

NIST Technical Note 1857

Assessing Effects of Asymmetries, Dynamics, and Failures on a Cloud Simulator

Kevin Mills
James Filliben
Christopher Dabrowski

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.TN.1857>

NIST Technical Note 1857

Assessing Effects of Asymmetries, Dynamics, and Failures on a Cloud Simulator

Kevin Mills
James Filliben
Christopher Dabrowski
*Advanced Network Technologies Division
Information Technology Laboratory*

This publication is available free of charge from:
<http://dx.doi.org/10.6028/NIST.TN.1857>

March 2015



U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie E. May, Acting Under Secretary of Commerce for Standards and Technology and Acting Director

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Technical Note 1857
Natl. Inst. Stand. Technol. Tech. Note 1857, 62 pages (March 2015)

This publication is available free of charge from:

<http://dx.doi.org/10.6028/NIST.TN.1857>

CODEN: NTNOEF

Assessing Effects of Asymmetries, Dynamics, and Failures on a Cloud Simulator

K. Mills, J. Filliben and C. Dabrowski
 {kmills, jfilliben, cdabrowski}@nist.gov

Abstract. We characterize the effects of asymmetries, dynamics, and failures when introduced into a cloud computing simulator, which had previously been characterized under static, homogeneous configurations with various patterns of demand and supply. We aim to determine whether injecting these new factors into the cloud simulator causes fundamental shifts in macroscopic behavior and user experience. We find that introducing asymmetries, dynamics, and failures into the cloud simulator does not induce fundamental shifts in the factors driving simulator behavior, but these new parameters do exhibit interactions with the main driving factors, and with each other. Our findings suggest that a previous study, using the cloud simulator to compare virtual-machine placement algorithms, need not be extended to consider the effects of asymmetries, dynamics, and failures. These findings also increase our confidence in results from the previous study.

Keywords: cloud computing, sensitivity analysis, simulation modeling

Table of Contents

1. Introduction-----	1
2. Koala Cloud Simulator-----	3
2.1 Basic Model-----	3
2.1.1 Demand Layer-----	3
2.1.2 Supply Layer-----	6
2.1.3 Resource-Allocation Layer-----	6
2.1.4 Internet/Intranet Layer-----	8
2.2 Model Extensions-----	8
2.2.1 Dynamics-----	9
2.2.1.1 User Behavior-----	9
2.2.1.2 Cloud Reconfiguration-----	10
2.2.1.3 Cluster Reconfiguration-----	10
2.2.2 Failures-----	11
2.2.2.1 Node-Related Failures-----	11
2.2.2.2 Network-Related Failures-----	12
2.2.3 Asymmetries-----	13
2.2.3.1 Cloud-based Asymmetries-----	13
2.2.3.2 Network-based Asymmetries-----	13
3. Experiment Design-----	15
3.1 Variable Input Factors-----	16
3.1.1 Variable User Factors-----	16
3.1.2 Variable Cloud and Cluster Factors-----	16
3.1.3 Variable Node Factors-----	17

3.1.4 Variable Network Factors-----	18
3.1.5 Two-Level Values Chosen for Each Variable Input Factor-----	18
3.2 Fixed Parameters-----	22
3.2.1 Fixed User Parameters-----	22
3.2.2 Fixed Cloud and Cluster Parameters-----	23
3.2.3 Fixed Node Parameters-----	25
3.2.4 Fixed Network Parameters-----	25
3.3 Responses-----	26
3.3.1 User-Level Responses-----	26
3.3.2 Cloud-Level Responses-----	27
3.3.3 Cluster-Level Responses-----	28
3.3.4 Virtual-Machine-Level Responses-----	30
4. Results and Discussion-----	31
4.1 Response Dimensions-----	31
4.2 Factor Influence on Model Responses-----	33
4.3 Factor Ranking-----	43
4.4 Two-Term Interactions-----	44
4.5 Findings-----	52
5. Conclusions-----	53
References-----	54

List of Tables

Table 1. Descriptions of Virtual Machine Types Simulated in Koala-----	5
Table 2. Descriptions of selected simulated user types: processing users (PU), distributed modeling and simulation (MS) users, peer-to-peer (PS) users, Web service (WS) users, and data search (DS) users-----	5
Table 3. Descriptions of Selected Platform Types Simulated in Koala-----	6
Table 4. Alternative Criteria for Choosing Clusters and Alternative Heuristics for Choosing Nodes-----	7
Table 5. Variable User Input Factors-----	16
Table 6. Variable Cloud and Cluster Input Factors-----	17
Table 7. Variable Node Input Factors-----	17
Table 8. Variable Network Input Factors-----	18
Table 9. Values Chosen for Variable User Input Factors-----	19
Table 10. Values Chosen for Variable Cloud and Cluster Input Factors-----	20
Table 11. Values Chosen for Variable Node Input Factors-----	21
Table 12. Values Chosen for Variable Network Input Factors-----	22
Table 13. Values for Fixed User Parameters-----	24
Table 14. Values for Fixed Cloud and Cluster Parameters-----	24
Table 15. Values for Fixed Node Parameters-----	25
Table 16. Values for Fixed Network Parameters-----	26
Table 17. Identity and Definition of User-Level Responses-----	27
Table 18. Identity and Definition of Cloud-Level Responses-----	28
Table 19. Identity and Definition of Cluster-Level Responses-----	29
Table 20. Identity and Definition of VM-Level Responses-----	30

Table 21. Correlation and Cluster Analysis for Experiment Responses-----	32
Table 22. Ψ and Δ Computed for the Six Holdover Parameters-----	34
Table 23. Ψ and Δ Computed for the Four Asymmetry Parameters-----	34
Table 24. Ψ and Δ Computed for the Four Dynamics Parameters-----	35
Table 25. Ψ and Δ Computed for the Seven Failure Parameters-----	35
Table 26. Average Influence of Parameter Categories on Koala Behaviors-----	36
Table 27. δ Computed for Holdover Parameters on the 14 Koala Behaviors-----	37
Table 28. δ Computed for Asymmetry Parameters on the 14 Koala Behaviors-----	39
Table 29. δ Computed for Dynamics Parameters on the 14 Koala Behaviors-----	41
Table 30. δ Computed for Failure Parameters on the 14 Koala Behaviors-----	42
Table 31. Ranking of Input Factors Based on Decreasing Δ -----	43
Table 32. Large, Significant Two-Term Interactions from Previous Study-----	46
Table 33. Mapping of Eight Input Factors from Current Study to Previous Study-----	49
Table 34. Top 10 Two-term Interactions in the MM Group-----	50
Table 35. Top Eight Two-term Interactions in the MU Group-----	51
Table 36. Top Six Two-term Interactions in the UU Group-----	51

List of Figures

Figure 1. Schematic of Koala Organization into Five Layers-----	4
Figure 2. Finite-State Machine Representation of Koala User Behavior-----	23
Figure 3. Two-term Interactions from a Previous Study-----	45
Figure 4. Two-term Interactions from the Current Study-----	48

1. Introduction

In previous work [1], we compared various algorithms for placing virtual machines (VMs) on physical machines in Infrastructure-as-a-Service (IaaS) cloud computing systems. The comparison was based on Koala, an IaaS cloud simulator developed at NIST. Prior to conducting the comparison, we determined [2] the relative influence of Koala input factors on model behavior. Subsequently, we compared 18 VM-placement algorithms under carefully chosen combinations of the six input factors that most influenced Koala behavior. Three of the six input factors defined system loads, including: the number of users, the number and type of VMs desired by each user and the duration for which users held VMs. The remaining input factors defined system capacities, including: the number of clusters in the cloud and the number and type of physical nodes in each cluster. Each VM placement algorithm was evaluated under an identical set of 32 parameter combinations, chosen using a 2^{6-1} orthogonal fractional factorial experiment design. The chosen parameter combinations represented various relationships between demand and supply within a cloud. Further, clouds were constructed using homogeneous cluster sizes. Naturally, we later wondered whether our comparison of VM-placement algorithms would prove valid in more complex situations, where demand and supply relationships changed with time, where system components could fail, and where clouds could be composed from clusters of varying sizes? To answer this question, we conducted a follow-on study, documented in this Technical Note.

In outline, the plan for our follow-on study required a maximum of three steps. First, we extended Koala to permit various asymmetries, dynamics, and failures that were not included in the original simulator. We also devised input factors to control these extensions. Second, we conducted a sensitivity analysis to determine if the new input factors exhibited significant influence on Koala behavior, when compared to the six most influential input factors we found for the original model. If additional input factors were found to influence Koala behavior, then the third step would require us to repeat our initial comparison of VM-placement algorithms, but with an expanded set of input factors and derived parameter combinations. On the other hand, if our sensitivity analysis found that the same input factors proved most influential in driving Koala behavior, then we could omit the third step, because the VM-placement algorithm comparison would be invariant to the addition of asymmetries, dynamics, and failures. This Technical Note describes the Koala simulator extensions we devised, and also explains the design and results from a sensitivity analysis of the extended simulator. We do not describe the third step in our follow-on study because we found that step was unnecessary, since our sensitivity analysis revealed that Koala continues to be influenced mainly by demand-supply relationships, even when a simulated cloud exhibits asymmetries, dynamics, and failures.

The remainder of this Technical Note is organized into four main sections. Section 2 describes the basic Koala simulator and also the extensions we made to add asymmetries, dynamics, and failures. Section 3 identifies the 20 input factors we varied in our sensitivity analysis—along with the 2^{20-12} orthogonal fraction factorial (OFF) design we used to select 256 parameter combinations to simulate. Section 3 also defines the 45 responses we analyzed to understand how the 20 input factors influenced the behavior of

Koala. Section 4 reports our results, which include: identification of the behavioral dimensions exhibited by Koala, quantification of the influence that each factor had on Koala behavior, and a ranking of factor influence, from most to least influential. We also analyze two-term interactions among pairs of the 20 factors, in order to determine whether Koala is influenced more by interactions or main effects. While discussing our results, we compare them to results from our previous sensitivity analysis [2] of Koala. In a few cases, to aid our comparison, we introduce results from that previous study. Section 4 ends with a discussion of our findings. The Technical Note closes, in Sec. 5, with our conclusions.

2. Koala Cloud Simulator

Koala is a discrete-event simulator inspired by the Amazon Elastic Compute Cloud (EC2) [3]. Using published information describing the EC2 application programming interface (API) [4] and available virtual machine (VM) types [5], Koala models essential features of the interface between users and EC2. Since we intended to study resource-allocation algorithms, Koala needed to model only four EC2 commands: *Run Instances*, *Describe Instances*, *Reboot Instances* and *Terminate Instances*. On the other hand, no public information was available about the internal structure and operation of EC2. Lacking such details, the internal structure of Koala is based instead on the Eucalyptus (v1.6) open-source Cloud software [6]. Specifically, Koala models three Eucalyptus components: *cloud controller*, *cluster controller*, and *node controller*. As in Eucalyptus, Koala’s simulated cloud, cluster, and node controllers communicate using Web Services [7], which are also simulated. In constructing Koala, we modified the design of Eucalyptus in three ways. First, we extended the Eucalyptus *Run Instances* command to allow multiple VM types within a single request, which we inferred is possible in EC2. Second, we avoided centralization of node information at the cloud controller, permitting Koala to simulate clouds up to $O(10^5)$ nodes. Third, we allowed resource allocation to proceed partially in parallel (serializing only the commitment phase), which prevents long queuing delays during periods of intense user requests. In lieu of simulating details of a hypervisor and guest VMs, we added an optional sub-model based on analytical equations representing VM behavior with or without tasks. Below, we first describe (in Sec. 2.1) the basic Koala simulator, and then we discuss (in Sec. 2.2) the extensions we made to add dynamics, failures, and asymmetries.

2.1 Basic Model

Koala is organized as five layers (see Fig. 1): (1) demand layer, (2) supply layer, (3) resource-allocation layer, (4) Internet/Intranet layer, and (5) VM behavior layer. We describe each layer in turn, omitting the VM behavior layer, which is not used in the experiments discussed here.

2.1.1 Demand Layer

The demand layer consists of a variable number of users who, after a random startup delay, each perform cyclically over a simulation run. During each cycle a user requests a minimum and maximum number of instances of one or more of the VM types shown in Table 1. The VM types and quantities a user selects depend upon the user’s type (see Table 2), which is selected on each cycle based on a probability distribution. After selecting a type, a user randomly chooses a minimum (uniformly distributed from 1 to a max-min) and maximum (uniformly distributed from max-min to a max-max) number of instances to request for each associated VM type. The user then issues a corresponding *Run Instances* request to the cloud controller, which may respond with an allocation of instances between the minimum and maximum for each requested VM type or with a NERA (not enough resources available) fault. A *full grant* denotes that a user was allocated the maximum requested instances of each VM type. A *partial grant* denotes that allocated VMs were below the maximum requested. If given VM instances,

the user selects a holding time (Pareto distributed with a designated mean and shape). During the holding period, the user will first issue *Describe Instances* requests to determine when all instances are running, and will subsequently randomly reboot, terminate, and describe running instances. At the end of the holding period, the user will issue a *Terminate Instances* request to stop any running instances. After terminating all instances, the user will wait an exponentially distributed time (mean 30 min) and then start a new cycle.

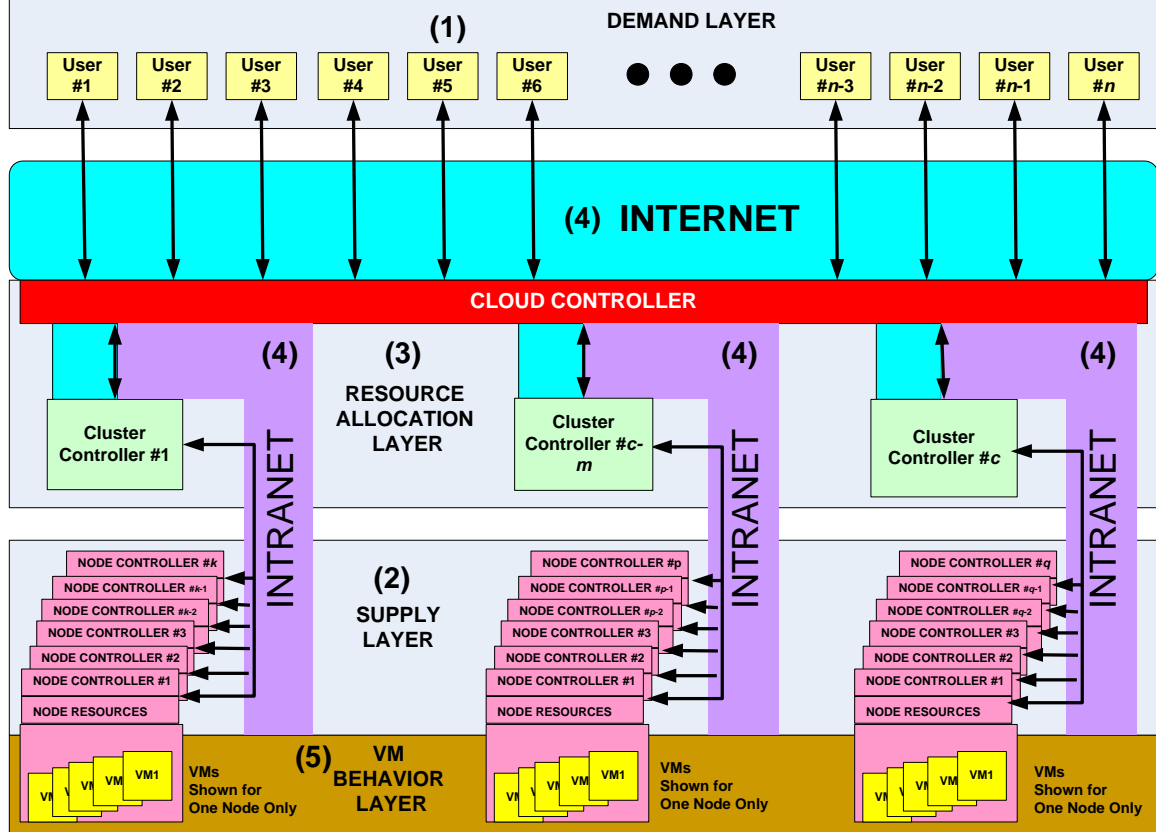


Figure 1. Schematic of Koala Organization into Five Layers

Since we believed differences in user persistence were not germane directly to the study of resource-allocation algorithms, we assigned fixed means for each stochastic distribution controlling related behaviors. If a user receives a NERA instead of being allocated instances, then the user waits an exponentially distributed time (mean 15 min) before retrying the request. A user will retry a failed request over a random period (mean 4 h) before resting for a random period (mean 16 h). If a user request cannot be honored within a random number of rest periods (mean 4), then the user abandons the request and starts a new cycle.

