A Fine-Grained Performance-Based Decision Model for Virtualization Application Solution*

Jianhai Chen, Dawei Huang, Bei Wang, Deshi Ye, Qinming He, and Wenzhi Chen

College of Computer Science, Zhejiang University,
Zheda Rd. 38, Hangdog 310027, China
{chenjh919,tossboyhdw,wangbei,yedeshi,hqm,chenwz}@zju.edu.cn

Abstract. Virtualization technology has been widely applied across a broad range of contemporary datacenters. While constructing a datacenter, architects have to choose a Virtualization Application Solution (VAS) to maximize performance as well as minimize cost. However, the performance of a VAS involves a great number of metric concerns, such as virtualization overhead, isolation, manageability, consolidation, and so on. Further, datacenter architects have their own preference of metrics correlate with datacenters' specific application scenarios. Nevertheless, previous research on virtualization performance either focus on a single performance concern or test several metrics respectively, rather than gives a holistic evaluation, which leads to the difficulties in VAS decision-making. In this paper, we propose a fine-grained performance-based decision model termed as VirtDM to aid architects to determine the best VAS for them via quantifying the overall performance of VAS according to datacenter architects' own preference. First, our model defines a measurable, in-depth, fine-grained, human friendly metric system with organized hierarchy to achieve accurate and precise quantitative results. Second, the model harnesses a number of classic Multiple Criteria Decision-Making (MCDM) methods, such as the Analytical Hierarchical Process (AHP), to relieve people's effort of deciding the weight of different metrics base on their own preference accordingly. Our case study addresses an decision process based on three real VAS candidates as an empirical example exploiting VirtDM and demonstrates the effectiveness of our VirtDM model.

Keywords: virtualization, performance evaluation, benchmark, datacenter, decision making, analytic hierarchical process.

1 Introduction

Virtualization technology has been widely applied across a wide-spread of contemporary datacenters due to its benefits of improved utilization, reduced-cost,

* This work is funded by the National 973 Basic Research Program of China under grant NO.2007CB310900 and National Natural Science Foundation of China under grant NO. 60970125.

R. Nambiar and M. Poess (Eds.): TPCTC 2011, LNCS 7144, pp. 180–195, 2012.

saved-energy, manageability and reliability. Gartner reported that the installed base of Virtual Machines (VMs) will grow 5 times from 2009 to 2012, and by 2012 half of the server installed base will be virtualized [1].

Moreover, contemporary datacenters and cloud infrastructures have grown to a grand scale. For example, Google had owned more than 450,000 servers early in 2006 [2]. On the other hand, various kinds of virtualization technologies has been designed and implemented, such as para-virtualization, hardware assistant virtualization, live migration strategies and so on, which offers abundant alternatives to deploy virtualization in a datacenter. As a result, datacenter architects face the crucial issue that how to choose a Virtualization Application Solution (VAS) so that it could maximize performance and best adapt to the demand of their datacenter. In other word, we have to find an evaluation method to compare different virtualization solutions of a datacenter.

Nevertheless, the performance of a VAS involves a great number of performance concerns, such as virtualization overhead, isolation, manageability, consolidation, and so on [3]. Furthermore, a datacenter has its own preference of metrics correlate with specific application scenarios.

Previous research on virtualization performance either focus on a single performance concern or test several metrics respectively. A great number of researches devoted to the characterization and analysis of server consolidation [4,5,6]. Matthews *et al.* investigated the evaluation on the performance isolation of virtual machine [7]. Several works summarized the primary performance perspectives of virtualization and discussed their metrics [8,9,10], while others studied their measurement and benchmarking method [11,12]. These studies didn't provide an overall evaluation method to adaptively compare different VASes, which leads to the difficulties of deciding a VAS best fit into a datacenter's requirement.

In this paper, we propose a fine-grained performance-based decision model for VAS, termed as VirtDM. It provides an overall quantification method to compare different VASes according to the architects' preference, to solve the VAS decision-making problem in a datacenter. VirtDM divides VAS decision-making problem into three sub-problems:

1. What metrics should be taken into account to measured a VAS?
2. How to quantify a datacenter architect's preference on these metrics?
3. How to achieve an overall decision from different metrics' results and architects' preference?

