB	1 x 50 NVMe SSD	\$0.096 per Hour

Рисунок 3. – Фрагмент списка

Эта величина (ECU) может быть агрегирована по приложениям или/и по большим наборам серверов (кластеров) в общую Используемую Вычислительную Мощность (Compute Capacity Utilization – CCU), которая имеет естественный (в %) способ проверки эффективности: чем ближе к 100% - тем лучше.

Расчет Используемой Вычислительной Мощности - ССА

ССА (Доступная вычислительная мощность) – это общая (сумма) всех закупленных в облаке ECU (Эластичных Вычислительных Единиц) для конкретного приложения или кластера серверов. Это рассчитывается как количество мощности, купленной и доступной для использования:

$$ССА = \sum ECU_i$$

Например, доступная вычислительная мощность серверов типа m5.xlarge и типа c5.4xlarge будет  $16 + 68 = 84$  ECU

### Расчет Использованной Вычислительной Мощности – CCU

Это сколько вычислительной мощности было использовано за определенный период времени:

$$CCU = \sum (ECU_i * CPU_i\% / 100)$$

Где CPU<sub>i</sub>% – это загрузка процессора «i-го» сервера (EC2), которую можно измерить с помощью мониторинга AWS CloudWatch или любых других инструментов, таких как, например, DataDog.

Наконец, использованную вычислительную мощность приложения или кластера в процентах нужно рассчитывать, как:

$$CCU_t \% = (CCU / CCA) * 100\%$$

ССА против CCU можно использовать для сравнения размера и эффективности использования облака двумя (или более) приложениями (или линиями бизнеса). Ниже приведен пример сравнения эффективности двух приложений, который показывает, что приложение APP\_1 имеет гораздо больше возможностей для сокращения используемой мощности и, соответственно, для большей экономии затрат.

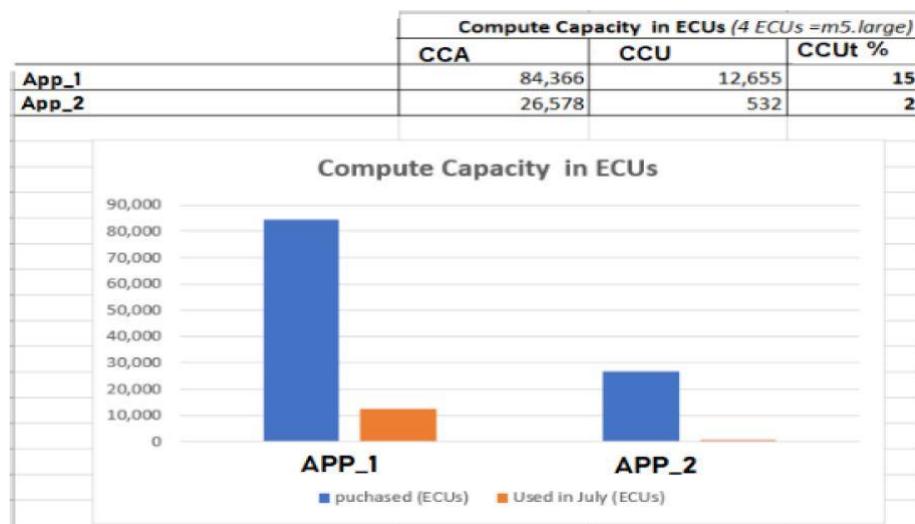


Рисунок 4. – Сравнение двух приложений

Оперативная память, дисковые операции ввод/вывод и использование емкости сети

Для рабочих нагрузок с интенсивным использованием памяти или операций ввода-вывода настройка правильного размера сервера только на основе оптимизации вычислительных мощностей не может быть выполнено правильно. Соответственно аналогичный расчет должен быть сделан для остальных подсистем используя их специфические метрики использования:

- **Используемая емкость оперативной памяти (RAM)** как сумма всех (общих) размеров RAM против процента ее использования.

- **Используемая полоса пропускания дискового ввода-вывода** в виде суммы операций ввода-вывода (в секунду) умноженной на размер операции в Кб против процента ее использования.

- **Используемая полоса пропускания сетевой карты** как сумма фактической полосы пропускания (Gbit / s) против процента ее использования.

