Is Your Capacity Available?

Igor Trubin, PhD
Capital One Bank, IT Manager

ABSTRACT

- *Capacity Management and Availability Management are two interconnected services. This connection is getting more important in the current era of virtualization, clustering, and especially cloud computing.*
- *It is obvious that IT customers want not only sufficient capacity for their applications, but even more importantly they want this capacity to be highly available.*
- *This paper shows how to incorporate availability requirements to satisfy this need into the current classical capacity planning.*

Introduction

- O’Reilly, “*Site Reliability Engineering. HOW GOOGLE RUNS PRODUCTION SYSTEMS*” example:
  - BEFORE: “*I want 50 cores in clusters X, Y, and Z for service **Foo**.*”
  - NOW: “*I want to run service **Foo** at 5 nines of reliability* “.
- This paper’s author has already worked on a similar effort to make availability modeling an essential part of capacity management and published on his blog:
  - ”Is your Capacity Available? - A topic for CMG Conference Paper” - “*System Management by Exception” technical blog. http://www.**Trub.in**/2013/05/is-your-capacity-available-topic-for.html*.

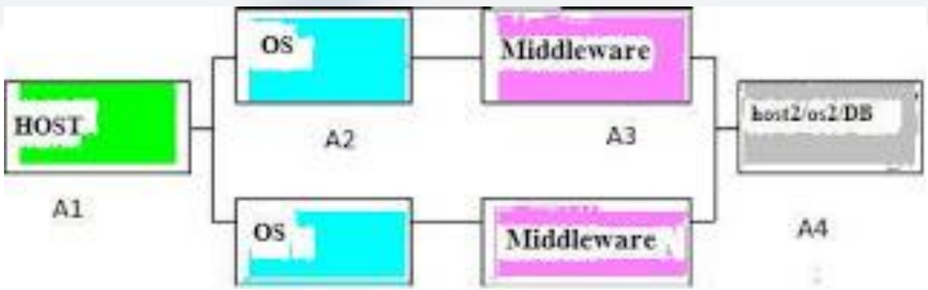
System Reliability and Availability Basics

Variable	Definition	Examples (usually in hours)	
MTBF	Mean Time Between Failures	20000	50000
MTTR	Mean Time To Repair	2	5
Availability	MTBF / (MTBF + MTTR)	0.9999000	0.9999000
Downtime per year	(1 - Availability) x 1 year	52m33s	52m33s
Series MTBF	$1 / (1/MTBF_1 + .. + 1/MTBF_n)$	14285.714	
Series Availability	$A_1 \times A_2 \times .. \times A_n$	0.9998000	
Series Downtime	(1 - Series Availability) x 1 year	1h45m6s	
Parallel MTBF	MTBF x (1 + 1/2 + ... + 1/n)	55714.285	
Parallel Availability	$1 - (1-A)^n$	0.9999999	
Parallel Downtime	(1- Parallel Availability) x 1 year	0.315297s	

The general cluster availability formula:

$$A = 1 - \frac{n!}{(s+1)!(n-s-1)!}(1-a)^{s+1}$$

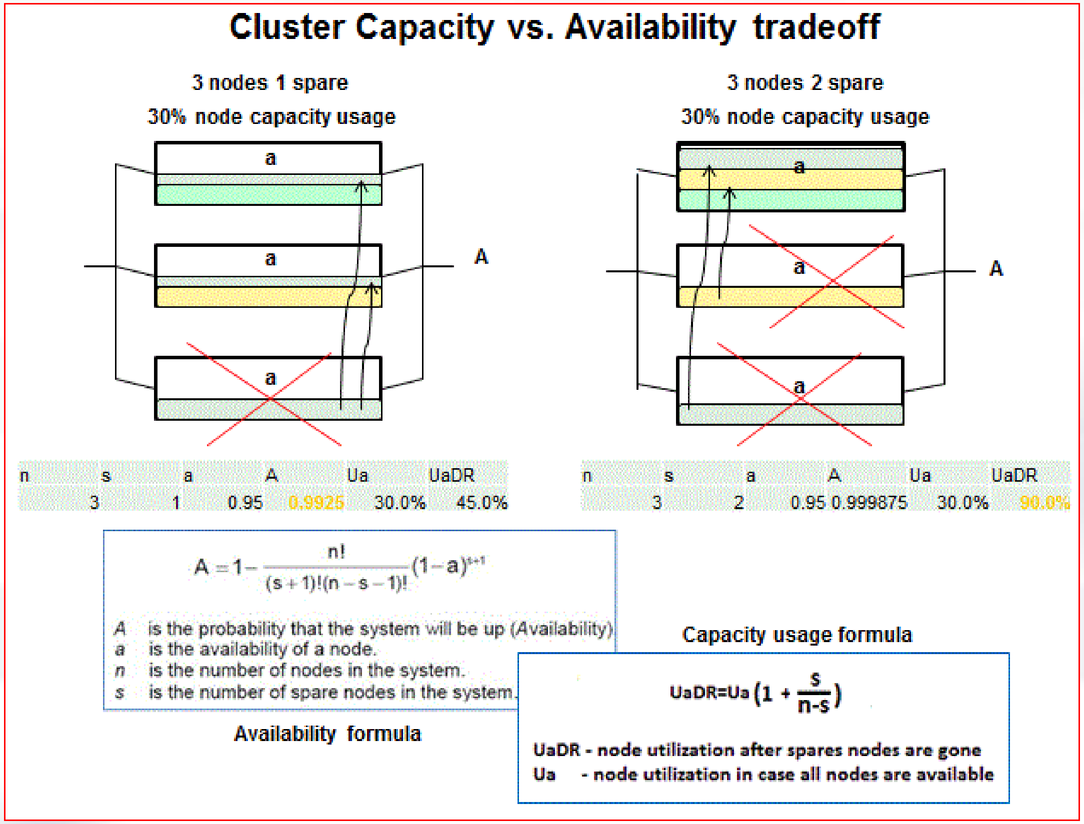
A is the probability that the system will be up (*A*vailability).
a is the availability of a node.
n is the number of nodes in the system.
s is the number of spare nodes in the system.



