SPECTRAL BASED METHODS THAT STREAMLINE THE SEARCH FOR FAILURE SCENARIOS IN LARGE-SCALE DISTRIBUTED SYSTEMS

Fern Y. Hunt  
Applied and Computational Mathematics Division  
National Institute of Standards and Technology  
100 Bureau Drive, Mail Stop 8910,  
Gaithersburg, Maryland, U.S.A.  
email: fern.hunt@nist.gov

Katherine Morrison  
University of Nebraska-Lincoln  
Department of Mathematics  
Avery Hall  
Lincoln, Nebraska 68588  
email:

Christopher Dabrowski  
Advanced Networking Division  
National Institute of Standards and Technology  
100 Bureau Drive, Mail Stop 8920,  
Gaithersburg, Maryland, U.S.A.  
email: christopher.dabrowski@nist.gov

ABSTRACT
We report our work on the development of analytical and numerical methods that enable the detection of failure scenarios in distributed grid computing, cloud computing and other large scale systems. The spectral (i.e. eigenvalue and eigenvector) properties of the matrices associated with a non-homogeneous absorbing Markov Chain are used to quickly compute the long time proportion of tasks completed at a given setting of parameters. This enables the discovery of critical ranges of parameter values where system performance deteriorates and fails.

KEY WORDS  
grid computing, cloud computing, failure, spectral expansion, non-homogeneous Markov Chain

1 Introduction
In recent years, the advent of large scale distributed systems such as computing grids and commercial cloud systems has made mass computing services available to populations of users on demand. These systems are dynamic, heterogeneous and due to the interactions of its many components subject to the emergence of unpredictable system wide behaviors that can have deleterious consequences [1]. Their rapid growth and increasing economic importance underline the need for tools and methodologies that enable understanding and prediction of complex system behavior in order to insure the availability and reliability of these services. Key questions are the effect of changes in workload, system design and other operational parameters on overall system performance. For example, studies of alternative economic strategies [2],[3],[4] and system failure scenarios [1] have shown that small variations in key system parameters can lead to large differences in performance. By large scale simulation we mean the discrete event simulations that simulate in detail the various stages encountered by each individual task over time. While the large scale simulations used in these studies are more practical than operational testbeds, computational expense rising dramatically with model size and number of tasks is a critical roadblock to extensive investigation of dynamical behavior in large scale systems.

To address this situation, we introduced in earlier work, a succinct Markov chain representation of the dynamics of a large scale grid system over time. The chain simulates the progress of a large number of computing tasks from the time they are submitted by users to the time they either complete or fail. The evolution of the Markov chain itself occurs in discrete time through a set of transition probability matrices (TPMs). Each TPM simulates the grid system over a distinct time period and thus the Markov chain is piece-wise homogeneous. Changes in system parameters can be modeled by perturbing the TPMs of the Markov chain. The corresponding sample paths are altered and represent altered system execution paths that arise as a result of perturbed system parameters [8]. Through systematic perturbation of the TPM matrices followed by simulation of the resulting perturbed Markov chain we were able to identify scenarios that led to degradations of system performance and system-wide failure. Our results compared very favorably with large scale simulation results and were obtained with a substantial reduction in computational cost [8]. One reason for this is that the statistics of the behavior of a population of tasks are summarized by the Markov chain while individual tasks must be tracked in the large scale simulation. Nevertheless, as the number of states of the Markov chain grows, the computational cost of this method significantly increases. Thus it is very difficult to quickly identify the set of perturbed TPMs that lead to system deterioration or failure when the number of states is large. Prediction and ultimately control of these systems will depend on the ability discover these scenarios quickly and perhaps in real time. This is the motivation for the work
that is briefly reported here. Our results constitute a proof of concept as the Markov model we discuss has just $n = 7$ states. However current work centers on larger systems including cloud computing systems.