Table 1. Descriptions of Virtual Machine Types Simulated in Koala

VM Type	Virtual Cores		Virtual Block Devices		# Virtual		Arch.	Price in \$/Hour
	#	Speed (GHz)	#	Size (GB) of Each	Network Interfaces	Memory (GB)		
M1 small	1	1.7	1	160	1	2	32-bit	0.12
M1 large	2	2	2	420	2	8	64-bit	0.34
M1 xlarge	4	2	4	420	2	16	64-bit	0.96
C1 medium	2	2.4	1	340	1	2	32-bit	0.17
C1 xlarge	8	2.4	4	420	2	8	64-bit	0.68
M2 xlarge	8	3	1	840	2	32	64-bit	1.00
M4 xlarge	8	3	2	850	2	64	64-bit	2.00

Table 2. Descriptions of selected simulated user types: processing users (PU), distributed modeling and simulation (MS) users, peer-to-peer (PS) users, Web service (WS) users, and data search (DS) users

User		Max-Min	Max-Max	User		Max-Min	Max-Max	
Type	VM Type(s)	VMs	VMs	Type	VM Type(s)	VMs	VMs	
PU1	M1 small	10	100	PS1	C1 medium	3	10	
				PS2			10	50
PU3		100	500	PS3		50	100	
PU5		500	1000	WS1	M1 large M2 xlarge C1 xlarge	1	3	
PU2	M1 large	10	100	WS2	M1 large M2 xlarge C1 xlarge	3	9	
PU4		100	500	WS3	M1 large M2 xlarge C1 xlarge	9	12	
MS1	M1 xlarge	10	100	DS1	M4 xlarge	10	100	
MS2	M2 xlarge	10	100	DS2		100	500	
MS3	M1 xlarge	100	500	DS3		500	1000	
MS4	M2 xlarge	100	500					

2.1.2 Supply Layer

The supply layer consists of a variable number of clusters that each manages a variable number of nodes. When visiting an Amazon EC2 data center, we noticed the supply of nodes was composed of a limited number of platform configurations. This observation motivated us to define a fixed set of possible platform configurations for nodes. Upon creation, each node manifests, with some probability, one of the configurations shown in Table 3. Nodes retain their established configurations for the duration of a simulation run. For an instance to be allocated to a node, available resources on the node must be sufficient for the requirements specified by the instance’s VM type.

Table 3. Descriptions of Selected Platform Types Simulated in Koala

Platform Type	Physical Cores		Memory (GB)	# Physical Disks by Size				# Network Interfaces	Arch.
	#	Speed (GHz)		250 GB	500 GB	750 GB	1000 GB		
C8	2	2.4	32	0	3	0	0	1	64-bit
C14	4	3	64	0	4	0	3	2	64-bit
C18	8	3	128	0	0	4	3	4	64-bit
C22	16	3	256	0	0	0	7	4	64-bit

2.1.3 Resource-Allocation Layer

Koala patterns resource allocation after Eucalyptus procedures, which involve two decisions: (1) on which cluster to allocate the requested VMs and (2) on which nodes within the cluster to allocate the VMs. Allocating all VMs in a single request to the same cluster makes good sense because inter-VM communications would be local to a single cluster. Koala simulates three alternative criteria that the cloud controller may use to choose a cluster and six alternative heuristics that cluster controllers may use to choose nodes. Combining these criteria and heuristics (see Table 4) creates (3 x 6 =) 18 possible resource-allocation algorithms that Koala may use to place VMs onto nodes in a cloud.

For the current experiment, we allowed Koala to use either of two criteria (*percent-allocated* or *least-full first*) to select a cluster, while we fixed the algorithm (*first-fit*) that clusters used to select nodes. We made these decisions for two reasons. First, our previous experiment [1] found that varying a cloud controller’s criterion for choosing a cluster stimulated different Koala behaviors. Second, that same experiment also found that varying the heuristic cluster controllers used to choose specific nodes made little difference in Koala behavior. In the current report, we explain only two cloud-controller criteria for choosing a cluster and two cluster-controller heuristics for choosing nodes. Interested readers may consult our previous study [1] for an explanation of all 18 resource-allocation algorithms simulated by Koala.

Table 4. Alternative Criteria for Choosing Clusters and Alternative Heuristics for Choosing Nodes

Criteria for Choosing a Cluster		Heuristics for Choosing Nodes	
Identifier	Criterion Name	Identifier	Heuristic Name
LLF	Least-Full First	FF	First Fit
		LF	Least-Full First
PAL	Percent Allocated	MF	Most-Full First
		NF	Next Fit
RAN	Random	RA	Random
		TP	Tag & Pack

At the cluster level, Eucalyptus allocates VMs to nodes using one of two heuristics, while Koala simulates six, as listed in the right-hand side of Table 4. Here, we explain only the two (*first-fit* and *next-fit*) that are implemented by Eucalyptus. First-fit simply searches the nodes by identifier from first to last until a node is found that can accommodate a given VM type. Next-fit remembers which node last received a VM and begins its search from the next node identifier. Under either first-fit or next-fit, if the selected node cannot accommodate the VM, then the node controller reallocates the VM to the next node on the list. This process continues until the VM is created or until all nodes have been exhausted. If no nodes can create the VM, then the cloud controller receives a NERA.

At the cloud level, Eucalyptus can accommodate a choice of algorithms to select a cluster to which to assign all VMs in a request, but only one algorithm is implemented. The implemented algorithm, called *least-full-first*, polls the clusters to find out which can accommodate the VMs requested and then orders the list from the least to most full (Koala resolves ties in order of increasing cluster identifier). Then the cloud controller selects the first cluster from the list and asks that the VMs be created. If the VMs are created successfully, then the cloud controller returns the positive result to the appropriate user; otherwise, the cloud controller reallocates the VMs to the next cluster on the list. This process continues until VMs are created or until all clusters have been exhausted. If no clusters can create the VMs, then the user receives a NERA. Our previous study [1] also simulated a second cloud-level criterion, which orders clusters based on *percent allocated*, i.e., the cloud orders clusters by decreasing proportion of the requested VMs that can be allocated (Koala still resolves ties in order of increasing cluster identifier). For a given simulation, one parameter specifies a cloud-level criterion to use and another parameter specifies a cluster-level heuristic.

2.1.4 Internet/Intranet Layer

Koala assigns the cloud controller, cluster controllers, and users to sites (1000 here) randomly located at x,y coordinates on a grid ($12875 \text{ km} \times 12875 \text{ km}$ here, which spans a distance consistent with the globe). While somewhat unrealistic, random geographic layout makes a reasonable starting point for simulating inter-site distances. Before a simulation commences, cloud and cluster controllers are randomly placed on some number of sites. Node controllers are placed on the same site as the related cluster controller. At the beginning of each user cycle, a user is assigned randomly to one of the sites not occupied by cloud components. This arrangement divides message communications into two categories: (1) inter-site (Internet) and (2) intra-site (Intranet). Koala components communicate through simulated Web Services (WS) messages, which each comprise a uniformly distributed number (1 to 10 here) of packets. Individual packets are subjected to transmission delays (1 Gigabits per second rate here, which is reasonable for intra-site communications and is somewhat optimistic for inter-site communications) and propagation delay. For inter-site messages, propagation delay depends on distance and simulated router hops, while propagation delay within sites varies randomly (mean 250 ns here; reasonable within a site). Individual packets are also subjected to a loss rate (10^{-12} here for intra-site packets, which are rarely lost in practice). To simulate Internet congestion, the loss rate for inter-site packets varies uniformly within a range (10^{-3} to 10^{-8} here). Lost packets are retransmitted, but only for a maximum number (3 here) of attempts, after which the related WS message is declared undeliverable.

2.2 Model Extensions

As described above, the basic Koala model simulates a population of users seeking and holding VMs of varying size and quantity from a cloud composed from physical nodes of varying size and quantity. Basic Koala also simulates cyclic behavior among users and various (more or less) realistic aspects of network characteristics, such as transmission and propagation delays, and packet losses and retransmissions. Koala also simulates network paths with characteristics that differ for communications within and between sites.

The basic model of Koala lacks several properties that might appear in particular real clouds. For example, Koala simulates clusters of homogeneous size. While this might be a realistic characteristic for large industrial clouds, ad hoc clouds could be cobbled together from available clusters of differing sizes. As another example, Koala simulates a cloud where all components stay operational during a simulation, whereas, operational clouds may change in size, as clusters and nodes are added or subtracted and as nodes and communication paths fail or degrade. As a third example, Koala simulates a probabilistic user demand profile that does not change for the duration of a simulation, whereas, operational clouds might see user demands changing over time. These three examples cover a space of possibilities that can be labeled as dynamics, failures, and asymmetries. The basic Koala model did not allow for such possibilities; possibilities that, intuitively, could introduce stress into a cloud infrastructure. This raised questions about the robustness of our previous study [1]. Would our findings hold for a cloud

attempting to allocate resources under the stresses induced by dynamics, failures, and asymmetries? To investigate this question, we extended the Koala simulator, as described below.

2.2.1 Dynamics

We extended Koala with increased dynamics in three areas: (1) user behavior, (2) cloud reconfiguration, and (3) cluster reconfiguration. We discuss each area in turn.

2.2.1.1 User Behavior

We included three extensions into Koala so that users could change demands dynamically. The first user extension addressed behavior while holding VMs. We introduced a rule that required each user to attempt to maintain the count of VMs held to match some target number, representing the number of VMs obtained, but not voluntarily terminated by the user. We added this behavior to ensure users would attempt to obtain additional VMs when the cloud controller indicated VMs were dropped involuntarily in the cloud. Without this user behavior, the number of VMs held could fall due to failures (as explained below in Sec. 2.2.2), which would suggest that the user would be unable to complete intended tasks.

The second user extension added behavior that enabled each user to voluntarily increase or decrease demand while holding VMs. Four parameters control this extension. One parameter defines the probability that a user changes demand. Each user can change demand at most once during its holding period. A second parameter specifies the probability that a user will increase its demand for VMs. Subtracting this probability from one yields the probability a user will shrink demand. The remaining two parameters define the minimum and maximum extent of a user's demand change. These parameters are expressed as a percentage of the number of VMs obtained during a user's initial reservation. Whenever a user obtains additional VMs, the user's holding target is increased accordingly. Similarly, when a user reduces VMs held, the holding target is also reduced.

The third user extension enables an experimenter to induce temporal changes in the entire demand pattern of arriving users. Whereas basic Koala defines only a single distribution for user type probabilities, the extension allows user type probabilities to alternate between two distributions at designated points during a simulation. Demand-pattern alternation is controlled through three parameters: (1) the user types in one distribution, (2) the user types in a second distribution, and (3) the proportion of simulation time that elapses between alternations. At simulation startup, Koala then computes two user type distributions, and schedules times at which the pattern of user demands is switched between them. The probabilities for each user type in each distribution are chosen randomly. Selection of the first distribution to use is chosen by a simulated coin toss.

2.2.1.2 Cloud Reconfiguration

We extended the Koala cloud controller to simulate cloud reconfiguration, that is, the ability to add and remove clusters. Cloud reconfiguration is controlled by three parameters: (1) average time between reconfiguration events, (2) probability a cloud will add clusters (one minus this value is the probability a cloud will remove clusters), and (3) proportion of existing clusters to be added or removed. As a simulation run progresses, the cloud controller may repeatedly add and remove clusters, as determined by the parameter values. The number of clusters in a cloud is bounded within a range determined by Koala parameters. Adding clusters (which increases cloud supply) is relatively straightforward, but removing clusters requires substantially more behavior.

When a cluster is added, the number of nodes in the new cluster is chosen based on parameters defining relative cluster sizes, which could be uniform or which could include large and small clusters (see Sec. 2.2.3 below). When a cloud adds clusters, the supply of physical nodes increases and additional user demand can be absorbed.

When a cluster is removed, existing VMs assigned to the cluster must be relocated to another available cluster, or else terminated. To relocate VMs, the cloud controller employs the same logic defined for initially placing requested VMs on a cluster, except that care is taken to ensure VMs are not relocated onto other clusters that are also being deactivated during the same reconfiguration event. Each VM that cannot be relocated is terminated and the appropriate user is notified. For each notification of a terminated VM, a user may attempt to obtain a replacement VM in order to maintain the user's holding target.

2.2.1.3 Cluster Reconfiguration

We extended the Koala cluster controller to simulate cluster reconfiguration, that is, the ability to add and remove nodes. Cluster reconfiguration is controlled by three parameters: (1) average time between reconfiguration events, (2) probability a cluster will add nodes, and (3) proportion of existing nodes to be added or removed. As a simulation run unfolds, each cluster controller may repeatedly add and remove nodes, as determined by the parameter values. The number of nodes in a cluster is bounded within a range determined by Koala parameters. Adding nodes (which increases cluster size, and thus also somewhat increases cloud size) is relatively straightforward, but removing nodes requires more behavior.

When nodes are added, the number of nodes is chosen as a fraction of the existing cluster size. Adding nodes may allow accommodation of additional user requests for VMs and may also permit a cluster to accept VMs relocated when another cluster is deactivated.

When a node is removed, existing VMs assigned to the node must be relocated to another node within the same cluster, or else terminated. To relocate VMs, the cluster controller employs the same logic defined for initially placing requested VMs onto cluster nodes, except that care is taken to ensure VMs are not relocated onto other nodes also chosen for deactivation during the same reconfiguration event. Further, care is exercised to ensure that VMs that are undergoing termination are not relocated. Each VM

that cannot be relocated is terminated and the appropriate user is notified. For each notification, a user attempts to obtain a replacement VM, if required to maintain the user’s holding target.

2.2.2 Failures

We extended Koala to simulate failures, primarily in nodes and the network, as described below. We also allowed Koala users to randomly formulate incorrectly formatted requests, which can be considered a user-level failure.

2.2.2.1 Node-Related Failures

We added four types of node-related failures to Koala. One failure type simulates a node returning, probabilistically, a NERA response when a cloud controller attempts to place a VM on the node. This can be considered a failure because the node had previously indicated, in response to a *Describe Resources* request from the cluster controller, that the VM could be accommodated. Subsequent to this failure, the cluster controller will need to select another available node for the VM.

A second failure type encompasses node crashes, after which the node must be repaired and restored to service before it can host VMs. We extended Koala to simulate such crashes. One parameter (pair) expresses the average time between node failures, modulated by a shape parameter for a Weibull distribution. A shape parameter below one simulates early failures, while a shape parameter set to one simulates a constant, exponentially distributed failure rate. A shape parameter set above one simulates a failure rate increasing with time. For the experiments reported here, we set the shape parameter to 3. A second set of parameters specifies restoration latency, represented by a triangular distribution defining the minimum, typical, and maximum latencies. When a node fails, all VMs assigned to the node also fail. Since the failure is abrupt, no notifications are sent. Eventually, the cluster will notice that a node has failed and then attempt to restart the assigned VMs on another node, using the same method devised to relocate VMs prior to a graceful node shutdown. Of course, a user may also notice VM failures and then attempt to obtain replacements.

A third failure type encompasses individual component failures within nodes. We extended Koala to simulate failures for physical processors, disks, and memory. A parameter defines the average time between component failures, based on a Weibull distribution with the same shape parameter discussed above. When a failure occurs, Koala randomly chooses (equal probability) one of the three component types. Subsequently, some number of the chosen component type is randomly selected (uniform distribution) to fail. As with node failure, a set of parameters specifies the repair latency, represented by a triangular distribution. The minimum, typical, and maximum component-repair times are independent of restoration times for nodes. After an instance of component failure, a node may be unable to support all assigned VMs. Unsupportable VMs are terminated. The choice of VMs to terminate is made randomly (uniform distribution). The node’s cluster controller is notified of VMs terminated due to component failure. The cluster controller then attempts to restart the VMs on another node, using the relocation procedures described earlier.

A fourth failure type encompasses VM crashes within nodes, as may occur when VMs contain bugs. Two cases arise: boot failures and crashes during execution. Koala provides separate probabilities for each type of VM failure. If either event occurs for a given VM, then the appropriate user is notified. The user may attempt to acquire a replacement VM from the cloud.

2.2.2.2 Network-Related Failures

We extended Koala to simulate two categories of network failure: lost messages and cluster communication cuts. We provided two message loss parameters, one specifying the probability of message loss for communications within sites and one for communications between sites. We took this step based on the assumption that loss characteristics could differ between intra-site and inter-site communications.

Previous work with Koala discovered that high-message loss rates could lead to orphaned VMs in two categories: creation (and relocation) orphans and termination orphans. These orphans arose when users, cloud controllers, and cluster controllers were unaware of successfully allocated VMs; thus, VM leakage could consume significant cloud resources, limiting the ability of users to obtain VMs. Subsequently, Koala was extended with orphan-control procedures in order to contain VM leakage. The details of these procedures, along with the controlling parameters, can be found elsewhere [8]. Here, we give only a short summary.

