DON'T LEAVE PATCH MANAGEMENT TO CHANCE

Ralph M. DeFrangesco
1505 Breckenridge Place
Harleysville, PA. USA 19438
rdefrangesco@hotmail.com

ABSTRACT

Patch management is a costly but necessary part of server management. Should more time be spent on patch evaluation and analysis or testing and implementation? This paper attempts to introduce conditional probability into the patch management process giving the reader a tool to evaluate where best to spend their time, analysis or testing. Bayes' theorem will be used to help predict the outcome of a patch management example.

INTRODUCTION

Today, servers host mission critical applications that are highly reliable and available 24 hours a day, 7 days a week (Treese, 2002). Customers usually establish server uptime in a Service Level Agreement (SLA). Server uptime can vary from 99%, for not very critical systems, to 99.999% for extremely critical systems (Forsythe, 2005). An SLA specifying a server to be up 99.999% of the time equates to 5 minutes of downtime a year (Treese, 2002). It is very challenging to apply patches to a server with this type of SLA. In an environment where patches get released weekly, how can we expect to analyze and test them thoroughly enough so there is no impact to production? One solution is to use a hot standby for failover purposes and to apply the patches to. This is an expensive option since the server sits around waiting to be used (Treese, 2002).
Systems Engineers and Administrators need a tool that can help predict the outcome of patch analysis, testing and installation. With time frames shrinking to analyze, test and install patches, how much time should we spend on each activity? Systems Administrators are caught between a Scylla and Charybdis, a rock and a hard spot, because management wants it all, but with minimal impact and cost.
This paper will interject the use of conditional probability theory, specifically Bayes' theorem as a tool to help predict the outcome of a patch management example. Conditional probability allows us to specify the degree of belief in some proposition based on the assumption that some other propositions are true. This paper takes a top down approach meaning we look at the probability an event will occur given a prior probability.

BAYES THEOREM

Bayes' theorem basically involves calculating a new event on the basis of earlier estimates which come from empirical data (Bayes, 2002). We can explain Bayes' theorem with the following equation:

$$p(A|X) = \frac{p(X|A)*p(A)}{p(X|A)*p(A) + p(X|\sim A)*p(\sim A)}$$

In a key feature of Bayes' theorem, we combine our initial probability with new sample data and calculate a revised probability distribution. This new distribution now becomes the prior probability for a new sample. The intent is that we make better decisions based on empirical data.

APPLICATION OF BAYES THEOREM

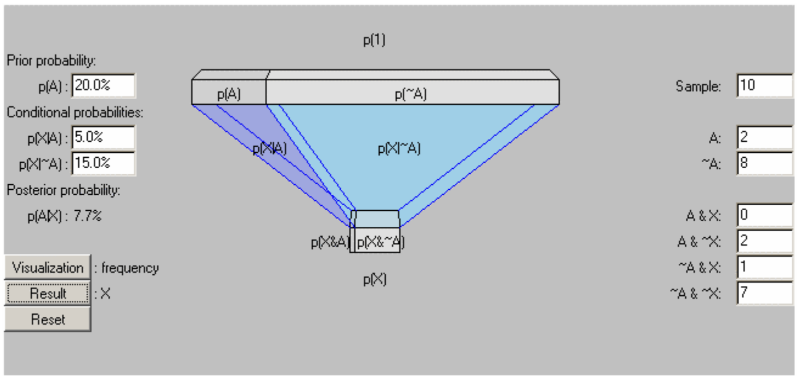
Let’s look at an example. Servers crash for a multitude of reasons. In our example, experience has shown us that when a server does crash, it’s due to System Administrator error twenty percent of the time and usually can be tied back to an incomplete or untested patch bundle. If a patch bundle is incomplete, there is a five percent probability that it was due to a Systems Administrator not installing a pre or post requirement. On a similar note, if the bundle was incomplete, there is a fifteen percent probability that it was not tested sufficiently enough. What we need to know is, what is the probability that a bundle is complete and tested but caused a server to crash?
For illustrative purposes we decided that our server has crashed ten times during the course of a year and that we spent fifty hours on patch evaluation and analysis. Figure 1 uses the BayesApplet (Yudkowsky, Rovner, 2004) to calculate the posterior probability.
Let p(A) our prior probability = the percent time that a server crashes due to Administrator error, .20%.
Let p(X|A), our first conditional probability, = the percent of the time a patch bundle is incomplete, .05%.
Let p(X~A), our second conditional probability, = the patch bundle was not tested sufficiently, .15%.
Using Bayes’ theorem, our equation is given by

$$p(A|X) = \frac{p(X|A)*p(A)}{p(X|A)*p(A) + p(X|{\sim}A)*p({\sim}A)}$$

in so that p (A|X) is the posterior probability of event X given that A is known is equal to the prior probability of X given A (.05), multiplied by the probability of A (.2) divided by the probability of X given A (.05) multiplied by the probability of A (.2) plus the probability of X given (not) A (.15) multiplied by the probability of (not) A (.8).