In the context of grid computing what do we mean by failure scenarios? All grid computing systems have basic requirements called service guarantees that must be fulfilled. The failure to do so results in deterioration or outright failure in system performance. Service guarantees are of three types. First, the service discovery guarantee refers to the ability of a grid system to provide necessary information to users about available computing services including relevant updates. The service engagement guarantee insures that qualified users who have discovered a needed and available service are allowed to engage that service. Finally, the service fulfillment guarantee simply states that once a service has been engaged i.e. a service level agreement (SLA) has been agreed to: both user and service provider must adhere to its terms. In the large scale simulation and the Markov model, a failure scenario is a setting of operational parameters modeling the non-fulfillment of one or more guarantees, whose corresponding execution paths lead to system failure. A major goal of perturbation studies is to answer questions such as "at what point of incremental increase of guarantee non-fulfillment does system performance begin to degrade rapidly?" and "what specific actions by providers or users affect non-fulfillment of a particular guarantee?". This brief paper and the work in [5] argue that a Markov Chain approach can be used to answer these questions by approximating the transient behavior of a real world grid system (using large-scale simulation as a proxy). This paper summarizes our work on the development of analytical and numerical methods for discovering variation or perturbations in operational parameters that lead to decreases in system performance and system failure. They are based on properties of absorbing Markov chains and their associated matrices. We refer the reader to [5] for more detailed discussion and derivation of our method. Our contribution is twofold, first, a method for quickly generating the time course of a key variable of the system, the proportion of tasks completed for TPMs modeling the normal operation of the system as well as for perturbed TPMs depicting the operating parameters that lead to decreased performance or failure. When the eigenvalues of the transient submatrices associated with the TPMs (see section 3) have well separated eigenvalues, the method works particularly well. Secondly we developed a function that measures the effect of perturbations on the spectral properties (i.e. the eigenvalues and eigenvectors) of the matrices associated with the Markov chain. Depressed levels of the function (large changes in the spectral properties) correlate well with deterioration in performance. We were able to identify all of the failure scenarios found by large scale simulations, however the correlation is not perfect. In a small number of cases, depressed values occurred without any sign of a decrease in system performance. Nevertheless, we believe that this methodology offers a promising approach to the development of a set of tools for the rapid, high-level monitoring of large scale systems. The idea of using the spectral properties of Markov chains to approximate the dynamics of a perturbed Markov process is an old one and there have many significant contributions to the theory and its applications to networks since then (see the references in [5]). The systems we consider here differ from these applications in several important respects. The dynamics of grid or cloud computing systems vary with time and system behavior can be conveniently represented in terms of distinct time periods. The resulting Markov chain is piecewise homogeneous rather than homogeneous and is thus time inhomogeneous. Moreover we are dealing with tasks that eventually leave the system so the dynamics are also absorbing rather than ergodic. Thus the assumptions commonly encountered in the literature on perturbations of Markov Chains do not apply here. In the next section we will present a very brief review of previous related work. A more extensive discussion along with references can be found in [5]. Following this in section (2.2), we will present a Markov chain model and explain how it is derived from the large scale simulation. This is followed in section 3 by a derivation of equation (8) on which our methods is based.

2 Previous Related Work and Description of Markov Chain Model

2.1 Previous Related Work

In grid computing, Markov chains have been used to model workload for schedule and load balancing [6], [7]. These works emphasize quantitative estimates of performance or reliability for a fixed set of operational parameters. Our interest here is understanding what effect the perturbation of these parameters have on overall system performance, particularly those associated with the failure to meet funda-
mental service guarantees as discussed in the introduction.

2.2 Markov Chain Model of Grid System

The lifecycle of an individual task can be represented in seven states, shown in Figure 1. This model is derived from a large-scale simulation [4] that studies operation of a grid over an 8-hour day. The Markov chain model is derived from a previous large-scale grid computing system model [1], [4] that simulates the progress of a large number of computing tasks from the time they are submitted to the grid for execution by an end user to the time they either complete or fail. The dynamics of the chain occurs at discrete time steps. Figure 1 shows this Markov model as a state diagram for a single task. The state diagram has complete or fail. The dynamics of the chain occurs at discrete time steps. Figure 1 shows this Markov model as a state diagram for a single task. The state diagram has $n = 7$ states: an Initial state, where the task remains prior to submission; a Discovering state, during which service discovery middleware locates candidate grid service providers to execute the task; a Negotiating state during which a Service Level Agreement (SLA) to execute the task is negotiated; a Grid Service Provider location state during which a discovered provider; and finally the Complete or Fail states. Transitions between states, illustrated by the arrows in Figure 1, represent actions taken by the grid system to process a task as described in [8].