Koala controls creation orphans by deleting VMs that have not been contacted by a user within a specified period of time (usually two hours) after creation. These procedures hold for newly allocated VMs. When reallocated through relocation, Koala provides a surrogate to generate artificial VM contacts, because a user is unaware of any VM relocation. Koala controls termination orphans by repeatedly issuing (with an increasing back-off interval) termination requests for a specified period of time or until receiving a reply stating that a VM has been terminated.

While simulated message losses are sporadic, we also extended Koala to simulate communication cuts on cluster interfaces. When a communication cut occurs, all affected messages flowing between a cluster and the cloud controller are discarded. This may include messages flowing into the cluster, messages flowing out, or messages flowing in both directions, depending on the specific nature of the cut. A parameter, mean time between failure (with Weibull shape), determines when a cluster suffers communication cuts. When a cut occurs, a direction is chosen (in, out, or both are equally probable). The cut continues until repairs are made. Repair latency is chosen randomly from a triangular distribution, specified with minimum, typical, and maximum latencies.

We extended Koala to simulate a (human) administrator to serve as the last line of defense against message losses and communication failures, which may leave the cloud and cluster controllers uncertain about whether specific VMs and nodes have actually been shut down. When such issues arise, the relevant nodes and VMs are assigned to the simulated administrator, who can take steps necessary to ensure that nodes and VMs are stopped. In the case of VM termination, if a cloud or cluster controller cannot establish for sure whether a VM has been terminated, then an assignment is generated for the administrator to manually terminate the VM, should it still be executing. In the case of node termination, a cloud or cluster controller can assign specific nodes to the simulated

administrator, who then manually shuts down the nodes. The simulated administrator is controlled by two parameters: (1) average attention latency and (2) average shutdown delay. Both parameters are exponentially distributed. Attention latency determines how often an administrator consults the queue of assigned tasks. Shutdown delay determines how much time it takes an administrator to complete one assigned task.

2.2.3 Asymmetries

We extended Koala to simulate a few asymmetries in two subsystems: the cloud and the network. We discuss each subsystem in turn.

2.2.3.1 Cloud-based Asymmetries

One asymmetry concerns whether the cloud resides on a single site or spans multiple sites. (Recall that site placement establishes relative geographic location.) As discussed earlier, a parameter specifies the number of sites reserved for the cloud. When only one site is reserved, then the cloud and cluster controllers share that site. When more than one site is reserved, then the cloud and cluster controllers are each assigned randomly (using a uniform distribution) to one of those sites. (Recall that all cluster nodes are placed automatically on the same site as the cluster controller.)

When distributed among multiple sites, Koala was extended with the option to assign each cluster to its own site. Under that option, when the number of sites reserved for the cloud is sufficient for all clusters (plus one site for the cloud controller), then each cluster will be placed on a unique site, and distinct from the site where the cloud controller resides. Such placement ensures that all communications between the cloud and cluster controllers transit the simulated Internet. If insufficient cloud sites have been specified, then some clusters will be placed together on a shared site. The assignment of a cluster to a shared site is made randomly, using a uniform distribution, and no cluster shares a site with the cloud controller.

Another asymmetry concerns whether all clusters have identical sizes, or whether cluster sizes may vary. We extended Koala with a parameter that specifies a rule for determining the relative size of clusters. The Koala version used for the current study implements two rules: uniform and 80/20. The uniform rule specifies that all clusters are the same size, as determined by another parameter: number of nodes per cluster. The 80/20 rule specifies that 20 % of the clusters will be large (containing four times the specified number of nodes per cluster), while 80 % of the clusters will be small (containing $\frac{1}{4}$ the specified number of nodes per cluster). This means the cloud contains the same number of nodes regardless of whether the rule is uniform or 80/20, but that the distribution of nodes to clusters is asymmetric under the 80/20 rule.

2.2.3.2 Network-based Asymmetries

For the network, we extended Koala to include two asymmetries: one affecting inter-site communication and one affecting intra-site communication. In the inter-site case, we added a parameter that allows distances between sites to be multiplied. Increasing inter-site distances induces additional communication delays, as packets must

transit more routers and links, and so suffer increased queuing and propagation delays. In the intra-site case, we added a parameter that allows delays on intra-site messages to be multiplied. We included a second parameter to specify the probability that any given intra-site message suffers delay multiplication.

3. Experiment Design

A typical problem arising with experiment designs for large models, which could have many parameters that can each take on very many values, is the number of experiments that must be run to explore every possible parameter combination at every possible of value. For example, a model with 20 parameters that can each take on 2^{32} values would require more than 10^{192} simulation runs, an infeasible number, requiring over 10^{187} years to execute, assuming each simulation could finish within 10 min. Exploiting the power of a 10,000-node computational cloud would reduce the required run time only to 10^{183} years. To overcome such problems, experimenters often adopt two-level designs, which examine each parameter at only two values, reducing the number of simulation runs required in the above example to just over 1 million. Adopting a two-level design for our example would reduce the required run time to below 20 years—still a long time. Renting 10,000 nodes from a computational cloud could finish such an experiment in as little as two weeks, but with a cost of around \$500K.

To reduce the time (or expense) arising from a full-factorial design, experimenters can adopt an orthogonal fractional factorial (OFF) design, as we did for our previous sensitivity analysis of the Koala simulator [2]. A fractional-factorial design samples only an affordable portion of the entire (full-factorial) space of parameter combinations. An OFF design selects the sampled combinations so that each parameter value occurs an equal number of times (balance) and each pair of parameter values also occurs equally often (orthogonality). By ensuring balance and orthogonality among the parameter combinations sampled from a full-factorial design space, OFF designs achieve two desirable properties, given the limits of the selected sample size. First, main effects estimates are representative of the main effects that would be found if a full-factorial experiment were conducted. This means that a list of experiment parameters, ranked by main effects, tends to be close to the true ordering that would result from a full-factorial experiment. Second, main-effects estimates will exhibit an uncertainty as small as possible, given the sample size.

In our previous study [2], we investigated 11 parameters using only 64 simulations. We did this by adopting a 2^{11-5} OFF design, which sampled the two-level, full-factorial design space of $(2^{11} =) 2048$ parameter combinations at only $(2^{11-5} =) 64$ points. In the current experiment, we examine 20 parameters (or input factors), which for a full-factorial, two-level experiment would require over a million simulation runs. To reduce the number of runs, we adopt a 2^{20-12} OFF experiment design, requiring only 256 simulation runs. In our experiments, each Koala simulation run required different processing times, varying from a few hours to a few days, depending upon the parameter combinations simulated. We deployed the simulations in parallel on a local cluster of 200 cores, which allowed us to complete all 256 simulations in less than two weeks.

Below, we identify and define the 20 input factors (i.e., parameters) we examined. These factors include the six parameters found, in our previous sensitivity analysis, to most influence behavior for the basic Koala model, as well as 14 factors controlling the dynamics, failures, and asymmetries included in the extended Koala model. After describing the input factors for our experiment, we then identify and define 45 system responses we analyzed to understand the behavior of the extended Koala model. While many of these responses were included in our previous studies [1-2], we added some

responses to reflect user behaviors that appeared only in the extended Koala model.

3.1 Variable Input Factors

We organize our discussion of variable input factors into four parts: (1) user factors, (2) cloud and cluster factors, (3) node factors, and (4) network factors. In the discussion, we distinguish factors that drive behavior in the basic Koala model from factors introduced to inject the dynamics, failures, and asymmetries included in the extended Koala model. After introducing the twenty input factors used for our experiments, we define the two-level values we chose for each factor included in our 2^{20-12} OFF experiment design.

3.1.1 Variable User Factors

Table 5 lists the five variable user factors (x_1 to x_5) included in our experiment. Three of these factors are holdovers from our previous experiments [1-2]. The holdovers include: the number of users (x_1), the distribution of user type probabilities (x_2), and the mean user holding time (x_3). In our previous experiments, these three input factors combined to determine demand for cloud resources. We must note, though, that in our previous experiments we assigned x_2 to be either of two, different, fixed distributions of user type probabilities, while here we assign x_2 values differently. As explained in detail below (Sec. 3.1.5), one assigned value for x_2 does define a fixed distribution of user type probabilities, while the other assigned value defines oscillating periods with differing distributions of user type probabilities. For this reason, we categorize input factor x_2 as both a holdover and a dynamics factor. Factor x_4 encompasses variables that control the ability of users to increase or decrease dynamically the number of VMs required during the holding period. Factor x_5 determines the likelihood that a user will generate ill-formed *Run Instances* requests, which the cloud cannot interpret, and thus must reject.

Table 5. Variable User Input Factors

Factor	Factor Name	Factor Category
x_1	Number of Users	Holdover
x_2	User Type Probability	Dynamics/Holdover
x_3	Holding Time	Holdover
x_4	Changes in User Demand While Holding	Dynamics
x_5	Probability of Bogus User Request	Failures

3.1.2 Variable Cloud and Cluster Factors

Table 6 lists six variable cloud and cluster factors (x_6 to x_{11}) included in our experiment. Three of these factors are holdovers from our previous experiments. The holdovers include: the algorithm (x_6) that the cloud controller uses to select a cluster on

which to allocate a set of VMs requested by a user; the absolute size (x_8) of each cluster (i.e., number of nodes per cluster); and the distribution of platform type probabilities (x_{10}) for cluster nodes. In our previous experiments, these three input factors combined to determine the supply of cloud resources. In the current experiment, we introduce a new input factor (x_9) that allows us to skew, asymmetrically, the distribution of cluster sizes, so that a few are relatively large, while most are relatively small. When skewing cluster sizes, we maintain the overall mean cluster size defined by input factor x_8 . The remaining cloud and cluster input factors control injection of cloud and cluster reconfigurations, which entail adding and removing clusters (x_7) and nodes (x_{11}).

Table 6. Variable Cloud and Cluster Input Factors

Factor	Factor Name	Factor Category
x_6	Algorithm for Choosing Cluster	Holdover
x_7	Cloud Reconfiguration	Dynamics
x_8	Absolute Cluster Size Variation	Holdover
x_9	Relative Cluster Size Variation	Asymmetries
x_{10}	Cluster Platform Type Probability	Holdover
x_{11}	Custer Reconfiguration	Dynamics

3.1.3 Variable Node Factors

Table 7 lists the three variable node factors (x_{12} to x_{14}) included in our experiment. All three factors relate to failures. One factor (x_{12}) defines the probability a node will refuse to accept a VM after having previously indicated sufficient resources existed to support the VM. A second factor (x_{13}) specifies the probability that a node will fail, and then need to be repaired or replaced. The third factor (x_{14}) describes the chances that specific cores, disks, or memory banks within a node will fail, reducing the node's capacity and requiring remedy before the node returns to full operation.

Table 7. Variable Node Input Factors

Factor	Factor Name	Factor Category
x_{12}	Probability of Node NERA	Failures
x_{13}	Probability of Node Failure	Failures
x_{14}	Probability of Node Component Failure	Failures

3.1.4 Variable Network Factors

Table 8 lists the remaining six variable input factors (x_{15} - x_{20}), which all relate to network characteristics. The first three of these factors relate to asymmetries, while the second three relate to failures. One factor (x_{15}) determines how clusters that compose a cloud will be distributed over the geographical extent available in the experiment. Clusters can be physically concentrated with the cloud controller, or can be disbursed across the globe. Two factors specify how propagation delays will be assigned to messages. Factor x_{16} defines a multiplier that can scale latencies on messages transiting between simulated geographical sites, while factor x_{17} defines a multiplier that can scale latencies on messages traveling within sites.

Table 8. Variable Network Input Factors

Factor	Factor Name	Factor Category
x_{15}	Cloud Distribution	Asymmetries
x_{16}	Variability in Inter-Site Latency	Asymmetries
x_{17}	Variability in Intra-Site Latency	Asymmetries
x_{18}	Probability of Inter-Site Message Loss	Failures
x_{19}	Probability Intra-site Message Loss	Failures
x_{20}	Probability of Cluster Communication Cut	Failures

Among the three factors relating to network failures, two factors specify the probability of loss for messages sent either between (x_{18}) or within (x_{19}) geographical sites. The final factor (x_{20}) characterizes cuts in communications between clusters and the cloud controller. Such cuts must be diagnosed and repaired before fully functional communications can be restored between the cloud controller and affected clusters.

3.1.5 Two-Level Values Chosen for Each Variable Input Factor

For each of the 20 variable input factors, we chose two values to assign for use in our experiment design. We label one value as the “Plus Level” and one as the “Minus Level”. We chose these designators to match the +1 and -1 labels typically used when formulating two-level experiment designs [9]. In our discussion, we refer to the “Plus Level” as (+) and the “Minus Level” as (-). We will discuss our chosen values in four categories, where each category is displayed in a separate table (Tables 9 through 12). We begin, in Table 9, with values for variable user factors.

We parameterized Koala (x_1) with either 1250 (-) or 2500 (+) users, which means that demand can increase due to the number of users by a factor of two. Similarly, we selected mean holding times (x_3) of either 8 (-) or 32 (+) hours, which means that demand can increase due to user holding times by a factor of four. This implies that the

highest level (+, +) combination of these factors (2500 users each holding VMs for a mean of 32 hours) increases demand eight times over the lowest level (-, -) combination (1250 users each holding VMs for a mean of 4 hours). The other two possible combinations simulate intermediate demands.

Table 9. Values Chosen for Variable User Input Factors

Factor	Plus Level	Minus Level
x1	2500 users	1250 users
x2	switching user type distribution 8 times between 3 user types (selected randomly with each switch) and 18 user types, where the probability assigned to each user type is also selected randomly	1 user type distribution with uniform probability for each of six user types: PU1, PU2, WS1, MS1, PS1, DS1
x3	32 h	4 h
x4	P(change) = 0.5, P(grow)= 0.5 range = 20 % to 50 %	No change
x5	1/50	1/1000

The distribution of user types (x2) also affects user demands. At the (-) level, we chose a uniform distribution of probabilities among six user types (from among those defined in Table 2). This means that, under the (-) level, each arriving user has a 1/6 probability of being any of the six types shown in Table 9. The situation is more complicated under the (+) level, which specifies that the probability of user types switches (eight times during a simulation run) between two different distributions. The starting distribution is determined by tossing a fair coin. The switching occurs at fixed intervals, i.e., after completing each 12.5 % of a simulation run. With each interval, the user probability distribution alternates between one consisting of only three user types and one with up to 18 user types. The possible user types include those shown in Table 2.

To assign a probability to each of the user types in a distribution, Koala first randomly orders the types in the distribution. For each type, Koala randomly (uniform) selects a value between 0 and 1. The selected probability is assigned and the number space is reduced by the value, forming a new cap that is below 1. The next probability is selected randomly (uniform) between 0 and the new cap, and assigned to the second user type. This process continues until all user types have been assigned a probability, or the new cap reaches 0.

At the (+) level, variable input factor x4 also affects user demands, but in an incremental manner. Specifically, each user has a 50 % chance of changing the number of VMs first acquired from the cloud through a *Run Instances* request. When a user does change demand for VMs, there is a 50 % chance the user adds VMs and a 50 % chance the user removes VMs. The number of VMs added or removed is randomly (uniform) selected between 20 % and 50 % of the number obtained initially. A user changes demand for VMs at most once; the change is attempted at a random time during the user's holding period. If the user cannot obtain the required additional VMs, then retries

will commence, using the same logic implemented for retrying initial requests. On average, since half of the incremental requests are increases and half decreases, these incremental dynamic changes do not increase overall user demand. On the other hand, the incremental changes cause some users to exhibit larger demands than usual, and some users to exhibit smaller demands. At the (-) level, no users express incremental changes in demand.

Factor x_5 spans two probabilities of users issuing ill-formed *Run Instances* requests. At the (-) level, one in 1000 requests are ill-formed. At the (+) level, one in 50 requests are ill-formed.

Table 10 lists the two values we chose for each of the six variable input factors related to the cloud and clusters. To select a cluster on which to place the VMs in each *Run Instances* request, the cloud controller uses either of two algorithms (x_6): percent-allocated (+) or least-full-first (-). A cloud consists of 10,000 nodes. When clusters are deployed in a uniform manner (x_9), the cloud is composed of either 10 clusters with 1000 nodes each (x_8) or 100 clusters each with 100 nodes (x_8 -). When clusters are deployed according to an 80/20 rule (x_9 -), the cloud is composed of either 8 clusters of 250 nodes and 2 clusters of 4000 nodes (x_8) or 80 clusters of 25 nodes and 20 clusters of 400 nodes (x_8 -). The platform types (x_{10}) of nodes are either: (-) all of the largest type (C22) or (+) randomly (uniform) selected from among four types (C8, C14, C18 and C22).