$$A = A1*(1-(1-(A2*A3)^{n}) *A4$$

- This approach opens up the possibility to quantitatively justify architectural decisions (not just using "best practices" or "gut feelings")

Capacity vs. Availability

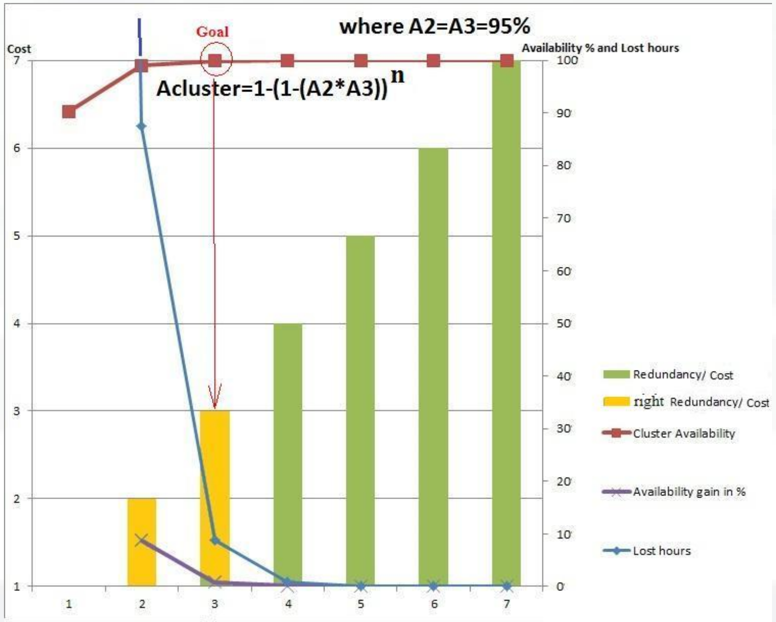


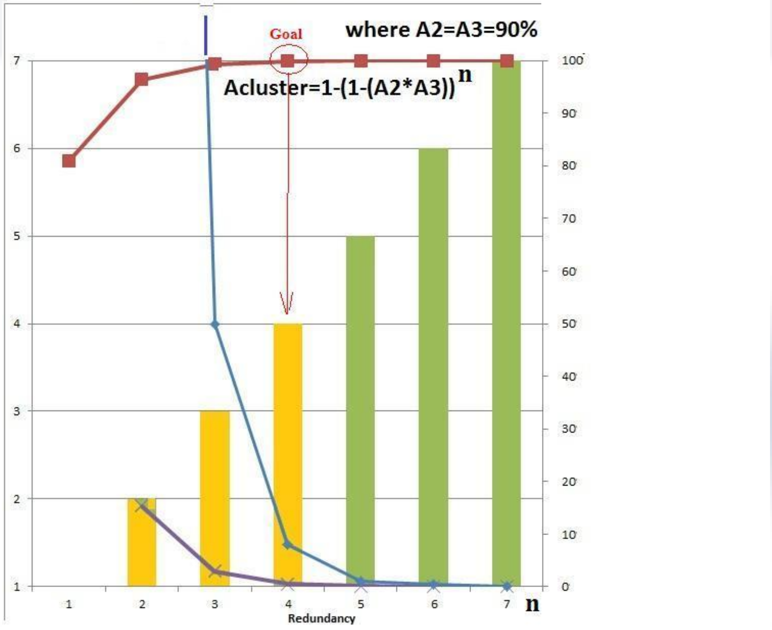
That can be a way to save money by allocating less capacity with the same number, but more reliable nodes.

- If **MTTR** for each individual component (software and hardware) is known, the whole infrastructure availability can be estimated using this approach. But how to obtain the individual MTTR?
  - From Monitoring (e.g., synthetic-robotic)
  - From Incident Management

The Right Number of Cluster Redundancy to Achieve Availability Goal

- The following two charts illustrate how the same availability goal can be achieved by different numbers of redundancy.
- This is all possible if the less redundant configuration has more available individual components.





RULE of NINES

If the component availability has one **9** (90.00%), then increasing the redundancy by +1 gives the additional **9**'s to the cluster availability

$$1-(1-A)^{n} = 0.\overbrace{99\ldots9}^{n}$$

Solution is **A = 0.9**

for any integer **n** within the interval **(0,** ∞**).**

Based on this "Rule of 9s", each additional node adds one more 9 to overall cluster availability. This is exactly true only if the single node has only one **9** (A = 0.9), as shown in the above equation.

But how would this work for other single node availability numbers? What if it has two or three **9**'s?

The cluster availability number of 9's will be increasing in arithmetic progression:

$$1-(1-0.\overbrace{99\ldots9}^{m})^{n} = 0.\overbrace{99\ldots9}^{m \cdot n}$$

Questions?

Thank you!

Igor Trubin

www.Trub.in