UDK 004.75

CLOUD SERVERS USAGE EFFICIENCY AND COST OPTIMIZATION

I. A. Trubin
PhD, IT Manager, Capital One Bank, USA
Director, Computer Measurement Group, USA

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

I.A. Trubin

Igor Trubin has started in 1979 as an IBM/370 system engineer. In 1986 he got his PhD. in Robotics at St. Petersburg Technical University (Russia) and then worked as a professor teaching CAD/CAM, Robotics for about 12 years. He published 30 papers and made several presentations for conferences related to the Robotics and Artificial Intelligent fields. In 1999 he moved to the US and worked at Capital One bank as a Capacity Planner. His first CMG.org paper was written and presented in 2001. The next one, "Exception Detection System Based on MASF Technique," won a Best Paper award at CMG 2002 and was presented at UKCMG 2003 in Oxford, England. He made other tech. presentations at IBM z/Series Expo, Southern and Central Europe CMG and ran several workshops covering his original method of Anomaly and Change Point Detection (www.Performalist.com). He is an author of the online class "Performance Anomaly Detection" (https://cmg1.teachable.com). After working more than 2 years as the Capacity team lead for IBM, he had worked for SunTrust Bank for 3 years and then at IBM for 2+ years as Sr. IT Architect. Now he works for Capital One bank as IT Manager at the Cloud Engineering department and since 2015 he is a member of CMG.org Board of Directors. He runs his tech blog at www.Trub.in

Abstract. The public cloud has unlimited capacity if you have an unlimited budget to buy it. But the reality is that budgets are never truly unlimited, and one needs to do rightsizing of the cloud objects to stop wasting money on unused or unneeded cloud capacity.

During this session, Igor Trubin will discuss methods of tracking and reporting on cloud usage. He will discuss how cloud cost optimization can be done to show the efficiency of individual applications, lines of businesses (LOB), or the entire infrastructure. Lastly, participants will learn how to access and plan for organic cloud growth and cost increases.

Keywords: cloud, servers.

Problem statement

–The public cloud has unlimited capacity if you have unlimited budget to buy it. But if the budget is tight, one needs to do Rightsizing of the cloud objects to stop wasting money on capacity that are not used.

–TOOLS to analyze cost and get Rightsizing recommendations are:

–*AWS (Amazon Cloud)*

– Amazon EC2 resource optimization recommendations/ AWS Cost Management

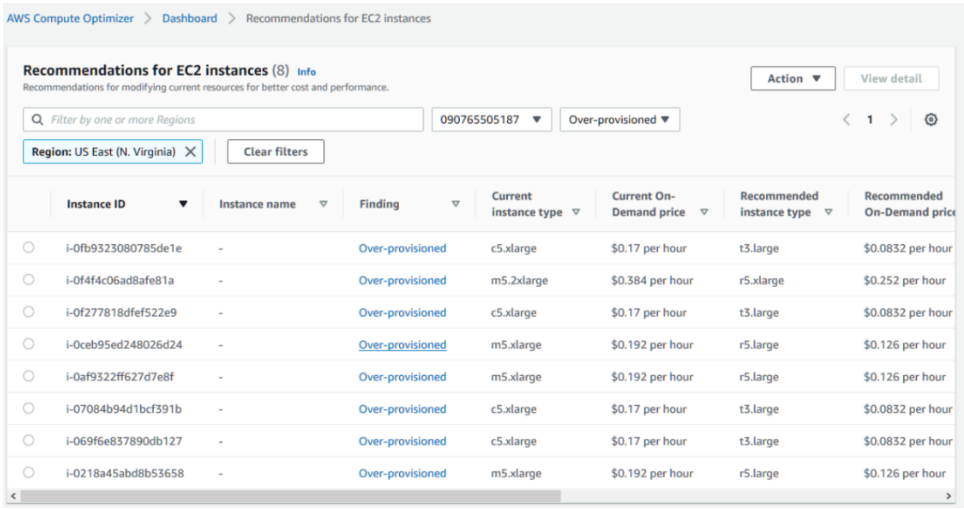

Figure *1.* – Recommendations for EC2 instances

EC2 sizing recommendation example from *AWS Compute Optimizer:*
–*Trusted Advisor;*
–*VMWARE CloudHealth;*
–*Rightsizing;*
–*Cost management;*