To solve problem 1, we define a fine-grained, hierarchical metrics system and provide their measurements or quantification methods. Certainly the metrics should be chosen so much human-friendly that can be easily used for decision. For problem 2, VirtDM allows people to input pairwise comparison ratios other than to directly give the weights, thus eases people's comparison effort as well as improves the accuracy and precision. For problem 3, we harness a number of classic Multiple Criteria Decision-Making (MCDM) methods, such as Analytical Hierarchical Process (AHP) [13]. VirtDM will normalize different metrics' results, calculate metrics' weights and finally provide an overall numeric result for each VAS. Then datacenter architects could decide the VAS best fit into their requests.

The contributions of our work are three-fold. First, we design an effective model VirtDM to assist the VAS decision making for a datacenter. We implement an algorithm for our model and validate our model by a case study. Second, we build a fine-grained hierarchical metrics system to evaluate different performance characteristics of VAS. We give their measurements or quantification methods. Third, we offer a convenient way to calculate metrics' weights using classic MCDM methods, and provides an overall VAS evaluation method adaptive to datacenter architects' preferences.

The rest of this paper is organized as follows. Section 2 presents the architecture of VirtDM. Section 3 describes metrics system for how to choose the metrics for our model. Section 4 explains the implementation of VirtDM, especially of the metrics' results normalization and metrics' weights identification. Section 5 demonstrates a case study, including both the experiments and the overall decision process. Finally, section 6 provides our conclusions & future work.

2 Architecture of VirtDM Model

In this section, we describe the architecture of VirtDM model.

The VirtDM model is designed to achieve the right decision from different VAS candidates for a datacenter. In VirtDM, we embraces different components which will contribute to the accuracy of final decision result. Fig. 1 shows the architecture of VirtDM. It consists of five abstract components: VAS Candidates, Metrics System, Preference, MCDM-Processor, Decision Result. Basing on the Metrics System and the Preference, the MCDM-Processor will carry out the decision making process over different VAS candidates, and finally yields the Decision Result.

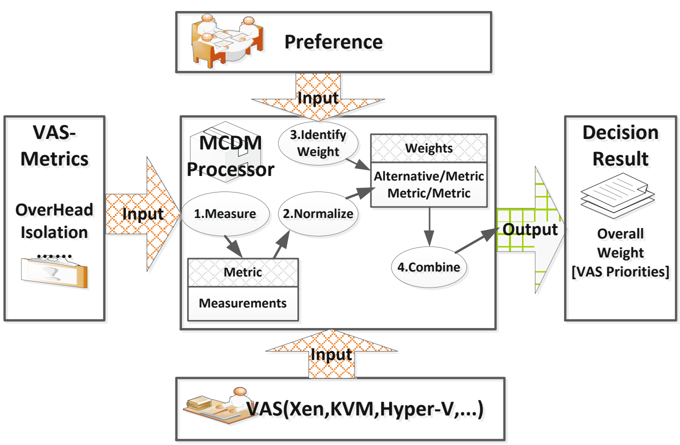

Fig. 1. The architecture of VirtDM

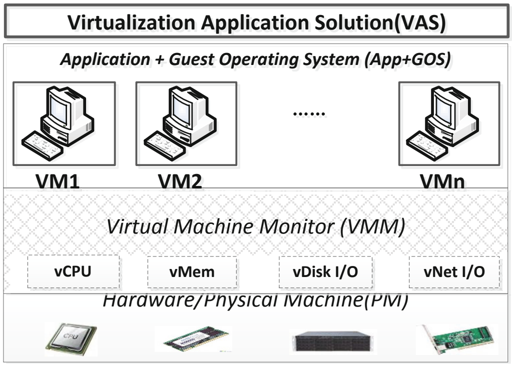

Fig. 2. The components of VAS

A VAS candidate refers to a specific software and hardware implementation of virtualization technologies. It involves Hardware, Virtual Machine monitor(VMM) and Virtual Machine(VM), as Fig 2 shows. The performance of a VAS depends on the software virtualization technologies regarding the VMM, e.g. as Xen, KVM, VMWare etc., and the hardware virtualization technologies like Intel-VT, AMD-SVM, Extended Page Tables, and so on.

Metrics System determinates which virtualization performances are concerned and how they are measured. In VirtDM, we construct a fine-grained, quantifiable, hierarchical and human friendly metrics system, covering essential performance characteristics of virtualization in a datacenter, and set up their measurement or quantification methods as accurate and precise as possible. We will discuss the details of our Metrics System in section 3.