The Markov model is described by a state at a given time and the transition to a successor state occurs at the next discrete time step. It is considered an absorbing chain because all tasks ultimately must enter one of two absorbing states, Complete or Fail, from which they cannot leave. The Markov model is random in the sense that the successor state can only be identified by the probability of its occurrence, given the history, i.e. the past states of the chain. To understand how the transitions in time occur, the Markov property must be defined. Given a fixed time $m$, let $X_m$ be the state of the chain at that time. $\text{Prob}[X_m = s_j | X_{m-1} = s_i, \ldots, X_1]$ is the probability that $X_m = s_j$ given the past states of the chain. The Markov property states that the only relevant part of the past that is needed to determine the probability of transition to $s_j$ is $X_{m-1}$. So that $\text{Prob}[X_m = s_j | X_{m-1} = s_i, \ldots, X_1] = \text{Prob}[X_m = s_j | X_{m-1} = s_i] = p_{ij}(m)$

To convert the state model to a discrete time Markov chain we observed that the large scale simulation was time inhomogeneous over the period of a day with 2 hour periods where the state transitions were homogeneous. Letting $d=7200s$ be the length of this homogeneous time period, $h=85s$ was defined to be the duration of a single Markov chain step. Therefore the number of Markov chain time steps in a single time period is $S = d/h = 85$. The values of the transition probabilities $p_{ij}(m)$ were computed counting the frequency of transitions between states $i$ and $j$ over a simulated duration. Specifically, if $f_{ij}$ was the number of transitions from $s_i \rightarrow s_j$ that occurred during the homogeneous time period into which $m$ fell, and $\sum_{k=1}^{n} f_{ik}$ was the number of transitions out state $s_i$ during that period, then

$$p_{ij}(m) = \frac{f_{ij}}{\sum_{k=1}^{n} f_{ik}}.$$

Here $n$ is the number of states. The computation was repeated for each pair of states and there resulted a matrix that can be used to describe the possible states of the Markov Chain at time $m$. To see this let a complete description of the state of the chain be given by the row vector $v_m$ whose $j$th element is the probability that $X_m = s_j$. The vector $v_m$ is called the state vector at time $m$. Denote the index of the homogeneous time period into which $m$ falls by the notation $tp(m)$, the time period for $m$. Equation (1) is a formula for calculating the elements of the matrix $P_{tp(m)}$ of transition probabilities. The matrix itself is called the transition probability matrix or TPM. Using properties of basic (conditional) probability we have,

$$v_m = v_{m-1}P_{tp(m)}$$

Arguing inductively, it can be seen that the value of the state vector at any time $m$ subsequent to an initial time can be found by multiplying the current state vector on the right by the appropriate matrix $P_{tp(m)}$. If the initial state vector is $v_0$, the state vector at any time $m$ can be expressed in terms of the TPMs for the periods that occurred during the $m$ steps. Therefore if the number of these periods is $k$ we have $v_m = v_0P_1^k \cdots P_k^kP_{k+1}^k$.

The elements of the state vector $v_m$ at each time step are ordered so that the first element is the probability or proportion of tasks in the Initial state at time step $m$, the 6th element is the probability or proportion of tasks that are in the Complete state, and the 7th element gives the probability or proportion of tasks in the Fail state. The states are divided into absorbing states Complete and Fail and non-absorbing states—the remaining states. The rows of the TPM corresponding to the absorbing states have a single non-zero element 1, non-absorbing states have non-negative elements. In all the cases the sum of the elements in any row is 1. Ordering the states as we have also means that all the TPMs are in the canonical form for absorbing Markov Chains. This form is illustrated in Figure 2. Here the TPM is divided into 4 submatrices, $Q$ the matrix of transition probabilities of $s_i \rightarrow s_j$, where $s_i$ and $s_j$ are non-absorbing states; $R$ is the submatrix of transition probabilities from non-absorbing states to absorbing, $0$ is a submatrix of zeroes because transition from an absorbing state to a non-absorbing state is impossible. Finally, the identity matrix $I$ shows that once the chain reaches an absorbing state it remains there.\(^1\)

For our application all tasks are initially in the Initial state so that $v_0$ is the vector with 1 in the first component and 0

\(^1\)The TPMs are available online at...
3 Derivation of formulas used for the Results