Table 10. Values Chosen for Variable Cloud and Cluster Input Factors

Factor	Plus Level	Minus Level
x_6	Percent Allocated	Least-Full First
x_7	Avg. Time until Reconfiguration = 720 h, $P(\text{grow}) = 0.5$, change = 20 %, limiting number of clusters within range 50 to 100 when x_8 is Minus Level and 5 to 15 when x_8 is Plus Level	No Reconfiguration
x_8	10 clusters x 1000 nodes/cluster	100 clusters x 1000 nodes/cluster
x_9	Uniform	80/20 Rule
x_{10}	Platform Types chosen randomly (uniform distribution) among C8, C14, C18, C22	All Platforms Type C22
x_{11}	Avg. Time until Reconfiguration = 168 h, $P(\text{grow}) = 0.5$, change = 20 %, limiting number of nodes per cluster with range 10 to 10,000	No Reconfiguration

No cloud reconfiguration occurs at the x_7 (-) level and no cluster reconfigurations occur at the x_{11} (-) level. Reconfigurations do occur at the (+) levels of these variables. The mean time between cloud reconfigurations is 720 h (one month). Half of the cloud reconfigurations add 20 % more clusters and half remove 20 % of existing clusters. The number of clusters is bounded within a range of either 5 to 15 (x_8) or 50 to 100 (x_8 -).

The mean time between cluster reconfigurations is 168 h (one week). Half of the cluster reconfigurations add 20 % more nodes and half remove 20 % of existing nodes. Cluster sizes are bounded within a range of 10 to 10,000 nodes.

Table 11 lists the value pairs chosen for the three node-related failure parameters. A node rejects (x_{12}) either one in 1000 (-) or one in 50 (+) *Run Instance* requests that the cluster controller allocates to the node. Nodes have a mean time between failure (x_{13}) of either 8000 h (-) or 400 h (+). Failed nodes recover after some randomly (triangular) selected time between 30 min and 24 h, with 4 h being typical. Node components exhibit the same mean time between failures (x_{14}) as the nodes themselves. The recovery profile for node components is similar to that of nodes, except that the minimum repair time is 1 h instead of $\frac{1}{2}$ h and the typical repair time is 8 h instead of 4 h.

Table 11. Values Chosen for Variable Node Input Factors

Factor	Plus Level	Minus Level
x_{12}	1 in 50	1 in 1000
x_{13}	400 h MTBF (Mean Time Between Failure) failure duration: minimum $\frac{1}{2}$ h, mode 4 h, maximum 24 h	8000 h MTBF failure duration: minimum $\frac{1}{2}$ h, mode 4 h, maximum 24 h
x_{14}	400 h MTBF failure duration: minimum 1 h, mode 8 h, maximum 24 h	8000 h MTBF failure duration: minimum 1 h, mode 8 h, maximum 24 h

Table 12 lists the value pairs chosen for the variable network input factors. We chose to have cloud elements geographically arranged on either a single site (x_{15} -) or spread out (x_{15} +) so that the cloud controller and each cluster are located on different sites. Site coordinates are chosen randomly from the available 8000-x-8000-mile grid, and distances between sites are computed based on their relative locations (x_{16} -). Inter-site distances may be multiplied by a factor of 10 (x_{16} +) . Distances between sites determine the propagation delay of inter-site messages. Messages sent within sites experience a much shorter, random (triangular) propagation delay (x_{17} -) that ranges between 100 ns and 500 ns, with a typical value of 250 ns. Intra-site delays may be multiplied by a factor of 10 with a 50 % probability (x_{17} + level).

The three remaining variable input factors define network-related failures. Inter-site messages are lost with a probability (x_{18}) of either 1 in 1000 (-) or 1 in 50 (+). The same loss probabilities are also assigned (x_{19}) to intra-site messages. The final variable input factor (x_{20}) assigns a mean time between cuts in communication between clusters and the cloud controller. Such cuts occur on average every 1140 hours (+) or every 22800 hours (-). These values mean that communication cuts can occur about every month and a half (+) or every two and a half years (-) or so. Communication cuts are repaired and restored after a randomly (triangular) selected time that varies between 2 h and 24 h, with a typical repair time of 8 h.

Table 12. Values Chosen for Variable Network Input Factors

Factor	Plus Level	Minus Level
x15	1 different site per cluster	1 site for cloud
x16	Distance x 10 for sites	Distance x 1 for sites
x17	50 % of randomly chosen delays are multiplied by 10	random (triangular) delays: minimum 100 ns, mode 250 ns, maximum 500 ns
x18	1 in 50	1 in 1000
x19	1 in 50	1 in 1000
x20	1140 h MTBF cut duration: minimum 2 h, mode 8 h, maximum 24 h	22800 h MTBF cut duration: minimum 2 h, mode 8 h, maximum 24 h

3.2 Fixed Parameters

Koala can be parameterized through many input factors, with only a subset varied in the current experiment. We selected variable input factors using two criteria: (1) those (six) previously shown to significantly influence Koala behavior and (2) those that inject dynamics, failures, and asymmetries into a simulated cloud. We judged other Koala parameters to be unrelated to these two criteria. These parameters include: timeout values and other configuration settings, physical assumptions, assumptions about user behavior, and small failures shown not to significantly influence Koala behavior. We assigned fixed values to these parameters. For example, we chose to fix the length of each run to one simulated year. We chose this value to ensure a sufficient period of observation, while bounding the required wall-clock time to be within reason.

Most of the fixed input factors, such as those related to orphan control procedures [8], were simply left at the Koala default values. Below, we identify and discuss some of the more interesting fixed input factors, so the reader has an idea of their nature, and the values we used. This brief discussion should help the reader judge the reasonableness of our choices. The presentation follows the same categories we used to discuss the variable input factors.

3.2.1 Fixed User Parameters

As shown in Fig. 2, individual Koala users exhibit cyclic behavior controlled by numerous parameter settings, which we fixed in our experiments. The key fixed user parameters and associated values are given in Table 13. Users arrive at randomly (exponential) determined times with a mean inter-arrival time (15 min here). After arriving, a user thinks (REFLECTING in Fig. 2) for a random (exponential) time (mean 15 min here) prior to submitting a *Run Instances* request (REQUESTING in Fig. 2). The user waits, for a random (log normal) finite time (mean 120 s and standard deviation 100 s here), for a response from the cloud controller. If no response arrives before that time, or if the cloud sends a NERA response, then (after another think period) the user retries (RETRYING in Fig. 2). The user selects randomly (triangular) a maximum number

retries between four and 24, with a typical value of 16. If the user exhausts those retries without obtaining VMs, then the user rests (RESTING in Fig. 2) for a random (exponential) period of time (mean 16 h) before entering another round of retries. A user will persist for a randomly (triangular) selected number of rest periods between 1 d and 7 d, with a typical value of 4 d. If the user cannot obtain VMs before exhausting all rest periods and related retry cycles, then the user leaves the system, only to be arrive as a new user.

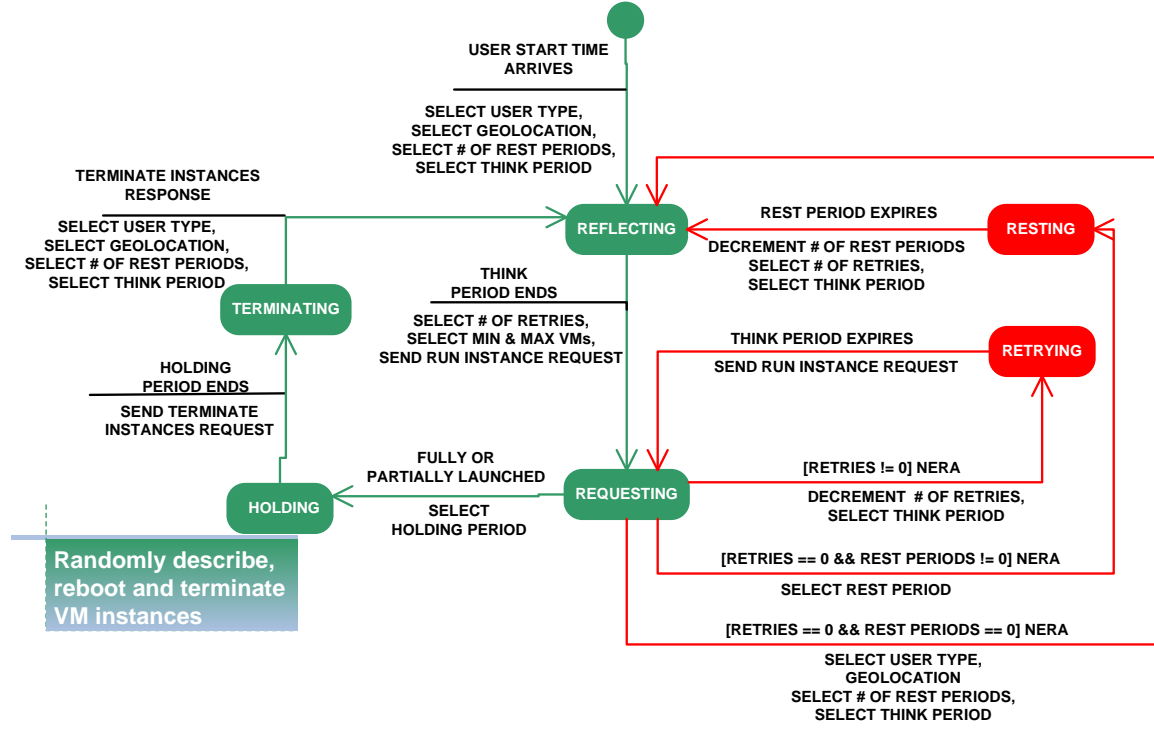


Figure 2. Finite-State Machine Representation of Koala User Behavior

If a user obtains at least the minimum requested number of VMs at any time during the request-retry-rest cycle, then the user randomly describes, reboots, and terminates obtained VM instances (HOLDING in Fig. 2) until a randomly selected holding period ends. Subsequently, the user terminates (TERMINATING in Fig. 2) the held VMs and then arrives as a new user.

3.2.2 Fixed Cloud and Cluster Parameters

The cloud control and cluster controllers have numerous configuration parameters that have default values, which we use in our experiment. Some of these parameters (see Table 14) are worth discussing here. The first concerns the maximum number of clusters that a cloud-controller will consider when attempting to allocate VMs. Some clouds may be composed of many clusters (up to 100 in our experiments). Prior to selecting a cluster for placement, the cloud controller creates an ordered list and then proceeds to make placement attempts from the top. Sometimes placements succeed and sometimes they fail, causing the cloud controller to try the next cluster on the list. In order to limit the

cloud-controller decision time, the maximum number of clusters to consider (15 here) can be restricted. In addition, the cloud controller can limit the time given for clusters to evaluate requests and respond. Here, we fixed the cloud evaluation response timeout to 90 s.

Table 13. Values for Fixed User Parameters

Fixed User Parameter	Parameter Value
Mean inter-arrival time	Exponential (15 min)
Mean think time	Exponential (15 min)
Reservation retries	Triangular (4, 16, 24)
Mean rest period duration	Exponential (16 h)
Rest periods	Triangular (1, 4, 7)
Request timeout	Lognormal (mean 120 s, std 100 s)

Table 14. Values for Fixed Cloud and Cluster Parameters

Fixed Cloud or Cluster Parameter	Parameter Value
Maximum clusters to consider	Fixed (15)
Cluster evaluation response timeout	Fixed (90 s)
Mean administrator duty cycle	Exponential (24 h)
Mean administrator activity delay	Exponential (2 m)
Heuristic for choosing nodes	First-fit
Describe resources interval	Fixed (30 min)
Node response timeout	Fixed (30 s)

When cloud or cluster controllers are unable to complete assigned actions associated with terminating VMs or with shutting down nodes or clusters, then the uncompleted tasks are assigned to an administrator, who periodically examines the work queue, and carries out any task backlog at some rate. Here, the administrator examines the task queue at randomly (exponential) selected intervals with a mean of 24 h. When

performing tasks, the administrator completes each task after a randomly (exponential) selected delay with a mean of 2 min.

In selecting nodes on which to place VMs the cluster controller uses the first-fit heuristic [1]. Periodically (every 30 min here), cluster controllers query the subordinate nodes to assess the status of resource availability in the cluster. Further, cluster controllers wait a limited time (30 s here) for nodes to respond to queries and requests.

3.2.3 Fixed Node Parameters

While most node parameters are set to Koala defaults, a few parameters (see Table 15) related to individual VMs are worth mentioning. First, VMs take random (triangular) periods to complete booting. Here, VMs boot no faster than 60 s and take no longer than 15 m; 165 s is the typical boot time. Second, a VM might be unable to boot because the boot image specified in the *Run Instance* request is not found in a cloud's image repository. Similarly, a VM image might be defective, causing the VM to crash while booting, and to be unable to run. Here, we fix the probability of each of these failures to one in a million. Finally, VMs linger on nodes for some time after being terminated. Here, we fix the VM lingering time to five minutes.

Table 15. Values for Fixed Node Parameters

Fixed Node Parameter	Parameter Value
Boot time	Triangular (60 s, 165 s, 900 s)
Instance crash probability	Fixed (10^{-6})
Image not found probability	Fixed (10^{-6})
Terminated instance linger delay	Fixed (5 min)

3.2.4 Fixed Network Parameters

Network models can be quite complicated, requiring many parameters. Table 16 identifies some key network parameters and the fixed values assigned for our experiment. Recall that Koala elements are assigned to sites, representing locations on a geographic grid. For that reason, messages communicated between sites must transit significant distances. Every mile separating a pair of sites entails a propagation delay, which we fix at $8.2 \mu\text{s}$ per 1.6093 km, which equates roughly to the time taken by a signal traveling at 65 % the speed of light. Prior to propagating, a message (composed of one or more packets of data) must be transmitted onto a medium. Here, we randomly (uniform) select the number of packets to be one to 10 per message. We fix the per-packet transmission delay to $12 \mu\text{s}$, which equates to a rate of one billion bits per second, assuming 1500-byte packets. In addition, messages transiting very long distances typically experience multiple router hops, where each hop can incur some queuing delay. We insert a router at every 1609.3 km between sites, and packets transiting each hop experience randomly

(triangular) selected queuing delays between 1 ms and 25 ms, with 10 ms being a typical value. Packets transiting between sites may also be lost, due primarily to congestion, and need to be retransmitted. Here, we randomly (uniform) select the loss probability (between 10^{-8} and 10^{-3}) for each packet. Since losses within sites are less likely, we fix intra-site packet loss probability to 10^{-12} . We limit retransmissions to at most three per packet, after which an exception is raised, which must be resolved by the message sender.

Table 16. Values for Fixed Network Parameters

Fixed Network Parameter	Parameter Value
Propagation delay per 1.6093 km	Fixed (8.2 μ s)
Per packet transmission delay	Fixed (12 μ s)
Distance between router hops	Fixed (1609.3 km)
Queuing delay per router hop	Triangular (1 ms, 10ms, 25 ms)
Inter-site packet loss probability	Uniform (10^{-8} to 10^{-3})
Intra-site packet loss probability	Fixed (10^{-12})
Maximum per packet retries	Fixed (3)
Packets per message	Uniform (1 to 10)

3.3 Responses

The Koala model is capable of producing many measurements. In our previous experiments [1-2], we evaluated up to 40 Koala responses. Here, we included 45 responses (designated y1 through y45). The number of responses increased because the extended version of Koala included several additional responses not found in basic Koala. Below, we discuss responses in four categories: (1) user, (2) cloud, (3) cluster, and (4) VM.

3.3.1 User-Level Responses

Table 17 identifies and defines the 12 responses (y1-y6 and y40-y45) that characterize user experience in our experiment. These responses include average user arrival rate (y4) and the average number of requests (y1) users submitted to obtain VMs, along with the proportion of users who gave up (y5) without obtaining VMs. Supporting responses include the proportion of requests that received NERA responses (y2) from the cloud, the proportion of arriving users who were granted some VMs (y40), and the proportion of grants that provided the maximum requested number of VMs (y3). We also measured the average grant latency (y6) for users that obtained VMs. As overall figures

of merit, we included responses measuring the ratio (y41) of granted to requested VMs, and the average cloud response time (y42) for *Run Instance* requests.