The preference of decision-makers impacts how much final Decision Result fit into a datacenter's definite demand. Because different datacenters might have their individual application scenarios, which results in different preferences. For example, if a datacenter would perform large mount of I/O processing, the architects will care much more about the I/O overhead metric of a VAS.

Furthermore, the MCDM-processor is central in our VirtDM model. It constitutes the key decision making process logic. The procedure of VirtDM involves four primary tasks including measuring, normalization, the weights identification and the combination of decision results. We will talk about more details in section 4.

3 Metrics Choosing

We should choose right performance metrics so that our VirtDM model could produce more accurate and precise quantitative results. We break down this big picture into following criteria while choosing the metrics:

– The metrics should cover all of the essential performance characteristics embracing both advantage and disadvantage facets of virtualization in a datacenter.

– The metrics should be able to produce precise and accurate numeric results. They must be fine-grained and quantifiable, in order to distinguish VASes' performance.
– The metrics should be human friendly. Our VirtDM model involves architects to make preference between them. The metrics must be well-organized and comparable on the right level of abstraction, thus architects can easily understand and compare them.

In VirtDM, we mainly divide the crucial performance characteristics of datacenter virtualization into the following five categories: overhead, manageability, isolation, consolidation and migration. For each category, we will discuss its significance and specific metrics. We describe the measurement methods for some metrics, and the quantification methods for others which cannot be directly measured.

3.1 Virtualization Overhead

Virtualization overhead is usually one of the major roadblocks getting in the way of employing virtualization. Added layer of VMM introduces extra resources consumption and performance degradation of the Guest OS, due to its tasks of hardware resources managements and its interactions with the Guest OS.

We should define specific overhead metrics and their measurements. The VirtDM requires human's participation in weighting the importance of different metrics. In VirtDM, the overhead is measured by calculating the performance degradation of a workload running on a virtualization solution platform against a non-virtualization platform on the same physical host, to exclude the performance impact of other factors.

We test the overhead of a VAS through the following four essential workloads: 1) CPU task; 2) Memory task; 3) I/O task; 4) Context switch. We consult the micro-benchmarks of LMbench [14] to generate these workloads and acquire their throughput results. The workload of context switch is implemented by the fork() system call function. Each workload is implemented to last long enough for precision.

3.2 Manageability

Manageability leads to the operational efficiency and automation, e.g. rapid provisioning, automated workload management, workload live migration etc. In VirtDM, we define the following specific metrics to represent the manageability of a VAS:

1. **VM resource scalability**. It refers to how much virtual or physical resources could be allocated to a virtual machine, usually limited by VMM implementation, such as vSMP Scalability, pSMP Scalability [8].
2. **Migration function**. It refers to whether a VAS has the capacity of live migration or storage migration.
3. **Consolidation functional scalability**. It refers to how many VMs could be allocated to a physical machine, usually limited by VMM implementation.

4. **VM snapshot save/resume efficiency**.
5. **VM start/shutdown efficiency**.

We measure 4) and 5) using their response time. For 2) and 3) we consider that their functions are provided or not, but leave the evaluation of their efficiency in following subsections. Notice that 1), 2) and 3) are not measurable metrics. In Section 4.2, we will describe how to obtain the numeric values for these immeasurable metrics.

Here we just think about what functions are provided to facilitate manageability. We temporarily exclude the consideration of how well a VAS manages physical resources, since it would involve automatic management policies which is difficult to measure.

3.3 Isolation

Virtualization enhances the degree of isolation by restricting multiple software stacks in their own VMs. But security isolation and environment isolation problems will remain as long as the physical resources are shared among different VMs. Therefore, we dwell on the performance isolation. Performance isolation refers to how well a virtualization solution is able to limit the impact of a misbehaving VM on other well-behaving VMs.

We consult previous works on isolation benchmarking [7]. In VirtDM, we run different stress tests - CPU bomb, memory bomb, I/O bomb - to cause extreme resource consumption and refer their VMs as bad VMs. Then we measure the performance degradation of the normal workloads on a well-behaving VM, caused by the bad VM sharing the hardware resource of the same physical host.

3.4 Consolidation

Server consolidation is the most common practice of virtualization in datacenters. It refers to running multiple VMs concurrently on one physical host, to increase resource utilization and reduce cost such as power, space and cooling devices etc. [5].

We use SPECvirt_sc2010 [15] to measure the performance of server consolidation. SPECvirt_sc2010 scales the workloads on the System Under Test (SUT) until the SUT reaches its peak performance, when additional workload VMs either fails the QoS(Quality of Service) or fails to improve overall metric.