Solution to solve the problem

For growing business cost management tools most of the time show growing expenses regardless of rightsizing efforts. The typical trend is below:

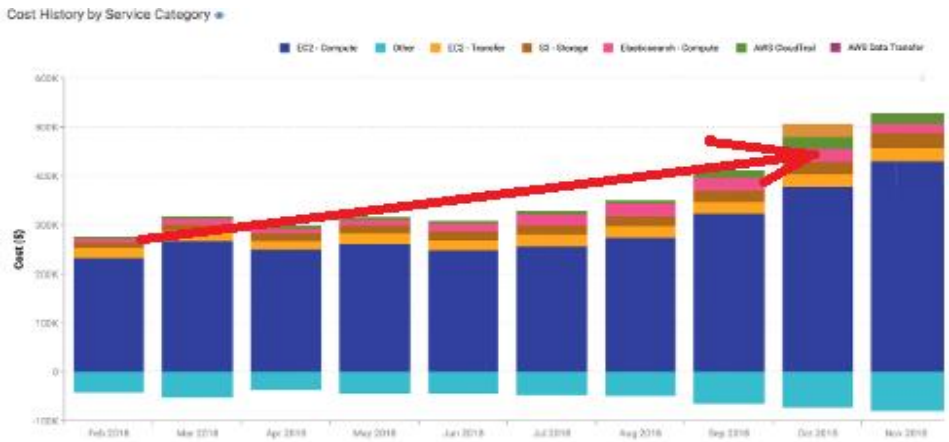

Figure *2.* – The typical trend

To show how effective the cloud is used another following normalized approach is suggested**.** Multidimensional capacity utilization report should cover:

1. Compute capacity utilization
2. Memory (RAM) capacity utilization
3. Disk I/O bandwidth utilization
4. Network bandwidth utilization

To show how effective the Compute capacity is used there is a need to do normalization and aggregation of all different virtual servers (EC2) types (sizes).

One of the approach is to use AWS Elastic Compute Units (ECUs) That is the comparable “horse power” metric and can be get from AWS EC2 price list (the fragment of the list is below):

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

Figure *3.* – The fragment of the price list

It could be aggregated by Applications or/and by LOBs into Compute Capacity Utilization (CCU) that has a natural (as %) way to check a progress: closer to 100% is better. For example, combined compute power of *m5.xlarge and c5.4xlarge* would be 16+68=84 ECUs

CCUt - Compute Capacity utilization

CCA (Compute Capacity Available) is overall (sum) of all "i" ECUs for particular application or LOB. That gives the capacity amount purchased and available:

$CCA = \sum ECU_i$

CCU (Compute Capacity Used) is how much compute capacity used:

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

Where $CPU_i\%$ is CPU utilization of "*i*" server (EC2), that could be get from the AWS CloudWatch tool or any other performance tools like DataDog. Finally Compute Capacity Utilization should be calculated as

CCUt % = (CCU/ CCA)*100%

CCA vs. CCU can be used to compare size and efficiency of cloud usage of two (or more) applications (or LOBs). Below is the example of two applications efficiency comparison, which shows that application APP_1 has much more opportunity to be downsized and respectively to get bigger cost savings.