Рассматривая все 4 измерения мощности сервера, можно увидеть, как работает предложенная настройка (текущее использование мощностей в сравнении со случаем, когда все рекомендации реализованы). В приведенном ниже примере показано, как использование всех 4 подсистем (измерений) улучшено для двух приложений:

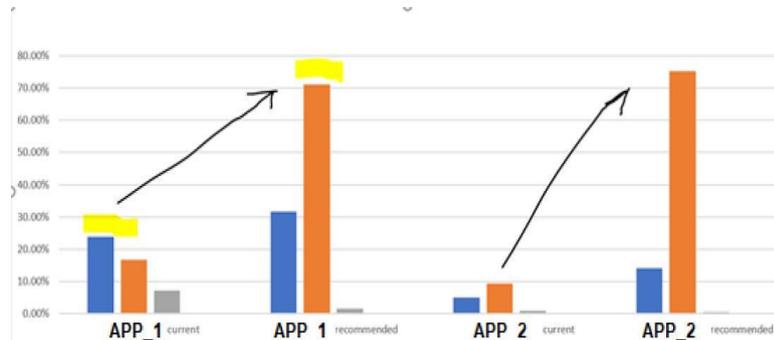


Рисунок 5. – Сравнение двух приложений по измерениям

Обратите внимание, что узкое место - наименее оптимизированная подсистема - может быть изменено или останется тем же в рекомендуемом размере (типе) сервера.

Для упрощения, предлагается использовать единую оценку использования серверов - максимум из 4-х описанных, называемый коэффициентом использования (OR - Operating Ratio).

Типичные паттерны (текущий OR против рекомендуемого)

И, наконец, запоминая (в базе данных) историю этих 4-х метрик в виде OR, можно построить тренды, и независимо от добавления дополнительной рабочей нагрузки (покупка новых серверов для обработки новой нагрузки), тренд должен быть плоским или повышаться до уровня насыщения 70% для хорошего оптимизированного облака, в то время как фактические затраты могут продолжать расти, но этот рост не был бы чрезмерным и просто отражал бы рост бизнеса!

На следующих нескольких графиках показаны типичные паттерны и антипаттерны:

**1. Далеко от оптимума – множество возможностей сокращения расходов:**

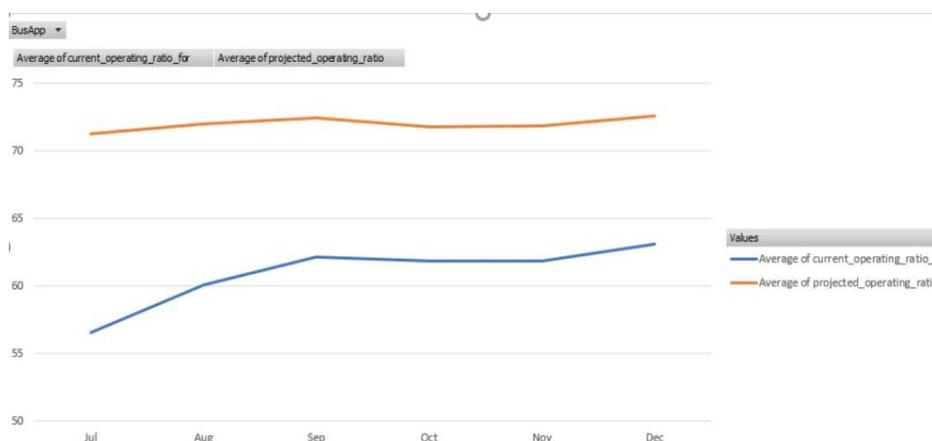


Рисунок 6. – Типичные и нетипичные паттерны

## 2. Оптимум достигнут!

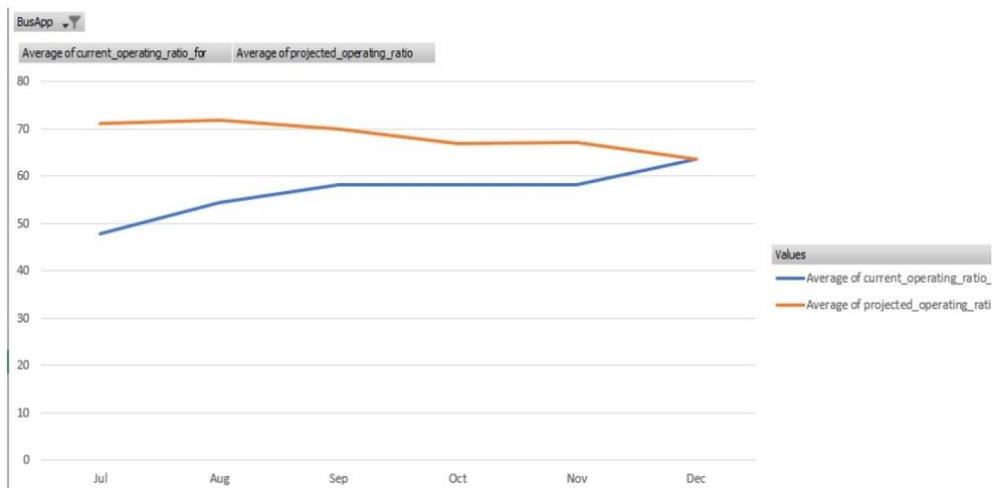


Рисунок 7. – Достижение оптимума

## 3. После оптимального использования происходит перегрузка:

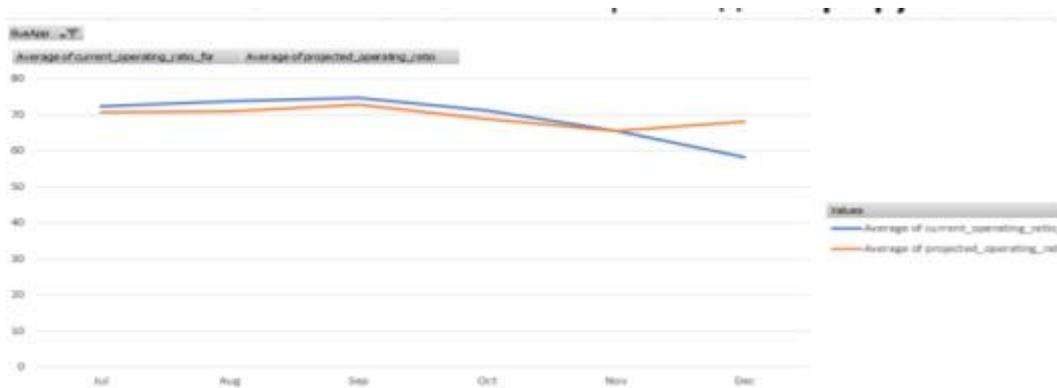


Рисунок 8. – Перегрузка

## 4. ПЛОХАЯ тенденция, эффективность снижается:

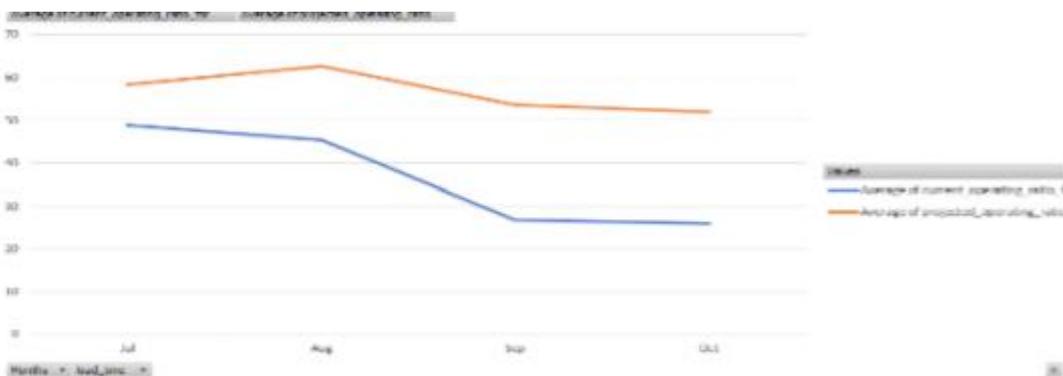


Рисунок 8. – Снижение эффективности

5. ОПАСНОСТЬ! Чрезмерная нагрузка. Может быть проблема дефицита мощностей серверов. Необходимы затраты на покупку дополнительных мощностей!