Table 17. Identity and Definition of User-Level Responses

ID	Response Name: Definition
y1	User Request Rate: Requests by All Users / # User Cycles
y2	NERA Rate: NERAs / Requests by All Users
y3	Full Grant Rate: Full Grants / (Full Grants + Partial Grants)
y4	User Arrival Rate: # User Cycles / Simulated Hours
y5	User Give-up Rate: # Users that Gave Up / # User Cycles
y6	Grant Latency: Weighted Avg. Delay in Granting VMs to Users that Got VMs
y40	User Success Rate: (Partial Grants + Full Grants)/Arrivals
y41	Proportion of Requested VMs Obtained: # VMs Granted / # VMs Requested
y42	Run Instance Response Time: Weighted Avg. of Run Instance Responses (for successful grants and NERAs)
y43	Early Termination Rate: # Users Who Terminated Early/ Total # Users who arrived
y44	Extension Giveup Rate: # Users Who Gave up on Requesting More VMs/(Total # extension requests - Total # extension request retries)
y45	Extension Grant Latency: Weighted Avg. of Acquisition Delay across all users who got additional VMs

Three responses (y43-y45) measure user success at maintaining the minimum needed number of VMs and at extending the size of VM holdings. Users terminate early if they cannot retain the minimum number of VMs needed. Response y43 reports the ratio of such users to the total number of arriving users. Users may attempt to acquire additional VMs while holding an initially granted set. The extension give-up rate (y44) reports the ratio of users who successfully acquire more VMs to the attempts made by those users. For those users who did acquire additional VMs, response y45 reports the average time delay incurred before the additional VMs were obtained.

3.3.2 Cloud-Level Responses

Table 18 identifies and defines the 12 responses that measure cloud-level performance in our experiment. The responses cover five categories: (1) cluster-related allocation performance (y7-y9), (2) resource utilization (y10-y12), (3) resource loading (y13-y15), (4) message traffic (y37-y38), and (5) revenue (y39). We cover each category in turn.

Prior to allocating VMs in a *Run Instances* request to a cluster, the cloud controller must poll clusters for availability information. Response y8 measures the average fraction of clusters offering full grants and response y9 measures the average fraction of clusters reporting NERA. The average fraction of clusters offering partial grants can be deduced from these two responses. Sometimes an allocation request fails,

and the cloud controller must choose another cluster. Response y7 measures the ratio of such reallocations to the number of full and partial grants given by the cloud.

Table 18. Identity and Definition of Cloud-Level Responses

ID	Response Name: Definition
y7	Reallocation Rate: # Times Alternate Cluster Chosen / Requests Granted
y8	Full Grant Proportion: Avg. Fraction Clusters Offering Full Grants
y9	NERA Proportion: Avg. Fraction Clusters Reporting NERA
y10	vCore Utilization: Avg. Fraction of Virtual Cores Use in Cloud
y11	Memory Utilization: Avg. Fraction of Memory in Use in Cloud
y12	Disk Space Utilization: Avg. Fraction of Disk Space in Use in Cloud
y13	pCore Load: Avg. Virtual Cores Allocated / Physical Cores in Cloud
y14	Disk Count Load: Avg. Virtual Disks Allocated / Physical Disks in Cloud
y15	NIC Count Load: Avg. Virtual NICs Allocated / Physical NICs in Cloud
y37	WS Message Rate: Avg. # WS Messages Sent Per Simulated Hour
y38	Intra-Site Messages: # WS Messages Sent within Sites / # WS Messages Sent
y39	Aggregate Revenue Per Hour: Based on 2011 EC2 prices for Windows instances in Northern Virginia

The cloud offers users virtual cores (vCores), memory, and disk space onto which VMs may be assigned. Aggregate utilization (over the entire cloud) for each of these resources is measured by responses y10-y12, respectively. Similarly, the cloud consists of physical cores, physical disks, and physical network interface controllers (NICs). As VMs are allocated to physical machines, the associated virtual cores, virtual disks, and virtual NICs are mapped to physical cores, physical disks, and physical NICs. The average number of virtual resources mapped to each of these physical resources (over the entire cloud) is measured by responses y13-y15, respectively.

Cloud operations, including interactions with users, require the exchange of Web Services (WS) messages. Response y37 reports the average number of WS messages sent per hour. Response y38 reports the proportion of WS messages exchanged within sites. The proportion exchanged between sites can be deduced from these two responses.

In Koala, the simulated cloud accrues revenue for every operating VM. Since VM costs are charged per hour, each VM operating at each hour is charged, with charges varying based on VM type (recall Table 1). Response y39 reports the average aggregate revenue (over all VMs) accumulated per hour by the cloud.

3.3.3 Cluster-Level Responses

Table 19 identifies and defines 13 responses that characterize cluster-level performance in our experiment. The responses cover three categories: (1) variance in resource utilization among clusters (y16-y18), (2) variance in resource loading among clusters (y19-y21), and (3) allocation-related performance (y22-y25), including variance among clusters (y26-y28). We cover each category in turn.

Table 19. Identity and Definition of Cluster-Level Responses

ID	Response Name: Definition
y16	vCore Util. Var: Avg. Variance in vCore Utilization across Clusters
y17	Memory Util. Var: Avg. Variance in Memory Utilization across Clusters
y18	Disk Space Util. Var: Avg. Variance in Disk Space Utilization across Clusters
y19	pCore Load Var: Avg. Variance in pCore Load across Clusters
y20	Disk Count Var: Avg. Variance in Disk Count Load across Clusters
y21	NIC Count Var: Avg. Variance in NIC Count Load across Clusters
y22	Node Reallocation Rate: # Times Alternate Node Chosen / VMs Allocated
y23	Cluster NERA Rate: # NERAs / # Responses Avg. across Clusters
y24	Cluster Full-Grant Rate: # Full Grants / # Responses Avg. across Clusters
y25	Allocation Rate: Times Cluster chosen / Cluster offered Avg. across Clusters
y26	SD-NERA: Stand. Dev. in Avg. NERA Rate across Clusters
y27	SD-Full-Grant: Stand. Dev. in Avg. Full-Grant Rate across Clusters
y28	SD-Allocation-Rate: Stand. Dev. in Allocation Rate across Clusters

Three responses (y16-y18) report variance in virtual resource (virtual cores, memory, and disk space) utilization among the clusters within the cloud. Similarly, three responses (y19-y21) report variance in physical resource (cores, disk devices, and NICs) loading among clusters. Measuring these variances provides insight into differences among the clusters.

As mentioned earlier, prior to allocating VMs to a cluster, the cloud controller polls the clusters to assess available resources, and then subsequently chooses one cluster on which to assign the VMs. Clusters may respond to such queries by saying they can accommodate all (full-grant response), some (partial-grant response) or none (NERA response) of the VMs. Response y23, cluster NERA rate, measures the ratio of NERA responses given by clusters to all responses. Similarly, response y24, cluster full-grant rate, reports the ratio of full-grant responses to all responses. The ratio of partial-grant responses can be deduced.

After receiving availability reports from all subordinate clusters, the cloud controller allocates a user *Run Instances* request to one cluster, which then maps the needed VMs to individual cluster nodes. Response y25, cluster allocation rate, reports the ratio of the number of times a cluster was selected to the number of times the cluster offered to accept a *Run Instances* request. The response is averaged across all clusters.

Three responses (y26-y28) report the standard deviation across clusters for the cluster NERA rate (y23), the cluster full-grant rate (y24), and the cluster allocation rate (y25), respectively. Measuring these standard deviations provides insight into differences among the clusters.

After being chosen by the cloud controller, a cluster controller chooses an available cluster node on which to assign each VM in a *Run Instances* request. Occasionally, a chosen node will be unable to accept a VM, which the cluster controller

must then reallocate to an alternate node. Response y22 measures the ratio of instances when an alternate node was chosen to the number of VMs allocated. This ratio is taken across all cluster controllers in the cloud.

3.3.4 Virtual-Machine-Level Responses

Table 20 identifies and defines eight responses that characterize the VMs deployed within the cloud. Response y29 reports the average number of VMs in the cloud, and the remaining seven responses (y30-y36) the average fraction of deployed VMs that are of each of the seven types available in the cloud (recall Table 1).

Table 20. Identity and Distribution of Virtual-Machine-Level Responses

ID	Response Name: Definition
y29	Current Instances: Avg. # VM Instances Extant in Cloud
y30	M1small Instances: Fraction of Current Instances that are M1 small VMs
y31	M1large Instances: Fraction of Current Instances that are M1 large VMs
y32	M1xlarge Instances: Fraction of Current Instances that are M1 xlarge VMs
y33	C1medium Instances: Fraction of Current Instances that are C1 medium VMs
y34	C1xlarge Instances: Fraction of Current Instances that are C1 xlarge VMs
y35	M2xlarge Instances: Fraction of Current Instances that are M2 xlarge VMs
y36	M4xlarge Instances: Fraction of Current Instances that are M4 xlarge VMs

4. Results and Discussion

Using two values for each of the 20 variable input factors, as explained in Sec. 3.1.5, we implemented a 2^{20-12} OFF experiment design, which required running 256 simulations, where each simulation was configured with a unique combination of values for the 20 factors. For each simulation, we collected the 45 responses described in Sec. 3.3. We organized the collected data into a 256 row x 46 column matrix, one row per simulation run, one column for the simulation run identifier (1 to 256), and one column for each of the responses (y1 to y45). This matrix served as input to our analysis, which we explain below.

We begin, in Sec. 4.1, by describing how we reduced dimensionality of the 45 responses, using a correlation and analysis technique that we have explained and applied elsewhere [2, 10]. We compare the dimensionality found by this experiment against an earlier sensitivity analysis [2], where we explored only 11 input factors and 40 responses. Subsequently, in Sec. 4.2, we gauge the influence of each of the 20 variable input factors on the 45 measured responses, as well as on the dimensions identified in Sec. 4.1. In Sec. 4.3, we rank the variable input factors from most to least influential across responses. In Sec. 4.4, we identify the influence of two-term interactions, and compare them against two-term interactions from a previous sensitivity analysis [2] of the basic Koala model. We close, in Sec. 4.5, with a summary of our findings.

4.1 Response Dimensions

We applied correlation analysis and clustering (CCA) [10] to the results data set in order to remove redundancy from the 45 responses, thus creating a lower dimensional response space, which identifies the salient, unique behaviors present in the Koala simulator. CCA begins with computation of correlation coefficients (r) between each pair of responses. Analysis of a histogram of the resulting r values enabled us to identify a threshold ($|r| > 0.65$ here) above which to retain correlation pairs. We then clustered the 124 retained pairs into mutually correlated groups that represent the main response dimensions. Table 21 reveals the 14 correlated response clusters, where each cluster represents a unique behavioral dimension exhibited by the Koala simulator. A response uncorrelated with any others appears as a singleton group (five cases in Table 21).

As a next step, we identified one response (highlighted in red, enlarged font in Table 21) to represent each dimension. We selected the response in a cluster that exhibited highest average correlation with other responses. We compared the dimensions in Table 21 with the eight dimensions found in our previous experiment [2], which considered only 11 inputs and 40 responses. Table 21 contains annotations showing the results of the comparison: \checkmark denotes a dimension also found in the previous experiment, \approx denotes a variation of a dimension found in the previous experiment, and + denotes a new dimension.

Note that seven of the eight dimensions found previously are also found in the current experiment. One dimension, Reallocation Rate, from the previous experiment divides into two (Cluster Reallocation Rate and Node Reallocation Rate) in the current experiment. A third dimension, Standard Deviation in Cluster Estimates, which was

incorporated into Variance in Cluster Load in the previous experiment, appears separated as its own dimension in the current experiment.

Table 21. Correlation and Cluster Analysis for Experiment Responses

	Response Dimension	Response Variables
✓	Cloud-wide Demand/Supply Ratio	y1, y2, y3, y5, y6, y8, y9 y11, y23, y24, y27, y30, y31, y32, y34, y35, y36, y39, y40, y41
✓	Cloud-wide Resource Usage	y10 , y13
✓	Variance in Cluster Load	y16, y17, y18 , y19, y20, y21
✓	Mix of VM Types	y11, y31, y33, y34, y35
✓	Number of VMs	y12 , y14, y15, y29, y39
≈	Cluster Reallocation Rate	y7 , y37
✓	Variance in Cluster Choice	y25 , y28
+	Extension Difficulty	y44 , y45
≈	Std. Dev. in Cluster Estimates	y2, y27 , y26
✓	User Arrival Rate	y4
≈	Node Reallocation Rate	y22
+	Fraction Intra-site Messages	y38
+	Run Instance Response Time	y42
+	Early Termination Rate	y43

✓ found in previous experiment

+ new dimension

≈ variation of dimension found in previous experiment

The results in Table 21 reveal four new behavioral dimensions, not seen in our earlier experiment. One new dimension, Run Instance Response Time, reflects how long the user must wait for the cloud controller to provide a verdict (either positive or negative) to a request for virtual machines. The related response (y42) was not included in our previous experiment, but in our current experiment the response represents a model

behavior that is independent from other behaviors. A second new dimension, Extension Difficulty, reflects the relative success of users in obtaining additional virtual machines to increase the size of the initial allocation granted by the cloud. This dimension reflects behavior that was not included in the original Koala model, but that was added for the current experiment. Similarly, a third new dimension, Early Termination Rate, reflects a behavior not included in the original Koala model, but added for the current experiment. Specifically, users now try to maintain a minimum number of virtual machines required to support their application. The Early Termination Rate reflects the proportion of users who release their virtual machines early because they could not maintain the minimum number required, either due to virtual machine crashes or cloud terminations of virtual machines due to changes in resource availability. A fourth new dimension, Fraction of Intra-site Messages, reflects the relative proportion of message exchanges within a cloud, as virtual machines are relocated due to reduction in the number of available clusters and nodes. In our previous experiment, the Fraction of Intra-site Messages was grouped together with other responses into the main dimension reflecting the ratio of cloud-wide supply and demand. The current experiment finds the Fraction of Intra-site Messages to be independent of other responses. This change occurs because the modified Koala model contains added internal processing that is used to relocate virtual machines, as clusters and nodes are removed through failures or planned shutdowns. The Fraction of Intra-site Messages varies in response to such virtual-machine relocations, which can entail the exchange of a significant number of messages within the cloud.

Overall, then, the current experiment identifies roughly the same behavioral dimensions within Koala as the previous experiment. There are a few cases where responses that had previously been grouped now appear as separate dimensions. The current experiment also found four new dimensions, all of which can be traced to new behaviors injected into the Koala model for the current experiment. In subsequent analyses related to Koala behavioral dimensions, we adopt the 14 dimensions, represented by the indicated responses, shown in Table 21.

4.2 Factor Influence on Model Responses

Next we applied main-effects analysis (MEA) [11] separately to each of the 45 responses listed in Tables 17 through 20. MEA iterates over each response, which will be represented by 256 data points, one for each parameter combination. For each selected response, MEA iterates over each of the 20 parameters, dividing the 256 data points into two groups of 128: results obtained with the parameter at the PLUS level and at the MINUS level. For each parameter, we applied a *t*-test [12] to determine whether the averages of the PLUS and MINUS level data points were significantly different, and if so at which confidence level: $p < 0.05$ or $p < 0.01$. For each parameter (x_1 to x_{20}), we computed the percent of responses influenced (Ψ), weighting $p < 0.05$ at $\frac{1}{2}$ and $p < 0.01$ at 1, as shown in the following equation:

$$\Psi = (|\{y \mid p < 0.01\}| + \frac{1}{2} |\{y \mid p < 0.05\}|) / |\{y\}| \quad (1)$$

To consider the influence of each parameter on only the behavioral dimensions shown in Table 21, we computed the proportion of dimensions influenced. Specifically,

let y_d be the response selected to represent dimension d ($= 1$ to 14), where the chosen response for each dimension is noted in Table 21. For each parameter (x_1 to x_{20}), we computed the percent of dimensions influenced (Δ), weighting $p < 0.05$ at $\frac{1}{2}$ and $p < 0.01$ at 1, as shown in the following equation:

$$\Delta = (|\{y_d \mid p < 0.01\}| + \frac{1}{2} |\{y_d \mid p < 0.05\}|) / |\{y_d\}| \quad (2)$$

Table 22 shows Ψ and Δ computed for each of the six holdover parameters, which most influenced Koala in our previous sensitivity analysis. In Table 22, and subsequent similar tables, cells are color/symbol coded: green/ $\langle \rangle$ indicates strong influence (≥ 50), yellow/ $\{ \}$ moderate influence (≥ 30 and < 50), tan/ $()$ weak influence (≥ 10 and < 30), and gray/ $[]$ little influence (< 10). Table 22 indicates that the relationship of the number (x_1) and type (x_2) of users to the type of platforms (x_{10}) provided by the cloud primarily drives the Koala model behavior, whether considering all 45 responses (Ψ) or only responses representing the 14 behavioral dimensions (Δ) of Koala. Cluster size (x_8) and user holding time (x_3), two other factors affecting the demand-supply relationship, also exhibit moderate influence. The algorithm for choosing the cluster on which to place VMs has relatively little influence on Koala behavior. These results are congruent with the results from our previous sensitivity analysis [2], where the relationship between supply and demand also was the primary driver of Koala behavior.

