The Challenge of Risk Assessment for

Complex Information Systems

Robert Bonneau, DoD, osd.pentagon.ousd-re.mbx.communications@mail.mil

Abstract

Increasingly, we rely on complex information systems such as cloud and large data analytics driven by methods in artificial intelligence for many aspects of our daily lives. Most of these applications are highly software and infrastructure dependent, but through virtualized cloud environments, much of the safety and test and evaluation of these systems is not available to the typical user. Moreover, our ability to assess risk of not only the applications running on these infrastructures, but, also, the infrastructures themselves requires a definition and framework for risk assessment. We will detail some of the leading challenges for risk assessment in some of our current and future information service architectures and some paradigms to address these challenges as a step toward automated resilient systems.

I. Introduction

Our objective with this paper is to discuss the methodology for assessment of the performance and state of information systems for multiple system components such as network, computing, and software states. Many new system tools such as DevOps methodologies allow the instrumentation and measurement of cloud infrastructures for assessment. The possibility of automating system assessment with these methods can be posed in a Bayesian estimation framework given the dynamic nature of the statistical distributional model of a system. The challenge of such an approach is that the complexity of the system distributional model often is more complex than an analytic Bayesian approach may permit. We show a methodology for simplifying the model of a complex system and a means of bounding the Bayesian risk for such a system, thereby bounding the risk of system errors and failure.

II. Measurement

Figure 1 shows different types of system properties for measurement. Typically, a DevOps architecture will measure the state of hardware, software, and network in an ontology associated with a graph based format of metadata. We thus characterize the metadata associated with a given system from a set of measurements X which consists of a set of signal $S$and noise E vectors which can be sampled through a measurement function Φ_Λ. Equation 1 shows our signal vector.

We have some signal environment S and interference environment E and we wish to measure these with some measurement vector Φ_Λ

$$X = \Phi_\Lambda \otimes (\mathrm{S} + \mathrm{E}) \tag{1}$$

The problem in most measurement scenarios is getting the right Φ_Λ to separate S from E [1,2,3,4]. We define our measurement process such that Φ_Λ consists of the individual waveform projections on the environment defined by $\phi_{i,i\in 1,\ldots,N}$.

Generally speaking, the greater number of sampling vectors $\phi_{i,i\in 1,\ldots,N}$ for the metadata from our DevOps process, the lower the Bayes risk which will be discussed in section IV since with greater sample size, the more accurate our data sample of system behavior will be. The sample size for our model distribution in hypothesis testing is also affected by the

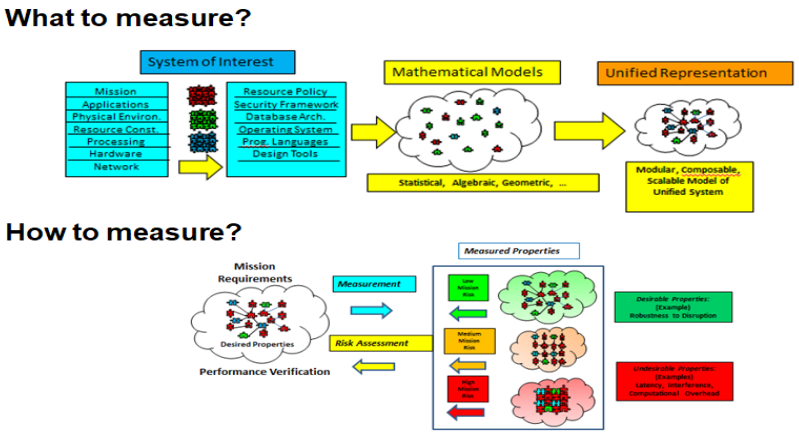

Figure 1) System Measurement

III. Modeling

Figure 2 shows un-validated vs. validated models. Un-validated models are can be significantly different than models derived from data experienced in an actual system. In a distributional context we will illustrate such a concept in terms of the mean vector, m and Covariance matrix of a normally distributed model, C_{xx}. Thus our mean and covariance matrices for our measured data from section II are indicated in equation 2.