3.1 Approximating the proportion of tasks completed

The canonical form for absorbing Markov Chains and the spectral representation theorem (see [5] for details and references) are used to derive a convenient approximation for the task completion probability as a function of time. If \( i \) is the index for the \( t \)th time period, the corresponding TPM in canonical form is

\[
P_i = \begin{pmatrix} Q_i & R_i \\ 0 & I \end{pmatrix}
\]

where \( I \) is the 2 x 2 identity matrix and \( 0 \) is the 4 x 2 matrix of zeroes depicted in Figure 2. The matrix product in (2) can be rewritten in terms of the submatrices as

\[
P_1^S \cdots P_k^S P_{k+1}^t = \begin{pmatrix} Q_1^S \cdots Q_k^S Q_{k+1}^t & A_m \\ 0 & I \end{pmatrix}
\]

where the matrix \( A_m \) is given by

\[
A_m = \left( I + \sum_{j=1}^{S-1} Q_j^t \right) R_1 + \sum_{l=2}^{k} Q_1^S \cdots Q_{l-1}^S \left( I + \sum_{j=1}^{S_l} Q_j^t \right) R_l + Q_1^S \cdots Q_k^S \left( I + \sum_{j=1}^{S_k} Q_j^t \right) R_{k+1}.
\]

where \( k = tp(m) - 1 \). The dimensions of the submatrices in (5) are the same as those in Figure 2. Since \( v_0 \) is a vector with a single non-zero element 1 in the first component we can find the proportion of tasks that complete by time \( m \) by computing the (1,6) of the matrix product on the left hand side of (2), i.e. the (1,1) element of \( A_m \).

The formula we use is an approximation of \( A_m \) based on the eigenvalues and corresponding projections onto the eigenspaces of the leading eigenvectors of the \( Q_i \). First we note that for all the TPMs derived from the long term simulations and for all loads, the eigenvalues of the \( Q_i \) were distinct. We will therefore assume this condition although it can be relaxed. Each \( Q_i \) has 5 eigenvalues which will be indexed by \( r = 1 \cdots 5 \) where the ordering is by absolute value (or modulus), so the first eigenvalue has the largest absolute value (or modulus). We found and we assume that the first eigenvalue of \( Q_i \) is not close to the boundary of the unit circle in the complex plane. The \( r \)th eigenvalue of \( Q_i \) is denoted by \( \lambda_i^{(r)} \). The matrix \( Q_i \) and its powers can be written in terms of the eigenvalues and corresponding projections as:

\[
Q_i = \sum_{r=1}^{5} \lambda_i^{(r)} \Psi_i^{(r)} \\
Q_i^t = \sum_{r=1}^{5} \left( \lambda_i^{(r)} \right)^t \Psi_i^{(r)}
\]

where \( \Psi_i^{(r)} \) is the projection onto the eigenspace of the \( r \)th eigenvector of \( Q_i \) and \( \epsilon \) is a power of \( Q_i \). Here the facts that the product \( \Psi_i^{(r)} \Psi_i^{(r')} = 0 \) if \( r \neq r' \) and \( \Psi_i^{(r)} \Psi_i^{(r)} = \Psi_i^{(r)} \) are used. Our approximation centers on the case in (7) where \( e = S \), the number of time steps in a period. Since \( S = 85 \), it is clear that \( |(\lambda_i^{(r)})| \) is very small for \( |\lambda_i^{(r)}| \) small enough. In fact any eigenvector with modulus less than .88 will satisfy \( |(\lambda_i^{(r)})| < 2 \times 10^{-5} \) so that if equation (7) is substituted in (6), the contribution from terms containing these eigenvalues is quite small. The approximation is based on retaining just the terms in the spectral expansion of powers of \( Q_i^t \) that come from eigenvalues with modulus more than .88. The choice of .88 is based on \( S \) and the desired accuracy. The question of how many leading eigenvalues are needed depends on the value of \( S \), the number of Markov chain steps in a period, and \( \epsilon \), the order of the approximation desired. In particular one would accept only eigenvalues \( \lambda \) for which \( |\lambda| > (\epsilon)^{\frac{1}{2}} \). In our applications the matrices do have a well separated spectrum and
for the choice of $S$ and $e$ (see section 2), two or three eigenvalues suffice. The argument we have presented here really only depends on a sufficient separation between the leading eigenvalues and the remaining ones. In all the TPMs derived from the 8 hour simulation, there were 3 eigenvalues larger than .88. Thus the expansion in (7) for $e = S$ and be approximated by the first 3 terms. In the 640 hour simulation there were 2 eigenvalues larger than .88 so 2 terms were used. A useful property of absorbing Markov Chains permits the submatrix $R$ of any TPM $P$ to be written in terms of the the eigenvalues and projections of $Q$ and the matrix $V$ whose columns are the leading eigenvectors corresponding to the double eigenvalue 1 of $P$ with the rows corresponding to the absorbing states removed. We omit many details because of space and refer the reader to [9] where this is discussed and then applied in [5]. For the 640 hour simulation where only terms containing the two leading eigenvalues are retained an approximation to $Q^3$ results. Substituting these expressions into (6) produces the following approximation for $A_m$:

$$A_m \approx \left[ I - (\lambda_1 S^i)^i - (\lambda^{(2)}_1 S^i)^i \right] V_i + \sum_{i=2}^k \prod_{i=1}^{i-1} \left[ (\lambda^{(1)}_i S^i)^i + (\lambda^{(2)}_i S^i)^i \right] V_i + \prod_{i=1}^k \left[ (\lambda^{(1)}_i S^i)^i + (\lambda^{(2)}_i S^i)^i \right] V_{k+1}$$

(8)

At each time step the complexity of using (8) for computing the cumulative proportion of tasks completed (including a count of the number of operations required to find the leading eigenvectors and eigenvalues) is $O(n^2)$ and is no larger than the complexity of (2) [10]. Thus (8) is significantly faster than the large scale simulation when $n$ is large. Equation (8) gives an analytical expression for the cumulative proportion of tasks completed as a function of time and compares well with the exact calculation obtained using (2) (see the Results section). Moreover, the formula links the changes in system performance arising from parameter perturbations to changes in the spectra of the submatrices $Q_i$. A natural question then is 'Can changes in the spectra due to perturbations signal the potential for system failure?' We address this question in the next section.

3.2 A Spectral Based Signal for Deleterious Perturbations

In addition to gaining some analytical insight into the mechanism of system failure, (8) also shows that $\{\lambda^{(p)}_i\}$, $\{\Psi^{(p)}_i\}$, and $\{V_i\}$, where $p$ is the index of eigenvalues that are retained and $i = 1, \cdots N$ is the index for completed time periods; determine (to a good approximation) the cumulative proportion of tasks that complete at time step $m$. Thus changes in the TPMs due to parameter changes will also change these spectral quantities. Starting with a set of TPMs with transition values that produce a normal set of execution paths and task completion profiles, we introduce measures of the deviation in spectral quantities resulting from a perturbation, $\Lambda_1$, the average over all $N$ time periods of the change (in percent) in the first two eigenvalues,

$$\Lambda_1 = 100 \cdot \frac{1}{N} \sum_{i=1}^N \frac{\left[ (\lambda^{(1)}_i + \lambda^{(2)}_i) - (\lambda^{(1)}_i + \lambda^{(2)}_i) \right]}{\lambda^{(1)}_i + \lambda^{(2)}_i}.$$  

(9)

The perturbed value of each variable in (9) and subsequent equations is distinguished by a prime symbol $'$. $\Lambda_2$ measures the average percentage change in the projections onto the eigenspace for the first two eigenvectors. Here $\text{norm}(A)$ is the square root of the sum of squares of the entries of the matrix $A$.

$$\Lambda_2 = 100 \cdot \frac{1}{N} \sum_{i=1}^N \frac{\text{norm}\left( (\Psi^{(1)}_i + \Psi^{(2)}_i) - (\Psi^{(1)}_i + \Psi^{(2)}_i) \right)}{\text{norm}(\Psi^{(1)}_i + \Psi^{(2)}_i)}.$$  

(10)

The next two quantities involve the percentage change in bilinear functions of the eigenvalues, eigenvectors and projections we discussed.