Table 22. Ψ and Δ Computed for the Six Holdover Parameters

	x_1	x_2^*	x_3	x_6	x_8	x_{10}
Ψ	$\langle 61 \rangle$	$\langle 60 \rangle$	(27)	(18)	$\{32\}$	$\langle 62 \rangle$
Δ	$\langle 57 \rangle$	$\langle 75 \rangle$	$\{43\}$	$[7]$	$\{43\}$	$\langle 61 \rangle$

* x_2 is both a holdover and dynamics parameter

To explore the influence of asymmetries, dynamics, and failures, we examined Ψ and Δ for the parameters within each related category. Table 23 depicts Ψ and Δ computed for the four parameters controlling asymmetries within Koala. Relative cluster size variation (x_9) is the sole parameter showing even moderate influence on Koala behavior, and that occurs only when considering all 45 responses. When considering only the 14 responses representing the Koala behavioral dimensions, the influence of cluster size variation fades to insignificance.

Table 23. Ψ and Δ Computed for the Four Asymmetry Parameters

	x_9	x_{15}	x_{16}	x_{17}
Ψ	$\{36\}$	$[4]$	$[2]$	$[0]$
Δ	$[7]$	(11)	$[7]$	$[0]$

Table 24 reports Ψ and Δ computed for the four parameters controlling dynamics within Koala. Only the user type probability (x_2) exhibits significant influence on Koala behavior. Recall, though, that user type probability is a dual parameter, which can influence both the supply-demand relationship and Koala dynamics. Our assessment is that the influence of user type probability on the supply-demand relationship is the primary driving factor. Below, after considering all four parameter categories, we will show that, whether or not we include x_2 , the set of dynamics parameters has little overall influence on Koala behavior.

Table 24. Ψ and Δ Computed for the Four Dynamics Parameters

	x_2^*	x_4	x_7	x_{11}
Ψ	<60>	[6]	(16)	[6]
Δ	<75>	[4]	(11)	(11)

* x_2 is both a holdover and dynamics parameter

Table 25 displays Ψ and Δ computed for the seven parameters controlling failures within Koala. Probability of inter-site message loss (x_{19}) is the only parameter showing even modest influence on Koala behavior. Most failure parameters show little influence. As we will demonstrate below, inter-site message losses strongly influence cluster and node reallocation rates, and also the time taken for the cloud to respond to *Run Instances* requests from users. In effect, increased loss of messages, flowing between the cloud controller and cluster controllers and between cluster controllers and node controllers, increases the rate at which the cloud believes replacement clusters must be chosen, and the rate at which cluster controllers believe replacement nodes must be chosen. This increasing rate of reallocations also drives up the time taken for the cloud controller to render a placement decision to the user.

Table 25. Ψ and Δ Computed for the Seven Failure Parameters

	x_5	x_{12}	x_{13}	x_{14}	x_{18}	x_{19}	x_{20}
Ψ	[0]	[2]	[7]	[7]	[3]	(17)	[1]
Δ	[0]	[7]	(14)	(11)	(11)	{36}	[4]

To summarize the information presented in Tables 22 through 25, we computed a simple average influence for parameters in each category: Holdovers, Asymmetries, Dynamics, and Failures. The resulting averages, taken over each category parameter on the 14 behavior dimensions (Δ) of Koala, are shown in Table 26. For the dynamics parameters, we give the average influence both with and without the dual parameter x_2 (user type probability). As seen in Table 26, the holdover parameters have the highest average influence (almost strong influence) on Koala behavior, while the other categories have little influence. This result demonstrates that the findings from our previous study

[1] of VM-placement algorithms would likely stand, even when injecting asymmetries, dynamics, and failures into Koala.

Table 26. Average Influence of Parameter Categories on Koala Behaviors

Factor Category		Avg. Δ
Holdovers		{48}
Asymmetries		[6]
Dynamics	(w/o x2)	[9]
	(w x2)	(25)
Failures		(12)

The data used to establish the significance of model parameters can also provide additional insights into Koala behavior by considering the relative effects on responses when parameter settings move from the MINUS level to the PLUS level. To measure the relative effect of a selected parameter on a specific response, we subtracted the mean response when the parameter was at the MINUS level (128 data points) from the mean response when the parameter was at the PLUS level (128 data points), and then divided that result by aggregate mean response for all 256 data points, yielding the percentage change (δ) in response due to changing the parameter level. A positive δ means that increasing the parameter level (i.e., from MINUS to PLUS) increased the response, while a negative δ means increasing the parameter level decreased the response. Table 27 gives the relative effect that each holdover parameter had on 14 responses, each representing one of the behavioral dimensions of Koala, as indicted in Table 21.

The first thing to note in Table 27 is that each cell gives a δ value representing the relative effect of a specific parameter (columns labeled under Factor) on a specific response (rows labeled under Dimension). The second thing to note is that the same color/symbol coding used in Tables 22 through 26 is retained, to indicate different strengths of relative effect. We will concentrate on cells showing large and moderate magnitude for relative effects, i.e., $\geq |31|$ %. Based on our discussion of those cells, readers will be able to interpret for themselves the cells showing lower relative effects. As our examination will reveal, a handful of parameters cause the greatest changes in model responses, and the observed directionality (increases or decreases) for those changes make logical sense. We organize our review based on response dimension (rows from top to bottom in Table 27), and we address mainly green/<> and yellow/{ } cells.

The proportion of requested VMs obtained by all users (y41) is influenced mainly by three factors. Increasing the diversity of user types (x2+), the number of users (x1+), and the diversity of platform types (x10+) all lead to decreases in the proportion of VMs obtained. A larger population of users with a range of differing VM-type needs is much easier to accommodate when the platform types are all of the largest capacity available. When platform types are divided among a more diverse set, many smaller than the largest capacity, some user types will inevitably be squeezed out because insufficient resources will be available among the set of cloud platforms on offer. Table 27 also shows that

increasing holding time ($x3+$) somewhat decreases ($\delta = -16\%$) the proportion of VMs obtained, as should be expected because when users hold VMs longer, new users will have more difficulty obtaining VMs because insufficient cloud resources are available.

Table 27. δ Computed for Holdover Parameters on the 14 Koala Behaviors

Dimension	ID	Factor					
		$x1$	$x2^*$	$x3$	$x6$	$x8$	$x10$
Demand/Supply Ratio	y41	{-40}	<-85>	(-16)	[2]	[8]	{-36}
Resource Usage	y10	[7]	[-5]	[0]	[-1]	[3]	[0]
Var. in Cluster Load	y18	[-4]	(-11)	[0]	[7]	[-8]	(-24)
Mix of VM Types	y35	[6]	<110>	[1]	[-5]	[-8]	[0]
Number of VMs	y12	[5]	(-20)	[1]	[3]	[5]	{-39}
Cluster Realloc. Rate	y7	<85>	<111>	<-64>	(-14)	<-65>	<58>
Var. in Cluster Choice	y25	{43}	{41}	[-9]	(18)	<146>	(28)
Extension Difficulty	y44	{32}	(26)	<135>	(19)	<-82>	{48}
Std. Dev. in Cluster Ests.	y27	<-75>	<-116>	(-18)	{-39}	(-24)	<-100>
User Arrival Rate	y4	<60>	(-25)	<-131>	[-2]	[-1]	[-9]
Node Reallocation Rate	y22	[-6]	(16)	(12)	[3]	[-4]	(-13)
Fraction Intra-site Msgs.	y38	(-11)	(-13)	[4]	[-1]	(21)	[-6]
Run Instance R.T.	y42	(-22)	[-8]	(-21)	(-18)	{-37}	(-21)
Early Termination Rate	y43	(20)	(23)	<138>	[-9]	[-7]	(13)

Increasing diversity of user types ($x2+$) also significantly increases y35, the proportion of VMs held that are of type M2xlarge. This occurs because when user types are less diverse ($x2-$) only one user type requires M2xlarge VMs, while four user types in the more diverse set require M2xlarge VMs.

The cluster reallocation rate (y7) increases significantly when there are more users ($x1+$) with greater diversity ($x2+$) attempting to gain access to a more diverse set of platform types ($x10+$). This combination of factors causes clusters to have greater difficulty accommodating VMs, thus initial estimates that clusters provide may prove inaccurate, as users add VMs, as nodes fail, as VMs are relocated, and so on. On the other hand, two other factors can decrease significantly the reallocation rate. First, increased user holding time ($x3+$) results in less variation in resource availability, as users who acquire VMs hold them for a much longer period. As a result, cluster estimates of resource availability tend to be more stable and thus more accurate. Second, composing the cloud from fewer, larger clusters ($x8+$) gives clusters larger safety margins when estimating resource availability, as changes in a few VM holdings will not generally invalidate an earlier estimate and thus lead to decreased reallocation rate. Small clusters have a much lower margin of safety when estimating resource availability, as taking on a few additional VMs can more easily invalidate earlier estimates.

The allocation rate (y25), i.e., the ratio of the number of times a cluster is chosen to the number of times the cluster stated sufficient availability, is mainly driven by the number of clusters. When the number of clusters is fewer (x8+) an offering cluster is more likely to be chosen. When the number of clusters is greater, an offering cluster is less likely to be chosen, simply because there are more available alternatives. Two other factors somewhat influence allocation rate. When the number of users is greater (x1+) and user types more diverse (x2+), then offering clusters are more likely to be chosen because the set of clusters that can accommodate specific user requests is likely to be fewer, meaning the number of available alternatives is lower.

The give-up rate (y44) among users attempting to acquire additional VMs during a holding period is driven primarily by longer user holding times (x3+). When users hold acquired VMs for longer periods, then the cloud has fewer resources to accommodate the need for users to increase the size of their VM holdings. Having more users (x1+) and a more diverse set of platform types (x10+) also increases extension give-up rate among users. Increasing the number of users absorbs more cloud capacity, leaving fewer resources available to permit extensions in VM holdings. A more diverse set of platform types generally decreases cloud capacity over a cloud composed of a homogeneous collection of large platforms. With fewer available resources, users have a more difficult time extending their holdings. One factor, however, can significantly decrease extension difficulty for users. A cloud composed from larger clusters (x8+) can more easily accommodate extension requests because more nodes are available on large clusters, and users needing to extend their holdings are restricted to placing the new VMs on the same cluster as their original VMs. Smaller clusters (x8-) will simply have less room to accommodate user requests for VM extensions.

Sometimes clusters can accommodate all VMs in a user request (for a so-called full grant) and sometimes clusters can only accommodate some VMs in a user request (for a so-called partial grant). Response y27 measures the standard deviation in the rate of full-grant estimates among the clusters in a cloud. Increasing the number (x1+) and diversity (x2+) of users and the diversity in platform types (x10+) all lead to significant decreases in this variance. This combination of conditions generally causes a cloud to have more difficulty accommodating user requests, leading to more cases where clusters report an inability to fully grant user VMs. This means a lower standard deviation in full-grant estimates across clusters. Another factor, cluster-selection criterion (x8), also somewhat reduces the standard deviation in full-grant estimates. In a previous study [1], we found that using the percent-allocated selection criterion (x8+) caused a cloud to be more heavily utilized. Full-grant estimates are less likely to be provided by heavily utilized clusters, leading to reduced variance in full-grant estimates.

User arrival rate (y4) is significantly influenced by only two factors. Increasing the number of users will increase the arrival rate, simply because there are more users in the simulated cloud. On the other hand, the user arrival rate will decrease when users hold VMs longer. This occurs because after liquidating VM holdings, a user is recycled as a new arrival. Longer holding times imply less frequent liquidation of VMs, which implies fewer users being recycled as new arrivals.

The *Run Instances* request response time (y42) decreases somewhat when the number of clusters is small (x8+). A cloud controller will consult up to a maximum number (15 here) of eligible clusters when attempting to place VMs for a user's request.

Under parameter setting $x8+$ the cloud consists of around 10 clusters, while under the alternative setting ($x8-$) the cloud consists of around 100 clusters. This means that generally, with only 10 clusters to consult, the cloud can give users faster responses than is the case when the maximum of 15 clusters are consulted for each user request.

Early termination rate ($y43$) is the ratio of the number of users who terminated early to the number of arriving users. A user will terminate early when that user obtains VMs from the cloud, but then cannot maintain the minimum number of VMs required for the user's intended purpose. Only user holding time ($x3$) significantly influences early termination rate. Increased holding times ($x3+$) require a user to maintain a minimum number of VMs for a longer period, which increases chances of failure to do so. Further, longer holding times increase cloud utilization, decreasing resource availability, which increases the difficulty for a user who must acquire an additional VM to replace one that has crashed or been terminated by the cloud.

Table 28 gives the relative effect that each asymmetry parameter had on the 14 selected responses that each represent one of the behavioral dimensions of Koala, as indicted in Table 21. The main point conveyed by the table is that asymmetry parameters stimulated very little change in Koala behavior. We will discuss only the yellow { } cells in Table 28. First, when clusters have uniform size ($x9+$), then the standard deviation in cluster full-grant estimates ($y27$) is substantially smaller. This stands to reason, because when cluster sizes are uniform, then full-grant estimates will be similar, while varying cluster sizes will lead to varying full-grant estimates.

Table 28. δ Computed for Asymmetry Parameters on the 14 Koala Behaviors

Dimension	ID	Factor			
		$x9$	$x15$	$x16$	$x17$
Demand/Supply Ratio	$y41$	[-1]	[-3]	[0]	[0]
Resource Usage	$y10$	[-1]	[0]	[0]	[0]
Var. in Cluster Load	$y18$	[-1]	[3]	[0]	[1]
Mix of VM Types	$y35$	[4]	[-1]	[0]	[0]
Number of VMs	$y12$	[0]	[-3]	[0]	[1]
Cluster Realloc. Rate	$y7$	(-26)	[0]	(16)	[-2]
Var. in Cluster Choice	$y25$	[-7]	(11)	[-1]	[1]
Extension Difficulty	$y44$	(18)	(15)	[2]	[1]
Std. Dev. in Cluster Ests.	$y27$	{-32}	{-36}	[-2]	[-2]
User Arrival Rate	$y4$	(13)	[7]	[0]	[-2]
Node Reallocation Rate	$y22$	[0]	[3]	[5]	[0]
Fraction Intra-site Msgs.	$y38$	[-1]	{-40}	[0]	[0]
Run Instance R.T.	$y42$	[8]	(12)	{36}	[3]
Early Termination Rate	$y43$	(-24)	[-5]	[6]	[-5]

A less obvious factor, distribution of cluster locations (x_{15}), also influences standard deviation in cluster full-grant estimates. When clusters are distributed to independent Internet sites (x_{15+}), then the standard deviation in full-grant estimates is reduced. Note that when the cloud controller communicates with cluster controllers across the Internet, then those message exchanges take more time than when cloud and cluster controllers communicate locally. On the other hand, the cloud controller will wait only so long to hear from cluster controllers. Those cluster controllers that can offer full grants take relatively little time to determine that status, and so respond to the cloud controller quickly. Cluster controllers that can offer only partial grants take more time to determine that status, and so respond to the cloud controller less quickly. Apparently, adding extra Internet delays to these latter cluster controller replies increases the likelihood that the cloud controller will have moved on with its decision making before the replies arrive. Under such circumstances, more of the replies received by the cluster controller will be full-grant estimates.

Table 28 also shows that when clusters are distributed to independent Internet sites (x_{15+}) then the fraction of intra-site messages (y_{38}) will be reduced because messages between the cloud and cluster controllers will cross the Internet rather than remain within a single site. Also as expected, when Internet communication distances are increased by an order of magnitude (x_{16+}), *Run Instances* request response times (y_{42}) increase significantly because message exchanges take longer between users and the cloud controller.

Table 29 gives the relative effect that each dynamics parameter had on the 14 selected responses that each represent one of the behavioral dimensions of Koala, as indicted in Table 21. The main point conveyed by the table is that, aside from factor x_2 (user type probability), dynamics parameters stimulated relatively little change in most of the Koala response dimensions. Recall that factor x_2 is both a holdover and dynamics factor, and that we already discussed related results above, when discussing the relative effects of holdover parameters. Here, we will discuss only the green $\langle \rangle$ and yellow $\{ \}$ cells under factors x_4 , x_7 , and x_{11} . First, note that when the cloud can be reconfigured (x_{7+}) then the variance in cluster load increases significantly. This occurs because cloud reconfiguration can add and remove clusters. When clusters are added, they start empty, which increases the variance in load among clusters. When clusters are removed, then VMs must be relocated to other clusters, or else terminated. Clusters receiving relocated VMs become more heavily loaded, which increases the variance in load among clusters. The cluster reconfiguration process also has a subsidiary effect on the standard deviation in estimates that clusters make on their ability to offer full grants (y_{27}). Higher variance in cluster loads should be reflected in higher variance in the ability of clusters to offer full grants.