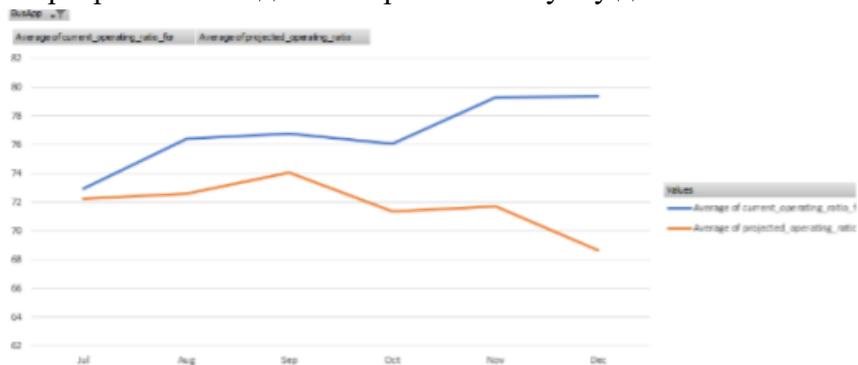
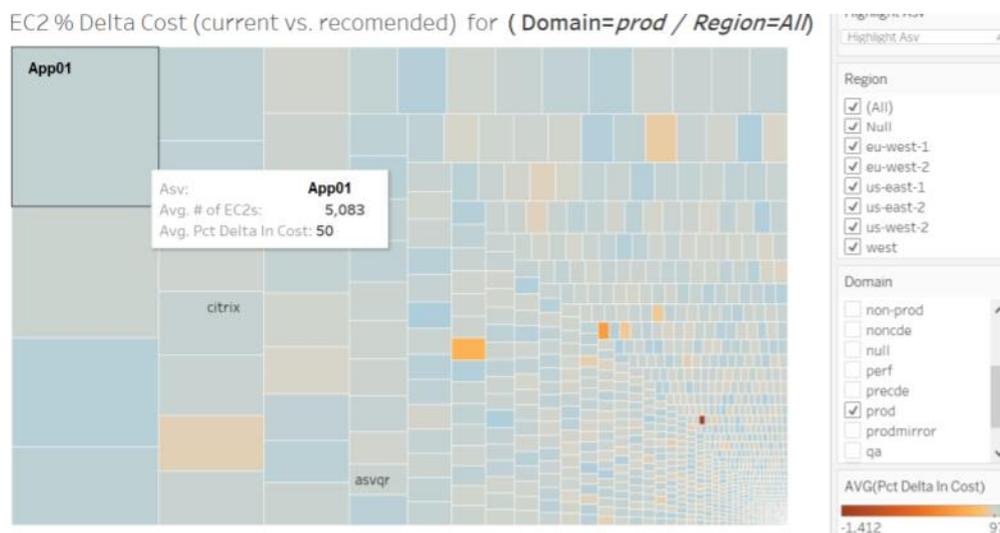


Рисунок 9. – Опасная ситуация

Карты общей эффективности затрат всей облачной инфраструктуры серверов Чтобы определить приложения, которые имеют наибольшие возможности экономии или наоборот (нуждаются в инвестировании), можно использовать древовидную карту (тепловую диаграмму), как показано на рисунке ниже (пример сделан с использованием Tableau):



EC2 % разницы стоимости серверов (текущей против рекомендованной)

Рисунок 10. – Древовидная карта

EC2 % разницы стоимости серверов (текущей против рекомендованной) Похожую карту можно строить и для OR. Достоинством предложенного подхода можно назвать избежание оперирования в абсолютных денежных единицах.

Дополнительные Ресурсы

1. Технический блог “System Management by Exception” [www.Trub.in](http://www.Trub.in)
2. Блог компании TTD <http://www.TruTechDev.com/>
3. ФБ Страница - [www.facebook.com/Perfomalist](http://www.facebook.com/Perfomalist)
4. Научно-технические статьи - [ww.researchgate.net/profile/Igor\\_Trubin](http://ww.researchgate.net/profile/Igor_Trubin)