$$m = E(X)\ ,\ C_{XX} = E((X - \mathrm{m})(X - \mathrm{m})) \tag{2}$$

Our approximated or un-validated covariance is indicated by equation 3.

$$\tilde{C}_{XX} = E\left\{\begin{bmatrix} \tilde{c}_{11} & \tilde{c}_{12} & \cdots & \tilde{c}_{1N} \\ \tilde{c}_{21} & \tilde{c}_{22} & \cdots & \tilde{c}_{2N} \\ \tilde{c}_{31} & \tilde{c}_{32} & \cdots & \tilde{c}_{NN} \end{bmatrix}\right\} \tag{3}$$

The difference between our validated and unvalidated model. with validated eigenvalues λ_m and un-validated eigenvalues $\tilde{c}_{mm}$,can be computed with the following Frobenius Norm or:

$$\left\|C_{XX} - \tilde{C}_{XX}\right\|_2^2 = \sum_{m=1}^{N} |\lambda_m - \tilde{c}_{mm}|^2 + \|C_{XX}\|_2{}^2 - \sum_{m=1}^{N} |\tilde{c}_{mm}|^2 \tag{4}$$

Once we have a distributional model we can use these models to form a likelihood function for testing whether our measured data conforms to our modeled distribution with equation 5

$$L(x) = \log[\ \exp(1/2[(x - m_1)^T \boldsymbol{C_{xx1}}^{-1}(\boldsymbol{x - m_1}) - (x - m_0)^T \boldsymbol{C_{xx0}}^{-1}(\boldsymbol{x - m_0})])] \tag{5}$$

From our log likelihood d is defined as [5]

$$d^2 = (m_1 - m_0)^T \boldsymbol{C_{xx1}}^{-1}(m_1 - m_0) \tag{6}$$

From this, we can configure our model for hypothesis testing. For the purposes of our system model this hypothesis testing determines whether our information system is within the specification of desired behavior H_0 or not H_1 as is shown in Figure 3. The distribution of each of these hypotheses is as follows.

$$H_0\ L(x)\ N(\frac{-d^2}{2}, d^2) \tag{7}$$
$$H_1\ L(x)\ N(\frac{d^2}{2}, d^2)$$

We can use this strategy to employ a hard threshold rule for detection such that

$$\mathrm{A}(x) = \begin{cases} 1\ L(x) \geq \eta \\ 0\ L(x) < \eta \end{cases} \tag{8}$$

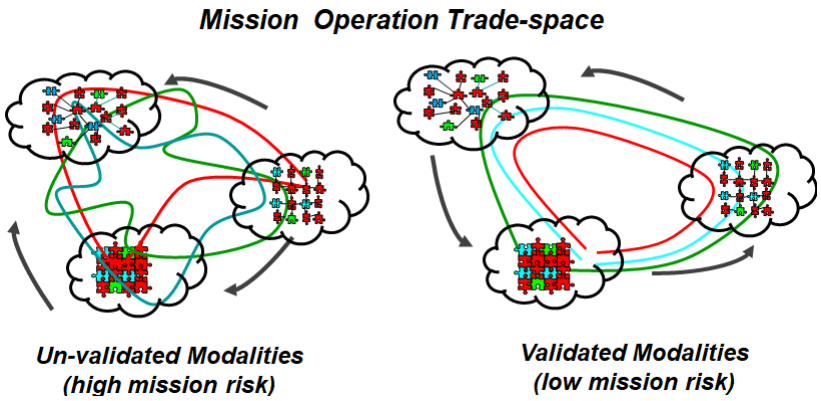

Figure 2) System Modelling

IV. Management

In order to automate our system lifecycle from code deployment to system monitoring, we need an efficient real-time methodology to evaluation system performance as is shown in Figure 3. Such performance is easiest with a Bayes risk assessment method that has an analytic expression for Bayes risk. To achieve a Bayes analytic risk function, we simplify the covariance model for our distribution using an L2 norm. Using the L2 norm gives us the largest eigenvalue of the Covariance matrix we have