Table 29 also shows that when clusters can be reconfigured (x_{11+}) then it becomes more difficult for individual users to extend their holding of VMs (y_{44}). Cluster reconfiguration can add and remove nodes. When nodes are added, cluster capacity increases, which attracts demands from more users. As the increased capacity is filled by newly arriving users, there are fewer available nodes to serve existing users who wish to increase the number of VMs that they hold. On the other hand, removing nodes from a cluster means that VMs on those nodes must be relocated or terminated. Successfully relocated VMs absorb demand on new clusters, making it difficult for existing users to

find nodes on which to increase the number of VMs they hold. Whether or not VMs can be relocated successfully, removing nodes from a cluster lowers capacity and makes it more difficult for existing users to extend their VM holdings. The cluster reconfiguration process also has a subsidiary effect on the ability of users to maintain the minimum number of VMs required to remain viable. This shows up as an increase in the proportion of users that must terminate early (y43).

Table 29. δ Computed for Dynamics Parameters on the 14 Koala Behaviors

Dimension	ID	Factor			
		x2*	x4	x7	x11
Demand/Supply Ratio	y41	<-85>	[-2]	[-1]	[-3]
Resource Usage	y10	[-5]	[1]	[0]	[1]
Var. in Cluster Load	y18	(-11)	[5]	<108>	[1]
Mix of VM Types	y35	<110>	[-7]	[1]	[3]
Number of VMs	y12	(-20)	[2]	[0]	[0]
Cluster Realloc. Rate	y7	<111>	{35}	[5]	[4]
Var. in Cluster Choice	y25	{41}	[2]	[0]	[6]
Extension Difficulty	y44	(26)	(29)	(-15)	<76>
Std. Dev. in Cluster Ests.	y27	<-116>	[4]	{30}	[-7]
User Arrival Rate	y4	(-25)	[-1]	[-2]	[3]
Node Reallocation Rate	y22	(16)	[-2]	[-1]	[-1]
Fraction Intra-site Msgs.	y38	(-13)	[-1]	[-1]	[-1]
Run Instance R.T.	y42	[-8]	[-4]	[-9]	[-9]
Early Termination Rate	y43	(23)	[-4]	[-8]	{30}

Table 29 also shows some increase in the cluster reallocation rate (y7) when users can change their demand profile while holding VMs (x4+). Increasing demands from existing users apparently can lead to an increase in the number of invalid estimates made by clusters. Here, the absolute increase is from a rate of 12% reallocations (x4-) to a rate of 17% (x4+), a change that is fairly small. We could determine no explanation for this effect, which might disappear under additional replications of the experiment. Further investigation is needed to determine whether this effect is real, and what the cause might be.

Table 30 gives the relative effect that each failure parameter had on the 14 selected responses that each represent one of the behavioral dimensions of Koala, as indicated in Table 21. The main point conveyed by Table 30 is that failures had limited influence on Koala behavior. Node (x13) and component (x14) failures, along with message losses between (x18) and within (x19) sites, had some influence. Here, we will discuss only the green <> and yellow {} cells in Table 30.

Among the failure parameters, intra-site message loss (x_{19}) had the most influence on Koala behavior. First, increasing intra-site message loss (x_{19+}) increased the need to reallocate VMs among clusters (y_7) and nodes (y_{22}). This occurred largely because either initial allocation messages were lost or else replies to the initial allocation messages were lost. In either case, VMs had to be reassigned to a different cluster or node, depending on which allocation was affected by a lost message. As a result, additional allocation messages and responses were needed, which caused the *Run Instance* response time (y_{42}) to also increase. Inter-site message losses (x_{18+}) had similar, but muted, effect on cluster reallocation (y_7). The effect occurred in cases where clusters were located on sites independent from the cloud controller, requiring that messages between the cloud controller and cluster controllers transited between sites.

Table 30. δ Computed for Failure Parameters on the 14 Koala Behaviors

Dimension	ID	Factor						
		x_5	x_{12}	x_{13}	x_{14}	x_{18}	x_{19}	x_{20}
Demand/Supply Ratio	y_{41}	[1]	[-1]	[-4]	[-1]	[1]	[2]	[-1]
Resource Usage	y_{10}	[0]	[1]	[1]	[0]	[0]	[-2]	[0]
Var. in Cluster Load	y_{18}	[0]	[-3]	[4]	[3]	[2]	[0]	[-1]
Mix of VM Types	y_{35}	[0]	[0]	[0]	[-2]	[-1]	[-6]	[0]
Number of VMs	y_{12}	[1]	[0]	[1]	[1]	[-1]	[-2]	[0]
Cluster Realloc. Rate	y_7	[0]	[-2]	[1]	{-37}	{31}	<72>	(24)
Var. in Cluster Choice	y_{25}	[-5]	[-3]	[-3]	[6]	[0]	[-7]	[-5]
Extension Difficulty	y_{44}	[7]	[0]	<-95>	<-51>	(-18)	{-36}	[1]
Std. Dev. in Cluster Ests.	y_{27}	(11)	(19)	(-14)	[3]	[0]	[6]	[-4]
User Arrival Rate	y_4	[0]	[2]	[8]	[4]	[-1]	[0]	[1]
Node Reallocation Rate	y_{22}	[0]	{35}	[1]	[-2]	[4]	<73>	[4]
Fraction Intra-site Msgs.	y_{38}	[0]	[0]	[2]	[-2]	[1]	[0]	[0]
Run Instance R.T.	y_{42}	[4]	[9]	[-3]	[-5]	(25)	<66>	(18)
Early Termination Rate	y_{43}	[-3]	[5]	<59>	(19)	(26)	(29)	(16)

Increased intra-site message losses (x_{19+}) also reduced the difficulty users had in obtaining additional VMs (y_{44}). This likely occurred because intra-site message losses interfered with the ability of users to obtain initial allocations, which made more space available to support extension requests from users who did obtain initial allocations. Extension difficulty (y_{44}) was also reduced under increased node (x_{13+}) and component (x_{14+}) failures. This likely occurred because the early termination rate (y_{43}) increased, meaning that users had more difficulty maintaining the minimum number of VMs required for viability. Subsequently, affected users terminated their VM holdings early, leaving more space in the cloud for unaffected users that wished to increase their VM holdings.

Table 30 reveals two remaining moderate effects from failures. First, when nodes exhibited an increased rate of issuing NERA (x_{12+}) after promising to accept a VM, then node reallocation rate (y_{22}) increased. This effect is as expected. When a node reneged on a promise to accept a VM, then the VM is assigned to a different node. Second, when the component failure rate increased (x_{14+}), the cluster reallocation rate (y_7) fell. Here, the absolute decrease is from a rate of 17 % reallocations (x_{12-}) to a rate of 12 % (x_{12+}), a change that is fairly small. We could conceive no explanation for this effect, which might disappear under additional replications of the experiment. Further investigation is needed to determine whether this effect is real, and, if so, what the cause might be.

4.3 Factor Ranking

To understand the degree to which each input factor influenced the overall behavior of Koala, we ranked the input factors based on decreasing percent of behavioral dimensions influenced (Δ), taken from Tables 22 through 25. We plot the factor ranking as Table 31, where the 20 input factors group into 10 ranks.

Table 31. Ranking of Input Factors Based on Decreasing Δ

Rank	Δ	Factor	Factor Name
1	<75>	x_2	User Type Probability
2	<61>	x_{10}	Cluster Platform Type Probability
3	<57>	x_1	Number of Users
4	{43}	x_3 x_8	Holding Time Absolute Cluster Size Variation
5	{36}	x_{19}	Probability Intra-site Message Loss
6	(14)	x_{13}	Probability of Node Failure
7	(11)	x_7 x_{11} x_{14} x_{15} x_{18}	Cloud Reconfiguration Cluster Reconfiguration Probability of Node Component Failure Cloud Distribution Probability of Inter-Site Message Loss
8	[7]	x_6 x_9 x_{12} x_{16}	Algorithm for Choosing Cluster Relative Cluster Size Variation Probability of Node NERA Variability in Inter-Site Latency
9	[4]	x_4 x_{20}	Changes in User Demand While Holding Probability of Cluster Communication Cut
10	[0]	x_5 x_{17}	Probability of Bogus User Request Variability in Intra-Site Latency

We draw two main impressions from Table 31. First, even considering the increase in behavioral dimensions to 14 from eight in our previous sensitivity analysis

[2], Koala behavior is influenced mainly by five of the six holdover input factors (x_2 , x_{10} , x_1 , x_3 , and x_8). One holdover input factor, the algorithm for choosing a cluster (x_6) on which to place virtual machines, exhibited little influence on Koala behavior. This same factor was least influential among the holdover factors in our previous sensitivity analysis. Second, only one new input factor, probability of intra-site message loss (x_{19}), exhibited even moderate influence on Koala behavior. We had previously found that message losses, both within and between sites, could lead to VM orphans within a cloud [8]. In the current experiment, orphan-control procedures were implemented within our Koala model, so intra-site message loss was influencing other Koala behaviors.

After reviewing the 14 Koala behavioral dimensions, we found that intra-site message loss (x_{19}) influenced, to varying degrees, six dimensions, which reveal an integrated picture. Increasing intra-site message loss rate (x_{19+}) interfered with a cloud's ability to perform cloud resource-allocation procedures, increasing the need to reallocate VMs to clusters (y_7) and to nodes (y_{22}). Such reallocations involved repeated message exchanges among cloud components, which led to increased *Run Instance* response times (y_{42}). Sometimes, reallocation procedures failed, due to repeated intra-site message losses, and VMs could not be allocated, which was reflected in an increased early termination rate (y_{43}), as some users were unable to maintain the minimum number of VMs required for viability. This led to a decrease in cloud-wide resource usage (y_{10}), which allowed some users to achieve better success in extending (y_{44}) their collection of VMs.

While increased intra-site message loss rate had a moderate influence on selected Koala behavioral dimensions, from our analysis we concluded that such influence appears likely to affect equally any of the cluster-assignment algorithms we investigated in our earlier experiment [1]. This means that rerunning those experiments while including intra-site message loss is unlikely to change the results and conclusions. Table 31 shows that the remaining (13 of 14) input factors added to inject asymmetries, dynamics, and failures had little influence on Koala behavior.

4.4 Two-Term Interactions

Sometimes, two-term interactions can have a significant influence on model behavior. In our previous study [2], we investigated two-term interactions but omitted reporting them. We did not report two-term interactions because we had limited space for publication. In our previous study, while only about 0.8 % of possible two-term interactions proved statistically significant, about 75 % (25) of those interactions exhibited magnitudes greater than main effects for the same response. The two-term interactions, however, influenced only 14 responses. Further, when considered across all responses, the magnitude of statistically significant main effects generally much exceeded the magnitude of interactions. Here, we introduce two-term interactions from our previous study, and compare them to interactions from the current study.

Figure 3 depicts a half matrix, summarizing two-term interactions from our previous sensitivity analysis [2]. For convenience, we include a table identifying an identifier (x_1 to x_{11}) for each the eleven input factors used. These identifiers relate to the integers across the top (and also down the right side) of the half matrix (i.e., $x_1 = 1$, $x_2 = 2$, and so on to $x_{11} = 11$). Excluding the diagonal, each cell in the half matrix reports

interactions between two, intersecting, input factors. In each cell, two-term interactions are reported in two classes: the count of those with statistical significance $p < 0.01$, followed by the count of those with statistical significance $p < 0.05$. Where no two-term interactions exist, a cell is left blank. Where present, a count refers to the number of the 40 responses (from Table V in [2]) for which a statistically significant two-term interaction was found. The count of interactions where $p < 0.01$ is listed first, followed by a slash, and then the count of interactions where $p < 0.05$. For example, the cell in row one, column eleven (contents are 2/1), shows that simulation duration and probability of packet loss interacted at $p < 0.01$ on two responses and at $p < 0.05$ on one response. Each cell on the diagonal sums all interactions, in each significance class, for the factor in the indicated column. For example, the first cell on the diagonal shows that simulation duration (x_1) had two interactions with other factors at $p < 0.01$ and six interactions at $p < 0.05$. Cells along the diagonal are color-coded to indicate input factors with more (blue) and fewer (gray) two-term interactions.

Figure 3. Two-term Interactions from a Previous Study [2]

		1	2	3	4	5	6	7	8	9	10	11	
		2/6			0/3	0/2						2/1	1
			4/2		0/1			2/0			2/1		2
				1/6	1/0	0/3		0/3					3
Factor	Definition				1/5						0/1		4
x1	Simulation Duration												
x2	Number of Users					1/8	0/1	1/1			0/1		5
x3	User Type Probability						0/5	1/4					6
x4	Average Holding Time							4/8					7
x5	Number of Clusters								0/1	0/1			8
x6	Nodes per Cluster									0/2		0/1	9
x7	Platform Type Probability										2/2		10
x8	Cluster-Assignment Algorithm											2/2	11
x9	Node-Assignment Algorithm												
x10	Number of Sites for Cloud Components												
x11	Probability of Packet Loss												

Figure 3 shows that only nine two-term interactions appeared at $p < 0.01$ and 24 appeared at $p < 0.05$. The largest number of interactions (12) involved platform type probability (x_7). Overall, the number of two-term interactions was quite limited, especially at the higher level of statistical significance.

Table 32 summarizes two-term interactions for cases where the magnitude of the interaction exceeded the magnitude of the main effect for individual response variables in our previous sensitivity analysis [2]. (This summarization is independent of whether or

not the corresponding main effect was statically significant or not.) Though 14 different responses were influenced by two-term interactions, only four of the eight behavioral dimensions were influenced. Below, we provide a general impression of the influence of the interactions on each response dimension.

Table 32. Large, Significant Two-Term Interactions from Previous Study [2]

Interaction*	Count	Responses Influenced	Behavioral Dimension
x5-x7	6	y16, y18, y19, y20, y21, y28	Variance in Cluster Load
x3-x4	4	y16, y19, y27, y28	Variance in Cluster Load
x1-x8	3	y16, y19, y27, y28	Variance in Cluster Load
x1-x11	2	y7, y22	Reallocation Rate
x3-x7	2	y11, y28	Resource Usage
x4-x5	2	y11, y28	Resource Usage
x1-x6	1	y3	Demand-Supply Ratio
x2-x10	1	y7	Reallocation Rate
x3-x6	1	y11	Resource Usage
x4-x2	1	y11	Resource Usage
x4-x7	1	y36	Demand-Supply Ratio
x6-x10	1	y26	Variance in Cluster Load
x9-x11	1	y26	Variance in Cluster Load

*Here, variable identifiers are taken from Fig. 3 and behavioral dimensions are taken from [2]

As shown in Table 32, variance in cluster load was influenced by a number of two-term interactions. First, the number of clusters (x_5) interacted with platform types (x_7) to influence variance in load among clusters. Similar influence was exhibited by interaction between user types (x_3) and average holding time (x_4). These interactions influence supply-demand relationships, which can naturally influence variance in load among clusters. Second, the interaction between simulation duration (x_1) and cluster assignment algorithm (x_8) suggests that one of the cluster assignment algorithms leads to greater variance in cluster load over longer simulated times. Third, one of interactions, nodes per cluster (x_6) and number of sites (x_{10}), influences NERA rate across clusters. Similar influence arises from interaction between node-assignment algorithm (x_9) and packet-loss rate (x_{11}), though this interaction is suspect because node assignment is an intra-site operation, and packet-loss rate affects only inter-site message exchanges.

Table 32 also suggests that resource usage is influence by several two-term interactions. First, user types (x_3) interact with platform types (x_7) to influence resource usage. This should be expected, as well as the fact that this demand-supply interaction has a primary influence on Koala behavior. Similar demand-supply interactions arise between user holding time (x_4) and the number of clusters (x_5) and between user types (x_3) and nodes per cluster (x_6). The interaction between two demand-related factors,

number of users (x_2) and user holding time (x_4), is also unsurprising. This group of interactions influences the relationship between demand and supply, which is the major factor driving Koala behavior.

Table 32 also shows some other two-term interactions of interest. First, the average holding time (x_4) interacts with platform types (x_7) to influence the proportion of M4xlarge instances within the cloud. This makes sense because only large platform types can hold M4xlarge instances, and so, when users obtaining such instances tend to hold them longer, then more such instances will be extant. Second, the full grant rate is influenced by an interaction between the length of the simulation (x_1) and the number of nodes per cluster (x_6). While such an interaction seems plausible, the nature of the interaction requires further investigation. Finally, a pair of two-term interactions influences the reallocation rate. The number of users (x_2) and number of sites for clusters (x_{10}) interact to influence cluster reallocation rate. This makes sense because cluster reallocations tend to occur more when the cloud is distributed among multiple sites, and the number of users influences the number of opportunities for allocation decisions to be made. An interaction between simulation duration (x_1) and packet-loss probability (x_{11}) also influences reallocation rate. This makes sense because a longer simulation gives more opportunity for packet losses to interfere with allocation decisions.