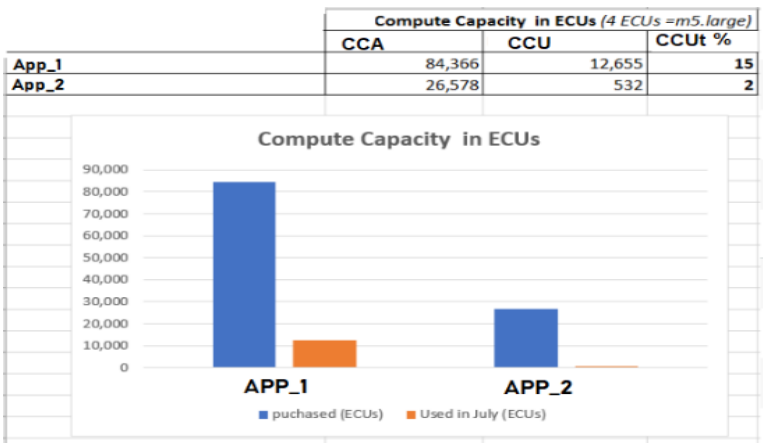

Figure *4.* – Compute Capacity in ECU

RAM, Disk I/Os and Network Capacity Utilization

For workloads that are memory or I/Os intensive the downsizing only based of Compute capacity optimization could not be done correctly. Respectively similar calculation needs to be done to get the following:

–*Memory (RAM) capacity utilization* as a sum of all (total) RAM sizes in Gb available vs. RAM used in Gb.

–*Disk I/O bandwidth utilization* as a sum of IOs (per sec.) *IO_size (KB) vs. sum of IO_bandwidths (KB per sec)

–*Network bandwidth utilization* as a sum of actual bandwidth (Gbit per sec) vs. Gbit used (per sec)

Considering all 4 dimensions of capacity usage allows to see how rightsizing works (current capacity usage vs. case when all recommendations are implemented), the example below shows how the capacity usage of all 4 dimensions are improved for two applications:

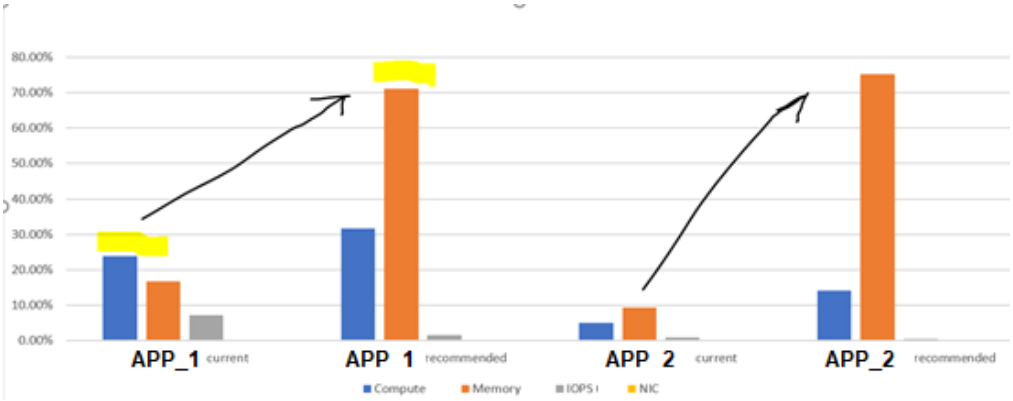

Figure 5. – The capacity usage of all 4 dimensions

Note, the culprit (bottleneck – least optimized dimention) could be changed or stay the same in the recommended right size.

Typical patterns (Current vs. Recommended)

And finally by keeping history of those 4 capacity usage metrics the trend could be built and regardless of adding additional workload that trend should be flat or going up to saturation level of 70% for good optimized cloud , while the actual cost still might keep growing, but that growth would be not excessive and just reflecting the growth of the business!