$$\|C_{XX}\|_2 = {\sigma^2}_{max} \tag{9}$$

Our new distributional metric becomes:

$$\hat{d}^2 = (m_1 - m_0)^T {\sigma^{-2}}_{max}(m_1 - m_0) \tag{10}$$

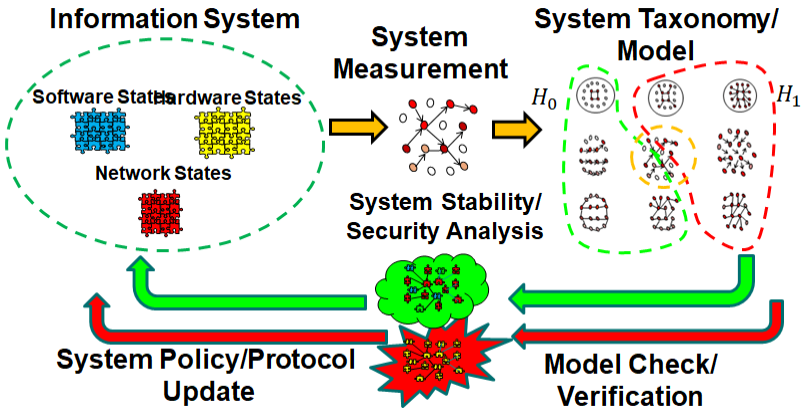

Figure 3) System Mangement

To compute our Bayes risk [5], we compute a false-alarm probability, withthe false alarm probability as

$$\alpha = 1 - Q(z) \tag{11}$$

where $Q(z) = \int_{-\infty}^{z} (2\pi)^{-1} exp\{-x^2/2\}dx$. Our miss probability is $1 - \beta$, where β

$$\beta = 1 - Q(z - \hat{d}) \tag{12}$$

We now can state that η is our threshold derived from our desire to minimize the maximum risk of the worst case distribution.

$$z = \frac{\eta + \frac{\hat{d}^2}{2}}{\hat{d}} \tag{13}$$

We now define our Bayes risk with

$$\Re(\alpha, \beta) = p_0 L_{01} \alpha + (1 - p_0) L_{10} (1 - \beta) \tag{14}$$

with L_{01} as our respective loss when H1 is decided and and L_{10} our loss when H0 is decided when H1 is true and p_0 the probability of H0.

V. Integrated Operation

Our goal for an integrated operation is a continuous deployedment starting with the ability to model and deploy a particular design on a distributed cloud architecture. We then use our Bayesian risk approach to manage the system through measured data from the deployed system. Our goal is to minimize our risk over the entire development and operations cycle of the system as is shown in Figure 4 and equation 15. We either can manage the system by changing system parameters by minimizing the risk over the maximum eigenvalue of σ_{max} , or bychanging the threshold η to define new regions of performance.