$$\Lambda_4 = \Lambda_1 \cdot \Lambda_2$$

(12)

$$\Lambda_5 = 100 \cdot \frac{1}{N} \sum_{i=1}^N \frac{\text{norm}\left( (\lambda^{(1)}_i \Psi^{(1)}_i + \lambda^{(2)}_i \Psi^{(2)}_i) V_i - (\lambda^{(1)}_i \Psi^{(1)}_i + \lambda^{(2)}_i \Psi^{(2)}_i) V_i \right)}{\text{norm}(\lambda^{(1)}_i \Psi^{(1)}_i + \lambda^{(2)}_i \Psi^{(2)}_i) V_i}.$$  

(13)

To determine a function for detecting deleterious perturbations we treated $\Lambda_r$, $r = 1, \cdots 5$, as independent variables and performed a fit to the percentage change in the proportion of tasks completed. Specifically, elements of the TPMs were systematically perturbed. Each perturbation corresponded to a change in the transition probability between two non-absorbing states. The spectral quantities $\Lambda_r$ were computed for each such perturbation and the corresponding percentage change in the cumulative proportion of tasks completed was also computed. A multilinear regression fit of these values resulted in a fitted expression for the percentage change in the proportion of completed tasks as a function of the spectral quantities:

$$F_{\text{spec}} = \sum_{r=1}^{5} c_r \Lambda_r$$

(14)
4 Results

Large scale simulations depicting the operation of a grid computing system over a day lasting 8 hours and another depicting 80 8-hour days (640 hours) with loads varying between 50 and 100% were compared with the Markov model and the theoretical approximations discussed in section 3.1. The cumulative proportion of tasks completed was plotted as a function of time for the large scale simulation. The corresponding quantity for the Markov model, the tasks completed or 6th component of the state vector was also plotted as a function of time, along with the theoretical approximation of this same quantity, based on the formula (8) for a variety of loads. In Figure 3, the 8 hour large scale simulation is in black while the Markov model and theoretical approximation are plotted in red and blue dashes respectively. Figure 4 shows the results of the 640 hour simulation. Both systems are at a 75% load. The results of (2) and (8) closely agree.

In the light of our discussion in the introduction we examine failure scenarios in terms of a critical level of non-fulfillment of a service guarantee. The degree of non-fulfillment can be quantified in both the large scale simulation and the Markov model by its effect on the frequency of transition between relevant states. For example the failure to fulfill the task service guarantee can increase the probability of transition from Monitoring to Negotiation while simultaneously and proportionately lowering the probability of transition from Monitoring to Completion. Methods for modeling these events in a Markov chain involve perturbation of individual elements of the TPMs. Choosing which elements to perturb and at what level is a difficult and computationally expensive task. In [8], we discussed a systematic method for doing this based on the Markov chain that resulted in a two orders of magnitude reduction in time to identify all failure scenarios including the service guarantee and the associated transition we mentioned. For lack of space we cannot describe the procedure here but refer to the references for a discussion. Figure 5 shows the change in the final cumulative proportion of tasks completed, as a function of the transition probability of Monitoring to Negotiation for the large scale simulation and the Markov model. The computation for the large scale simulation was done by direct simulation for each level of perturbation. Equation (2) was used to compute $v_m$ for the final time step $m$ at each level of perturbation. Using the approximation in (8) instead, we see that the cumulative tasks completed curve is in very close agreement to the exact Markov model. All of the curves show the deterioration and eventual failure of the system after a critical transition probability is reached. Figure 6 is the result computing the (cumulative) proportion of tasks completed as a result of a violation of the discovery guarantee. Here this event is measured in terms of the probability of transition from Discovery to itself. The Markov model and the approximate computation through (8) agree very well and are good enough for the approximation to identify the critical transition probability that leads to a significant decrease in tasks completed and then system failure.

We explored the question of how well changes in the spectral properties of the submatrices $Q_i$ predict decreases in the cumulative proportion of tasks completed, by fitting the $\{\Lambda_r : r = 1 \cdots 5\}$, a measure of these changes to changes in the tasks completed (see section 3.2). In the 640 hour simulation the changes were produced by systematic and exhaustive perturbations of the TPMs at 75% load. We used perturbation cases where an entry that is decreased is decreased to zero while the remaining entries are increased. This was done to maximize the chances of identifying deleterious perturbations. Using multilinear regression analysis a fitted function $F_{spec}$ was produced with $\{c_r : r = 1 \cdots 5\}$ as the regression coefficients. Two perturbation methods were employed. Under the primary decrease perturbation method we obtained