On the next few tend charts the typical pattern and antipatterns are shown:

– Far from Optimum – a lot of savings opportunities:

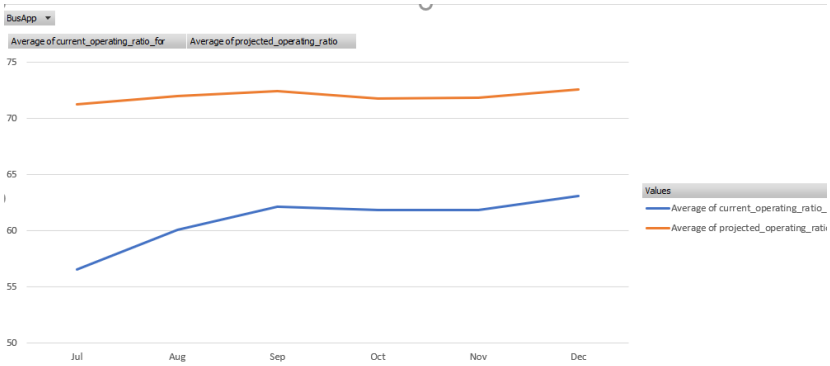

Figure 6. – A lot of savings opportunities

– Optimum is achieved!

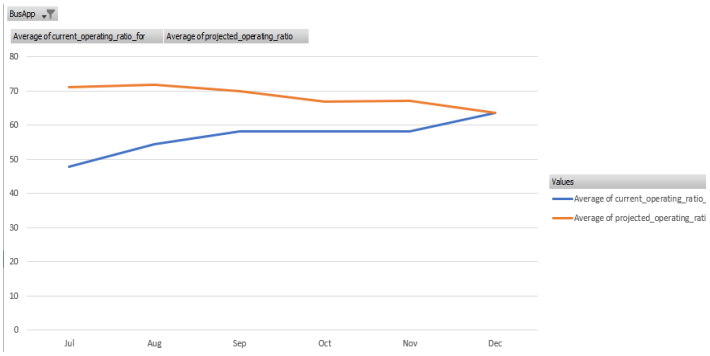

Figure *7*. – Optimum

– After optimum there is a bit of overutilized:

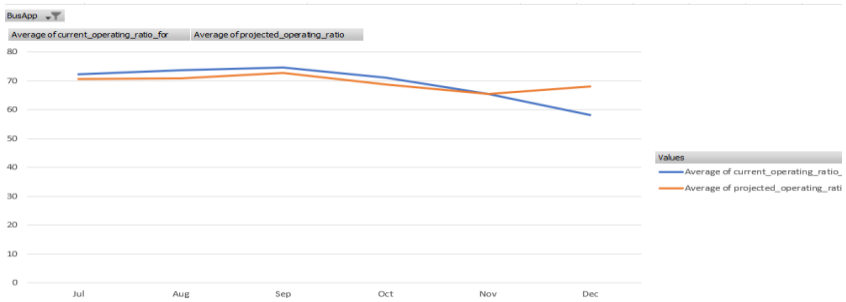

Figure *8*. – After optimum

– BAD trend, efficiency is declining:

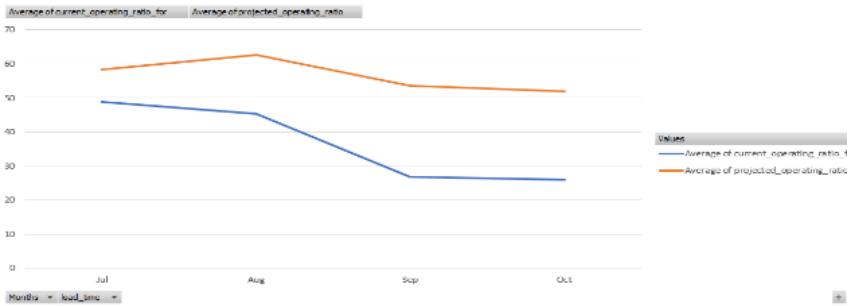

Figure *9*. – BAD trend

– DANGER! Overutilization. Could be a capacity issue. Buy more capacity!