$$\min_{\sigma_{max}} (\Re(\alpha, \beta))) \tag{15}$$

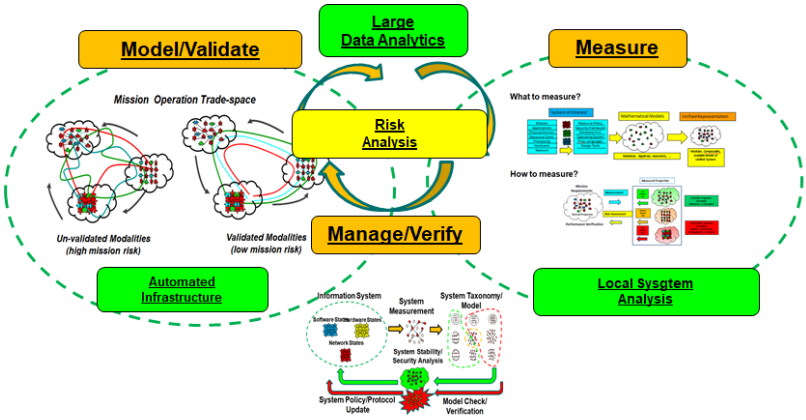

Figure 4) Integrated System Management

VI. Risk Bounds

Using this approach we can find an upper bound for the system risk to our distributed information system as is shown in Figure 5. The Chernoff Bound gives the average probability of error, P_e^* ,which can upper boundfor our Bayes risk

$$P_e^* \geq \exp(log \int f_{X|N}^{\alpha^*}(y|\Lambda = 1) f_{X|N}^{1-\alpha^*}(y|\Lambda = 0) dy) \tag{16}$$

where

$$\alpha^* = min_{0 \leq \alpha \leq 1} \int f_{X|\mathrm{N}}^{\alpha^*}(y|1) f_{X|N}^{1-\alpha^*}(y|0) \tag{17}$$

Our approach uses a fairly simplistic model for covariance. For more complex distributional models of systems, higher order methods that give a more complex estimate of the Bayes risk may be warranted. We use this as a simple example of how to bound the performance of such a distributed information system.

VII. Conclusion

We have developed an integrated strategy to measure, model, and manage a complex information system using a Bayesian risk model to enable the ability to assess whether a system is in a bounded and unsafe operating mode. Such an approach will allow, the ability to address high dimensional data but it uses a fairly simple model for

Bayesian risk. If system model accuracy is more important than computational efficiency, a more complex covariance model may be warranted to enable automated resilience .

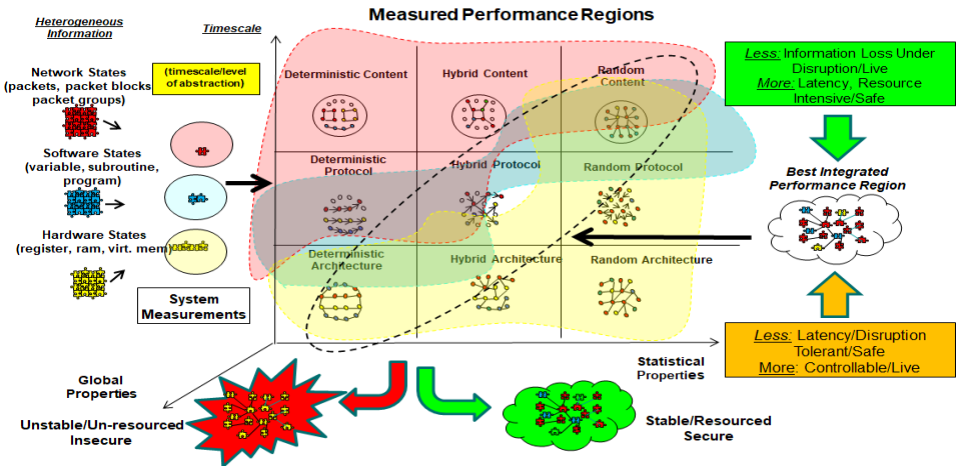

Figure 5) Regions of Risk Bounds

References

[1] Candès E., Romberg J., and Tao T., "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information." IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006

[2] Coifman, R.R., "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps", PNAS, 2005, vol. 102, no. 21.

[3] Donoho, D., Johnstone, I., "Minmax Estimation via Wavelet Shrinkage", Stanford University Technical Report, 1991.

[4] Mallat S., Zhang Z., "Matching Pursuits With Time Frequency Dictionaries", IEEE Transactions on Signal Processing, 41(12):3397-3415, December 1993.

[5] Scharf, L, *Statistical Signal Processing*, Addison Wesley, 1991.