$$c_1 = -6.6057, c_2 = 0.8297, c_3 = -1.0580$$
$$c_4 = -0.0102, c_5 = -0.0287$$

The quality of the fit was determined by the coefficient of determination $r^2 = 0.9373$ and the residuals shown in Figure 7. The horizontal axis indexes the perturbation cases while the horizontal axis shows the residuals (dots) and vertical bars that delineate the confidence interval for the residuals. There are 2 outliers at cases 5 and 34. Thus, $F_{spec}$ defined in (14) is a good fit. From the magnitude of the $c_r$ it can be seen changes that in the leading eigenvalues of $Q_i$ and the $V_i$, i.e. the leading eigenvectors of the TPM $P_i$ are the most influential in determining changes in the proportion of tasks completed. The same analysis was carried out for the same simulation using a different perturbation method and similar results were obtained. Elevated values of $\Lambda_r$ (see (9)-(13)) were associated with all of the deleterious perturbations found using graph theory methods. However in a small number of cases, values were elevated but there was no drop in the proportion of tasks completed. The predictive power is not absolute. Thus elevated values of $\Lambda_r$ or low values of $F_{spec}$ are necessary but not sufficient indicators of performance loss but are still valuable signals e.g. in exploratory efforts to identify deleterious perturbations.

5 Conclusion and Future Work

We reported on the development of analytical and numerical methods that enable the detection of failure scenarios in large scale systems. The properties of an absorbing non-homogeneous Markov chain model of the system are used to quickly compute the tasks completed under varying system conditions. Our method is particularly effective when the submatrices $Q_i$ associated with the homogeneous time periods have well separated eigenvalues (i.e. there is a large spectral gap). In the model the operating parameters of the system are values of the transition probabilities (elements of the TPMs) controlling the rate of transition between states in the system (see Figure 1). Changes in these
values depict perturbations in real system parameters that occur because of violations of service guarantees as discussed in the introduction. A systematic search for such deleterious perturbations is facilitated by measuring their effect on the spectral properties of the $Q_i$. In section 3.2, we introduced a function $F_{spec}$ that measures the deviation of spectrum corresponding to perturbed TPMs from the spectrum corresponding to unperturbed TPMs of a system under “normal” operating conditions. The multilinear regression analysis we performed indicates that low values of $F_{spec}$ are a good indicator of potential performance loss due to deleterious perturbations. The regression analysis shows that changes in the leading eigenvalues of $Q_i$ and the eigenvectors $V_i$ (see section 3.1) are most influential in affecting system performance. Although the correlation is not perfect, we demonstrate that $F_{spec}$ can be used as an effective warning signal indicating that further analysis of the large scale system is needed. Alternatively, it can be used in conjunction with other methodologies such as the minimal cut set analysis discussed in [5].

To be able to predict threshold effects, where large changes in dynamics occur with relatively small parameter changes, we introduce an analytical formula which like (2) quickly generates the system dynamics over time. The agreement between the predicted transient behavior of the system under arbitrary conditions, calculated according to a straightforward computation of the cumulative proportion of tasks completed (2) and the approximation using (8) is very good. Both calculations agreed well with the large scale simulation. Tracking the long term cumulative proportion of tasks completed as a function of transition probability revealed the existence of a critical range of values (and therefore perturbations) that produce system deterioration due to service guarantee violations. Increasing the transition from Monitoring to Negotiation models a scenario where the level of non-fulfillment of the task service guarantee violation increases. The computations based on the Markov model and the approximation (8) are in good enough agreement to conclude that this critical range and its threshold are correctly identified (see Figure 5). Figure 6 shows the analogous computation for the self transition of Discovery to itself, modeling the violation of the discovery guarantee. As in the previous case the threshold for performance deterioration is identified. Finally these figures also show that the Markov approach has uncovered the unintuitive fact that the system is very robust to violations of the discovery guarantee under this scenario; in contrast, under increasing violations of the task service guarantee, system performance deteriorates rapidly. Our future work will center on the application of the methods discussed here to large scale systems where the state space of the absorbing...
Markov chain is quite large. Research has already begun on a cloud computing system. The spectral properties discussed here depend subtly on the underlying topology of the network and it would be interesting to explore the connections between the spectral approach and the minimal cut set analysis. The latter discussed in [5] is based on the underlying graph topology of the Markov chain.

References