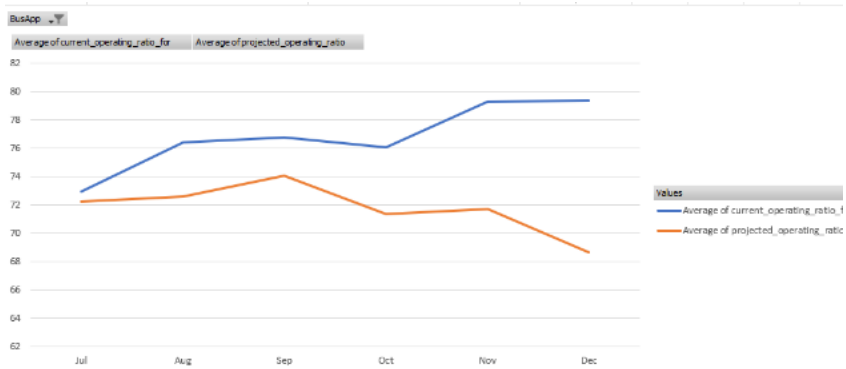

Figure *10*. – Overutilization

EC2 Overall Fleet Cost efficiency Treemap

To identify applications that have largest cost savings opportunities or opposite (– need some investing), the treemap (heat-chart) could be used as shown on the picture below:

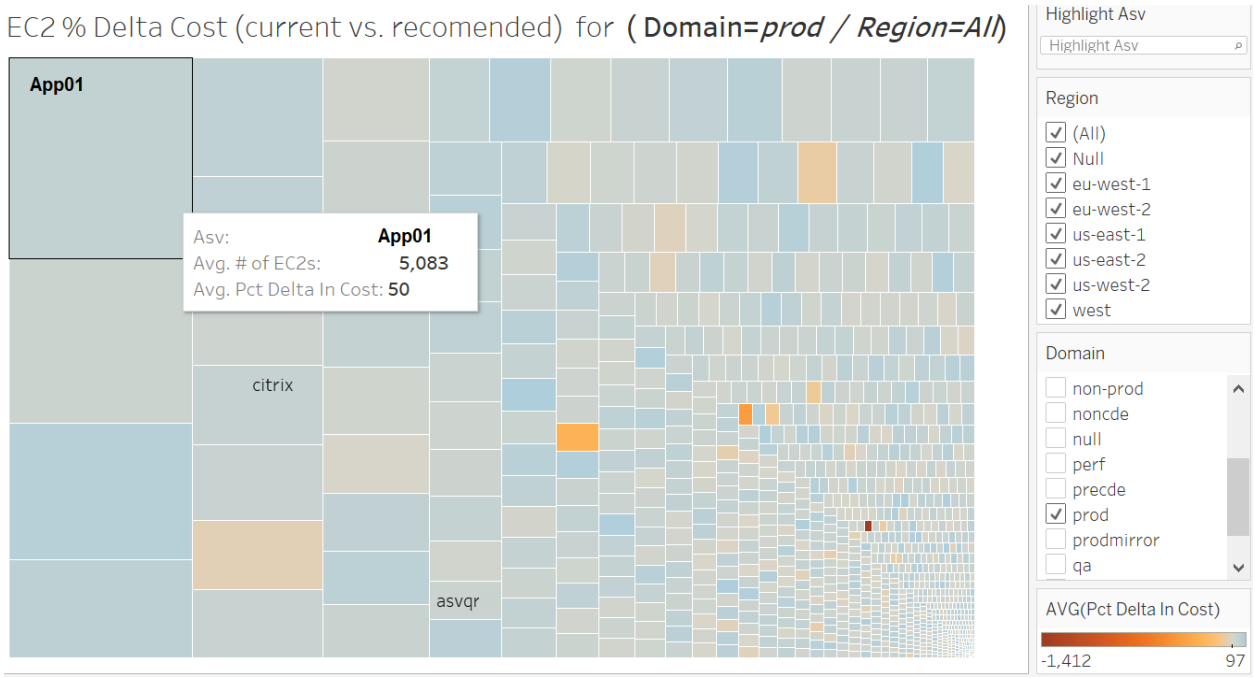

Figure *11*. –EC2 % Delta Cost