Figure 4 reports a half matrix similar to Fig. 3, but showing two-term interactions among the 20 input factors (listed in Tables 5 through 8) in the current sensitivity analysis. In Fig. 4, the counts refer to two-term interactions with respect to the 45 responses listed in Tables 17 through 20. In comparing Fig. 3 with Fig. 4, remember that Fig. 4 includes five additional responses, not possible in Fig. 3, and 20 input factors rather than 11. Figure 4 also has one additional feature not included in Fig. 3. Several cells in Fig. 4 are highlighted in beige to identify specific two-term interactions that occur more frequently than others.

Comparing Fig. 3 with Fig. 4 gives a general impression that the current study exhibited more two-term interactions than the previous study. Of course, this could be due in part to the fact that the current study has nine additional input factors and five additional responses. To compare the two figures on a fair basis, we computed the percentage of possible two-term interactions portrayed on each figure. Figure 3 shows 0.8 % of all possible two-term interactions. Figure 4 shows 2.9% of all possible two-term interactions. This confirms the general impression that the current study revealed more two-term interactions.

Comparing two-term interactions in more detail between the previous and current sensitivity analyses requires some care because the input factors intersect but do not overlap completely. Each study has factors that cannot be mapped directly to factors in the other. After due consideration, we were able to map (see Table 33) eight of the 11 input factors from the previous study [2] to input factors in the current study. This means that three input factors (simulation duration, node choice, and packet-loss probability) from the previous study have no analog in the current study. Similarly, 12 input factors (i.e., those omitted from Table 33) in the current study have no analog in the previous study.

As we consider two-term interactions in the current study, some of the interactions can be equated loosely to interactions in the previous study (we call this the MM group), but there are also two other classes of interactions: (1) those between factors

mapped from the previous study and factors not mapped (the MU group) and (2) those between factors not mapped from the previous study (the UU group). We will need to consider separately each of these three classes of two-term interactions.

Figure 4. Two-term Interactions from the Current Study

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
61/33	17/3	2/6	15/4		0/2		1/2	5/2	7/6		3/1		3/3	1/0	2/0		3/0	2/2		1
	57/31	5/2	7/7		3/1	0/1	3/4	2/5	15/4		0/2		1/2	1/0				3/0		2
		48/28	1/0				3/1	16/4	1/0	2/0		3/0	2/2	7/7		3/1	0/1	3/4		3
			52/32		0/2	1/0	2/0	1/0	15/4		0/1	3/0	1/3	5/2				1/0		4
				24/7		0/1		2/0		16/4					5/2				1/0	5
					24/14		0/1		0/1		16/4					5/2		0/1		6
						27/9	0/1	2/0	3/0			16/4					5/2			7
							34/20	2/2	1/3				16/4	1/0				5/2		8
								50/21	1/0				3/1	15/3		0/2		1/2		9
									51/24		0/2	1/0	2/0	3/4					2/0	10
										22/9		0/1			3/4				1/0	11
											21/16		0/2			1/6		1/0		12
												24/13	0/2			0/1	1/5			13
													31/25	0/1		1/0		2/5		14
														36/19		0/2	1/0	2/0		15
															17/13		0/1		7/6	16
																10/16		0/2		17
																	10/11	0/2		18
																		21/20	1/1	19
																			12/8	20

Table 33. Mapping of Eight Input Factors from the Current Study to Previous Study [2]

Factor	Current Study	Previous Study
Number of Users	x1	x2
User Type	x2	x3
Holding Time	x3	x4
Number of Clusters	x8	x5
Nodes per Cluster	x9	x6
Platform Type	x10	x7
Cluster-Selection Algorithm	x6	x8
Cloud Site Distribution	x15	x10

In our current study, while only 2.9 % of possible two-term interactions proved statistically significant, about 66 % (i.e., 317) of those interactions exhibited magnitudes greater than main effects for the same response. (Despite this, considering all responses, the magnitude of statistically significant main effects generally much exceeded the magnitude of two-term interactions, as was also the case in our previous study.) Among the 317 interactions that we considered, 93 (30 %) involved pairs in the MM group: where both factors could be mapped between our previous and current experiments; 165 (52 %) involved interactions in the MU group: where one factor could be mapped between experiments and the other could not; the remaining 59 (18 %) involved interactions in the UU group: where neither factor could be mapped between experiments. From this, we concluded that input factors causing asymmetries, dynamics, and failures (MU and UU groups) were involved in most (70 %) of the significant, two-term interactions seen in the current experiment. And most (74 %) of those interactions (the MU group) involved one factor that could be mapped from the previous study.

Table 34 lists two-term interactions in the MM group, sorted in decreasing order of number of responses influenced. We list only the top 10 (of 18) interactions. Though the factors involved in the MM group can all be mapped to factors from our previous experiment, most of the interactions are quite different. There is a main subgroup of five two-term interactions involving relative cluster size (x9), an asymmetry parameter that can be set to uniform cluster sizes or to an 80/20 rule, where a few clusters are very large and most are very small. Apparently, this factor interacts with numerous other factors, including holding time (x3), cloud distribution (x15), number (x1) and types (x2) of users, and absolute cluster size (x8). Further, these interactions can influence many responses. A secondary subgroup of interactions involves the number of users (x1), which interacts with platform types (x10), holding time (x3) and user types (x2). This subgroup is similar to interactions seen in our previous experiment. The remaining two MM interactions are singletons. Of note, holding time (x3) interacts with cloud distribution (x15) to influence 12 responses, all of which reflect the demand-supply ratio. Apparently, the influence of holding time differs depending upon whether a cloud is centralized or distributed. This

interaction did appear in our previous study, but a distributed cloud in the current study is significantly more distributed than in our previous study. The final MM interaction involves absolute cluster size (x8) and cluster assignment algorithm (x6). This interaction did not appear in our previous study, but the current study varies cluster size (and also number of clusters) by an order of magnitude (vs. two to six times in our previous study).

Table 34. Top 10 Two-term Interactions in the MM Group

Interaction	Count	Responses Influenced
x3-x9	21	y1, y2, y5, y7, y9, y10, y11, y13, y17, y23, y24, y27, y30, y32, y33, y36, y37, y40, y41, y43, y44
x9-x15	16	y1, y2, y5, y7, y11, y12, y17, y23, y24, y30, y32, y33, y35, y39, y40, y41
x3-x15	12	y6, y10, y13, y14, y15, y24, y26, y28, y29, y31, y34, y36
x1-x9	7	y1, y3, y5, y6, y7, y31, y40
x2-x9	5	y4, y5, y26, y27, y40
x1-x10	5	y13, y26, y28, y31, y34
x1-x3	4	y15, y24, y26, y27
x1-x2	3	y11, y43, y44
x6-x8	3	y10, y13, y27
x8-x9	3	y15, y42, y44

Table 35 lists two-term interactions in the MU group, sorted in decreasing order of number of responses influenced. We list only the top eight (of 35) interactions. The interactions can be further divided into two main subgroups, related to: (1) failures and (2) changes in user demand while holding VMs (x4). One should expect failures to interact with other factors. Table 35 highlights various failure types: component failures (x14), intra-site message loss (x19), and probability of Node NERA (x12). The second subgroup illustrates that changes in user demand (x4) interacts with user (x2) and platform (x10) types, absolute cluster size (x8), and cloud distribution (x15). These interactions are not surprising. There is one singleton interaction involving cluster assignment algorithm (x6) and intra-site latency (x17). This suggests that increase in intra-site latency has greater influence on one of the two cluster assignment algorithms. Further analysis is needed to discern the nature of the influence.

Table 36 lists two-term interactions in the UU group, sorted in decreasing order of number of responses influenced. We list only the top six (of 11) interactions. The interactions can be further divided into three subgroups: (1) bogus user requests, (2) cloud reconfiguration, and (3) changes in user demand. Each subgroup contains a pair of interactions. We discuss each subgroup in turn.

Table 35. Top Eight Two-term Interactions in the MU Group

Interaction	Count	Responses Influenced
x4-x10	20	y1, y2, y5, y7, y9, y10, y11, y13, y17, y23, y24, y27, y30, y31, y33, y37, y40, y41, y43, y44
x6-x12	20	y1, y2, y5, y7, y10, y11, y13, y17, y23, y24, y27, y30, y31, y33, y36, y37, y40, y41, y42, y43
x1-x14	19	y1, y2, y5, y6, y7, y9, y11, y12, y17, y23, y24, y27, y30, y31, y33, y36, y39, y40, y41
x8-x14	15	y1, y2, y5, y9, y11, y23, y24, y27, y30, y31, y33, y36, y40, y41, y43
x2-x4	14	y1, y6, y7, y8, y10, y13, y14, y15, y24, y26, y28, y29, y31, y34
x4-x15	6	y1, y4, y5, y7, y18, y40
x6-x17	6	y1, y5, y6, y7, y31, y40
x8-x19	6	y1, y4, y5, y6, y31, y40

Table 36. Top Six Two-term Interactions in the UU Group

Interaction	Count	Responses Influenced
x5-x11	18	y1, y2, y5, y7, y9, y10, y13, y17, y23, y24, y30, y32, y33, y36, y37, y41, y43, y44
x7-x13	18	y2, y5, y7, y9, y10, y11, y13, y23, y24, y27, y30, y32, y36, y37, y39, y40, y41, y44
x5-x16	7	y1, y4, y5, y6, y7, y31, y40
x7-x18	4	y4, y5, y7, y40
x4-x13	3	y18, y20, y21
x4-x14	3	y7, y20, y37

Unsurprisingly, the probability that a user issues an ill-formed request (x5) interacts with variability in inter-site delay (x16). All user requests transit between sites, across the Internet. Ill-formed requests are rejected by the cloud, requiring the user to retry the operation. So naturally, longer inter-site delays tend to lengthen the retry cycle, slowing the rate at which user requests arrive. Perhaps unexpectedly, ill-formed user

requests also interact with cluster reconfiguration (x_{11}). We surmise that an increased rate of ill-formed user requests spreads out the load of user operations on the cloud. This would then allow more time to complete cluster reconfigurations than might otherwise be available. Further analysis is required to confirm this explanation.

Cloud reconfiguration (x_7) interacts with node failure (x_{13}) and with inter-site message loss (x_{18}). Node failures require clusters to relocate VMs. Similarly, when removing clusters the cloud controller must also relocate VMs. The combination of these circumstances seems likely to increase the difficulty of relocating VMs. Confirming this explanation requires further analysis. When a cloud is distributed across multiple sites, inter-site messages are required for communication between the cloud controller and cluster controllers. In this case, inter-site message losses can interfere with cloud reconfigurations.

Finally, dynamic changes in the number of VMs required by users (x_4) interacts with both node (x_{13}) and component (x_{14}) failures. Such interactions should be anticipated. When a user needs to acquire additional VMs (increasing demand for cloud resources), failures of nodes and components (reducing available supply) would interfere. Confirming this explanation requires further analysis.

4.5 Findings

We can sum up our analysis through six findings. First, in the current study, Koala exhibited all of the behavioral dimensions found in our previous sensitivity analysis. We found a few minor variations, all of which were easily explainable, leading the eight dimensions from our previous study to appear as ten dimensions here. The current study also introduced four new behavioral dimensions. Three of these new dimensions arose from added user behaviors, which were not included in Koala during our previous study. The final new dimension (fraction of intra-site messages) arose because the current study ensured that each cluster was placed on a unique site, when distributing clusters on the Internet. The scope of distribution was not as large in our previous study. Thus we concluded that injecting asymmetries, dynamics, and failures into Koala did not fundamentally alter the behavioral dimensions found in our previous sensitivity analysis, though four behavioral dimensions were added.

Second, we found that the same primary factors drove Koala behaviors in both our current and previous studies. This was confirmed because the top five (of six) holdover factors, the most influential from the previous study, were most influential in determining Koala behavior in the current study. Only one new factor, intra-site message losses, showed even moderate influence on Koala behavior in the current study. And the influence of intra-site message losses fell below the influence of the top five factors from the previous study.

Third, though intra-site message losses showed moderate influence in Koala behavior during the current study, we concluded that this factor would be unlikely to influence choice of algorithm for assigning clusters and nodes in the cloud. We reached this conclusion based on the observation that intra-site message losses would interfere equally with all such algorithms. Further supporting this conclusion, our current study revealed no significant interaction between intra-site message loss and cluster allocation algorithm.

Fourth, the influence of main effects outweighs the influence of two-term interactions in both our current and previous studies. We reached this conclusion because statistically significant main effects showed magnitudes that far exceeded most two-term interactions. The magnitude of two-term interactions exceeded main effects mostly in cases where the main effects were not statistically significant, and not of large magnitude.

Fifth, the current study exhibited a substantial increase in two-term interactions over the previous study. This conclusion holds even when accounting for the increased number of responses and input factors in the current study. In the current study, two-term interactions could be classified into three groups: (1) interactions between factors that could be mapped between both experiments (MM), (2) interactions where one of the two factors could be mapped (MU), and (3) interactions where neither factor could be mapped (UU). Clearly, MU and UU interactions could not occur in our previous study. Even in the case of MM interactions, most of those found in our current study were different from those found in our previous study, though there was some overlap between studies.

Finally, we concluded there was no need to compare cluster and node assignment algorithms under the expanded set of input factors from our current study. We reached this conclusion because: (1) the current study found the same main behavioral dimensions as the previous study, (2) the most influential input factors from the previous study also proved most influential in the current study, (3) the only new input factor that showed even moderate influence in the current study appears likely to have similar effects on any cluster and node assignment algorithm, and (4) main effects proved more influential than two-term interactions in both studies, which mitigated the finding that the current study revealed more two-term interactions.

5.0 Conclusions

We conducted a sensitivity analysis to characterize the effects of asymmetries, dynamics, and failures when introduced into Koala, a cloud computing simulator. Previously, we had conducted a sensitivity analysis to characterize Koala under static, homogeneous configurations with various patterns of demand and supply. We compared results from both sensitivity analyses to determine whether injecting these new parameters into the cloud simulator causes fundamental shifts in the factors driving macroscopic behavior and user experience. We found that introducing asymmetries, dynamics, and failures into Koala does not induce fundamental shifts in the factors driving simulator behavior, but these new parameters do exhibit interactions with the main driving factors, and with each other. Our findings suggest that our previous study [1], using Koala to compare VM placement algorithms, need not be extended to consider the effects of asymmetries, dynamics, and failures. These findings also increase confidence in results from the previous study.

References

- [1] K. Mills, J. Filliben, and C. Dabrowski, Comparing VM-placement algorithms for on-demand clouds, in proceedings of the 3rd international IEEE conference on Cloud Computing Technology, Nov. 29-Dec. 1 2011, pp.91-98.
<http://doi.ieeecomputersociety.org/10.1109/CloudCom.2011.22>
- [2] K. Mills, J. Filliben, and C. Dabrowski, An efficient sensitivity analysis method for large cloud simulations, in proceedings of the 2011 IEEE conference on Cloud Computing, July 4-9, 2011, pp. 724-731.
<http://doi.ieeecomputersociety.org/10.1109/CLOUD.2011.50>
- [3] _____, Amazon Elastic Compute Cloud (Amazon EC2), Amazon Web Services <http://aws.amazon.com/ec2/>, [accessed 10/2/2010].
- [4] _____, Amazon Elastic Compute Cloud API Reference, API Version 2009-08-15.
- [5] _____, Amazon EC2 Instance Types <http://aws.amazon.com/ec2/instance-types/>, [accessed 10/2/2010].
- [6] D. Nurmi, et al., The eucalyptus open-source cloud-computing system, in proceedings of the 9th IEEE/ACM international symposium on Cluster Computing and the Grid, May 18-21, 2009, pp. 124-131.
<http://doi.ieeecomputersociety.org/10.1109/CCGRID.2009.93>
- [7] F. Curbera, et al., Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI, Internet Computing, IEEE, March/April, 2002, pp. 86-93.
<http://doi.ieeecomputersociety.org/10.1109/4236.991449>
- [8] C. Dabrowski and K. Mills, VM leakage and orphan control in open-source clouds, in proceedings of the 3rd international IEEE conference on Cloud Computing Technology, Nov. 29-Dec. 1 2011, pp.554-559.
<http://doi.ieeecomputersociety.org/10.1109/CloudCom.2011.84>
- [9] G. E. P. Box, W. G. Hunter, and J. S. Hunter, Statistics for Experimenters 2nd edition, John Wiley and Sons, Inc., New York (2005) pp. 173-233.
- [10] K. Mills and J. Filliben, Comparison of two dimension-reduction methods for network simulation models, J. Res. Natl. Inst. Stand. Technol., 116:5, 2011, pp. 771-783.
- [11] K. Mills and J. Filliben, An efficient sensitivity analysis method for network simulation models, NIST Publication #904961, presented at the Winter Simulation Conference, December, 2010.
- [12] J. L. Phillips, How to Think about Statistics, 6th ed., Freeman, 2000, 202 pages.