3.5 Migration

Migration allows a running VM to be moved from one physical machine to another without any disruption of service or perceived downtime. It provides an essence capacity required for dynamic load balancing, VM replacement, high availability of service during maintenance, and declined power consumption.

We use Virt-LM benchmark [16] to measure the performance of live migration, which provide the results of four metrics - downtime, total migration time, the amount of migrated data and migration overhead across a wide range of classic application workloads in a datacenter.

4 VirtDM Modeling

In this section, we specify our VirtDM to achieve an overall decision making method process and illustrate the relevant components. We will formulate the MCDM problem and detail its implementation. Before modeling, it is necessary to identify the candidate VASes with specific metrics, and give a clear definition for the MCDM problem of VirtDM.

Generally the goal of choosing VASes in datacenter is to satisfy the daily demand of the multiple application services such as web hosting, e-business sites, and enterprise systems etc. Datacenter architects will combine the existing physical machines and VMMs occasionally to full utilize the hardware/software resources with virtualization technology. The combinations compose a variety of VASes. Further, based on the performance measurements of multiple metrics these VASes will arise special performance features as a well proof of decision-making. Thus, the decision problem induces the considerations on the given VASes, the metrics, additionally, more importantly, as well as the human preference.

Besides, MCDM researchers have constructed a number of MCDM methods, such as AHP [13], LINMAP, TOPIS [17,18], etc. We primary consult the AHP technique which is one of the most efficient MCDM method to implement the MCDM-Processor of VirtDM. The VirtDM aims to find the optimal weight of attribute for a group of VAS alternatives, to determine a rational ranking order as well.

4.1 VirtDM Formulation

In this section we state the MCDM problem of VirtDM and present its formulation in order to express the decision-making process conveniently.

Problem 1. (**Generalization problem**). The MCDM problem of the VirtDM is provided with a hierarchy structure and must be decomposed into levels as shown in Fig. 3. It comprises L-levels ($L >= 3$): alternative(VAS candidate, one

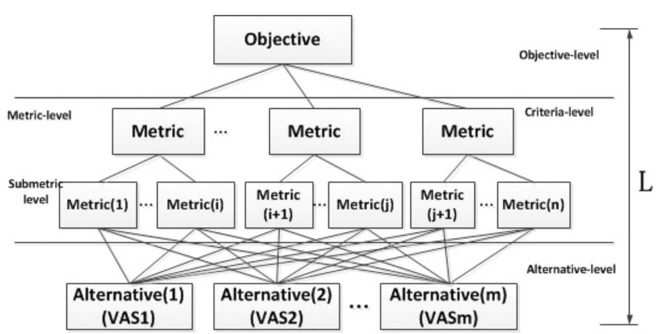

Fig. 3. The formulation of VirtDM with hierarchical structure

fixed level), criteria (one or more of metric levels, apparently equal to $L-2$ levels) as well as objective (decision objective level, one fixed level).

Each level incorporates several elements. The elements of a given level can be comparable with the elements of the same level which elements are mutually exclusive. VirtDM presumes that the elements of a given level are affected by elements at the level directly above them besides the top level.

Problem 2. (**Special case**). Let $L = 4$ in problem 1, then we have a new hierarchy MCDM problem with four levels. In this way, in addition to the alternative-level and objective-level, the metric-level of general problem is divided into two levels: ML(metric-level) and SML(sub-metric level) as illustrated in Fig. 3. Each metric of the metric-level can be composed of several sub-metrics of the sub-metric level.

For simplification, we use problem 2 to implement the decision process of VirtDM. Some constraints are initialized in the following two definitions.

Definition 1. ***(Size of the hierarchy structure)****. Assume problem 2 contain m alternatives(VASes), n sub-metrics and s metrics in the criteria level as well as* 1 *objective at the top level. Two adjacent levels are directly related.*

If the i_{th} given metric of the metric-level contains n_i sub-metrics, it will satisfy the equation: $\sum_{i=1}^{s} n_i = n$.

Definition 2. ***(Decision Attribute Matrix)****. For each alternative, we can obtain a numerical value, called an attribute, for each metric of sub-metrics. Then, in problem 2 we have $m \times n$ attributes which comprise the decision basis of VirtDM. To store the decision attributes, we give a matrix: $D = (d_{ij})_{m\times n}, (i = 1, ..., m, j = 1, ..., n)$, where the element d_{ij} represents the j^{th} sub-metric value of the i^{th} VAS alternative. This matrix is called Decision attribute matrix (DAM).*

4.2 VirtDM Implementation

Besides problem formulation, to achieve the aim, the implementation of VirtDM is covering with several procedures as the follows.