$$.077 = \frac{(.05 * .20)}{(.05 * 20) + (.15 * .80)}$$

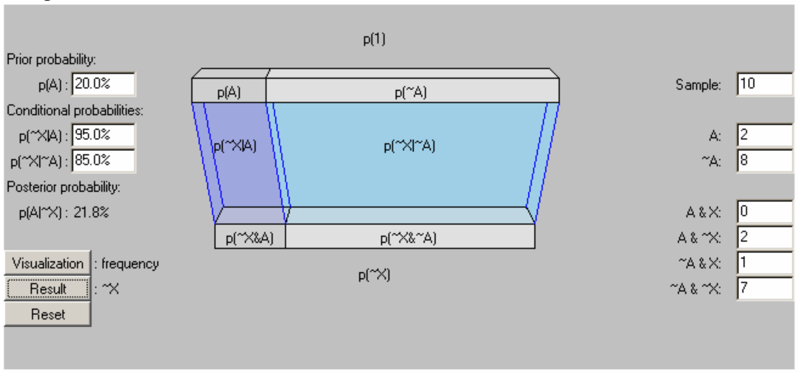
A & X – Complete (analyzed) and tested
A & ~X – Complete (analyzed) and not tested
~A & X – Not complete and tested
~A & ~X – Not complete and not tested


Prior probability:
p(A) : 20.0%
Conditional probabilities:
p(X|A) : 5.0%
p(X|~A) : 15.0%
Posterior probability:
p(A|X) : 7.7%
Visualization : frequency
Result : X
Reset

Sample: 10
A: 2
~A: 8
A & X: 0
A & ~X: 2
~A & X: 1
~A & ~X: 7

Figure 1

What we can conclude from figure 1 is that with prior and conditional probabilities as defined in our example, our posterior probability calculates to be 7.7%. We interpret this as meaning that although a patch bundle was complete (analyzed) and tested by a Systems Administrator that there is still almost an eight percent chance that a server could crash. In our example, it shows up as zero because of rounding (A & X).
In figure 2, we are trying to show what the posterior probability would be if a patch bundle was complete but not tested.


Prior probability:
p(A) : 20.0%
Conditional probabilities:
p(~X|A) : 95.0%
p(~X|~A) : 85.0%
Posterior probability:
p(A|~X) : 21.8%
Visualization : frequency
Result : ~X
Reset

Sample: 10
A: 2
~A: 8
A & X: 0
A & ~X: 2
~A & X: 1
~A & ~X: 7

Figure 2

Again using the BayesApplet, we calculate the posterior probability to be 21.8%. This tells us that if a patch bundle is complete but not tested, there is a twenty-two percent chance that it will cause the server to crash. The Applet rounds the probability to twenty percent (A & ~X). In analyzing both examples, we would be better off spending less time evaluating and more time on testing (~A & X) .10.
Now that we have both posterior probabilities, the next time we run the model, we must update our prior probabilities with our posterior probabilities because the prior probability was our initial belief. We added new evidence, conditional probabilities, and now we have a final belief.

Both of these examples can be expanded to see if there is any benefit to increasing the time we spend evaluating and analyzing versus testing and implementing. Lets say, using the above as an example, that with fifty hours of evaluating and analyzing we end up with a posterior probability of 7.7% as in figure 1. We decide to increase our evaluation and analysis time to one-hundred hours and only reduce the probability to 6.7%. Would it be worth the additional time to decrease the posterior probability by one percent? Probably not since a two fold increase in hours only yields a one percent decrease in the probability that a server might crash.

CONCLUSION

We can conclude that patching servers is a risk intense activity. Systems Administrators do not have a cadre of tools to predict the outcome of applying patches to critical servers, but yet are expected to keep their servers up and running without impact.

This paper introduced the concept of applying probability theory to the task of analyzing patches and predicting patching outcomes. In our example, we saw that there may be only incremental benefits to spending more time analyzing rather then testing and implementing. We used this example to demonstrate how to apply conditional probability and the benefits to using this type of theory in patch management. We were able to demonstrate that even though we spent fifty hours analyzing patches, indicating the bundle was complete, we still crashed the server twenty percent of the time. From this, we can conclude that our time would be better spent testing rather then analyzing because our server crash rate is only ten percent.

In a paper by Kwiatowska (2003), he described probability theory as the following,

> *"Probability is also used to quantify unreliable or unpredictable behaviour, for example in fault tolerant systems, communication protocols and computer networks, where properties such as component failure and packet loss can be described probabilistically."*

FUTURE WORK

This work can be expanded to break out the difference between the amount of crashes caused by COTS versus in-house developed software, the probability of backup failures and hardware or operating system failures.

The Markov process is very similar to Bayes' theorem in that it looks at a number of events and tries to guess the likelihood of the event based on current data. The output will look similar in nature to the original input (Deshpande, Karypis, 2004).

REFERENCES

Bayes. (2002. Bayes original Essay. Retrieved on June 7, 2005 from http://www.stat.ucla.edu/history/essay.pdf

Deshpande. M, Karypis. G. (2005). "Selective Markov Models for Predicting Web page Accesses", **ACM Transactions on Internet Technology**, Vol. 4, No. 2, May 2004, pp. 163-184

Forsythe, (2005). Five 9's, Retrieved on June 7, 2005 from http://www.forsythe.com/Forsythe/infraservices/servers/server_five9s.jsp

Kwiatkowska. M. (2003). **Proceedings of the 18th Annual IEEE Symposium on Logic in**

Computer Science (LICS'03)

Treese. W. (2002). Building big, reliable web sites, (Part 1). **Net Worker,** Vol. 6, Issue 1, pp. 15-18 March 2002

Yudkowsky.E., Rovner.C., An Intuitive Explanation of Bayesian Reasoning, Retrieved June 6, 2005 from http://yudkowsky.net/bayes/bayes.html