1. Metrics Quantification. The metrics (elements) in **DAM** must be quantified before the weight identification. In problem 2 the metrics are categorized into two groups: quantitative and qualitative. Only the immeasurable metrics which are qualitative require to be expressed by numerical value. The qualitative metrics are commonly quantified by fuzzy language such as Bipolar scale method. The Bipolar scale has 10 cells with the common used scope of $1-9$. Here "Bipolar" means it can be used to quantify both Cost-type metrics and Benefit-type metrics. Regardless of what type of metrics, the maximum preference is identified as 9.0, while the minimum one identified as 1.0.

2. Metrics Normalization. We can begin the weight identification after all metrics in **DAM** D are quantified.

However, the metric numerical values typically exist three issues: the inconsistency dimension, the mixture of qualitative and quantitative metrics, as well as the difference of attribute orientation. Moreover, before the weight determination, the values should be normalized to be consistent attribute orientation and dimensionless. The metrics can be categorized into two types: benefit-criteria and cost-criteria which have different normalization ways.

The VirtDM incorporates three normalization means such as vector normalization method, linear scale transformation method and $(0-1)$ interval conversion method [19].

Besides, for the using of the AHP technique, we also use Formula(1), a simple additive weighting method to over again normalize the metric value:

$$r_{ij} = d_{ij} / \sum_{i=1}^{m} d_{ij}. \tag{1}$$

As mentioned in above Formulas, we obtain the matrix $R = (r_{ij})_{m \times n}$, a new normalized decision making matrix.

The purpose of our MCDM is to calculate an overall score for each alternative based on the metrics.

The basis process of weight identification is as follows. From the bottom alternative level to the top objective level one-by-one, VirtDM identifies the weights of the elements in a given level relative to the elements of the level directly above them. VirtDM applies MCDM with a weighted sum model (WSM) [19] as a uniform evaluation method. The i^{th} alternative is given a score by Formula(2).

$$Score(A_i) = \sum_{j=1}^{n} r_{ij} * w_j. \tag{2}$$

3. Weight Identification. We use the pairwise comparison method to identify the weights of metric at each metric-level in associate with decision-maker's preference in problem 2. We consider the weights hierarchically. First the weights of metrics in metric-level need to be identified relative to one objective of the objective-level. Then, each metric of metric-level consists of several sub-metrics of sub-metric level. So N metrics of metric-level require N iterations identification to the weights of metrics of sub-metric level relative to the each metric of metric-level. Totally we will have $N + 1$ iterations weight identifications in problem 2.

We suppose the weights of metric in metric-level relative to objective of top level is denoted as W_2 and the weights of metric in sub-metric level relative to metric of metric-level with N metrics is expressed by $W_{3i}, i = 1...N$, where N denotes N metrics of metric-level and the dimensions of W_{3i} depends on the amount of sub-metrics.

Each iteration weight identification has the same process. We present an algorithm Alg_WIA for weight identification as shown in Algorithm 1.

Algorithm 1. Algorithm WIA : weight identification of pairwise comparison method

Require:

The amount of metrics for pairwise comparison, N;
The random index corresponding to N dimension, RI;
The pairwise comparison matrix, $P = (x_{ij})_{N \times N}$;

Ensure:

The weight vector of metrics, $W = (w_i)^T, i = 1..N$;

1: Determining and input the elements of P according to decision maker's preference by Satty's scale method [13].
2: Use geometric mean method as Formula (3): an approximate method to calculate the weight $W = (w_i)^T, i = 1..N$.

$$m_i = \prod_{j=1}^{N} x_{ij}, \qquad \overline{w_i} = \sqrt[n]{m_i}, \qquad w_i = \overline{w_i} / \sum_{i=1}^{n} \overline{w_i}. \tag{3}$$

3: Using Formula (4): an approximate method to calculate the maximum eigenvalue of P.

$$\lambda_{max} = \sum_{i=1}^{n} (\sum x_{ij} W)_i / n \cdot w_i. \tag{4}$$

4: Using Formula (5) to carry out CI and CR.

$$CI = (\lambda_{max} - n)/(n-1), \qquad CR = CI/RI. \tag{5}$$

5: If $CR < 0.5$ then goto step (6) to output the result weight vector: W; else goto step (1);
6: **return** W.

4. Weight Combination. Eventually, after all the weights identification in each level are completed, we can combine them into just one vector by multiplying all the weight vectors as the following Formula (6):

$$V = R * W3 * W2 * W1, \tag{6}$$

where R is a normalized decision attribute matrix; V is an overall VAS priority vector, stands for the satisfaction degree of the decision making results.

5 Case Study

In this section, we demonstrate a case study applying VirtDM model to make decision the best VAS alternative among three ones, supposing a datacenter be deployed preferring I/O performance. Fig. 4 shows the hierarchy structure of our decision case and the chosen metrics are simplified ignoring human subjects for the goal is just to illustrate the whole process of how VirtDM works to validate its usefulness. On the other hand, this case could easily be extended to complex ones with more VAS candidates or more complicated metrics concerns.

5.1 Experimental Environment Setup

We experiment our VirtDM method on three VAS platforms, as following shows.

1. VAS-XEN-HV: The physical host is a Dell PowerEdge T710, with dual quad-core Intel Xeon processor E5620 at 2.4GHZ and 24GB of memory. The VMM is Xen-3.3.1 with Linux Kernel 2.6.18.8-xen. The VM is Linux 2.6.21 with 1GB memory and 1 vcpu binding to a particular physical CPU core.
2. VAS-XEN-PV: Using the same host and VMM as VAS-XEN-HV but with a para-virtualized VM.
3. VAS-KVM: Using the same host and VM as VAS-XEN-HV but with a different VMM — KVM.

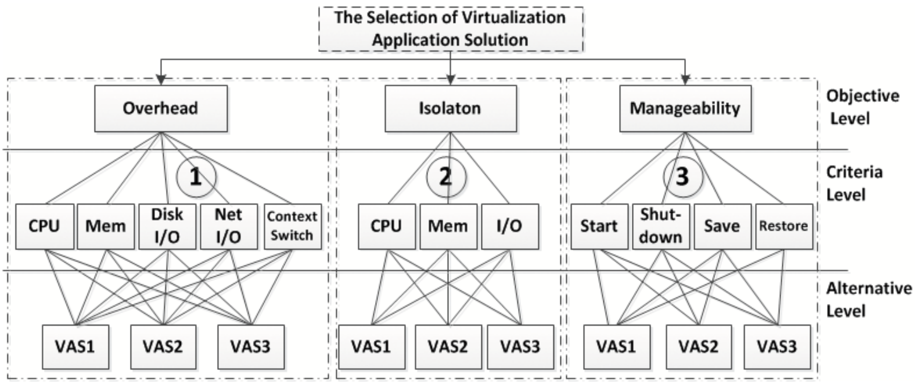

Fig. 4. The case of the hierarchy of the VAS decision making problem. $VAS1$ refers to VAS-XEN-HV, $VAS2$ denotes VAS-XEN-HV and $VAS3$ represents VAS-KVM.

5.2 Performance Measurement

To simplify the illustration of VirtDM process, ignoring human subjective or qualitative metrics, we choose three categories of measurable metrics — overhead, isolation, manageability — in our experiment. Of course our VirtDM model could be extended to other different metrics choices using the same course of our case study. Fig. 5 provides the metrics results of the three different VAS alternatives and Table 1 shows the performance measurements.

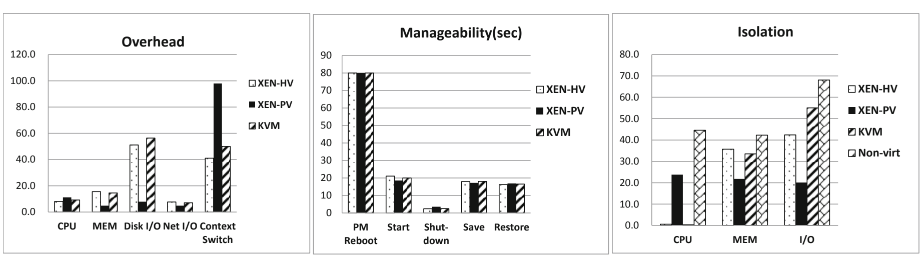

Fig. 5. The metrics results of three VASes

Table 1. Performance measurements from three VASes: XEN-HV, XEN-PV, KVM

	Overhead[%]					Isolation[%]			Manageability[sec]			
	CPU	Mem	Disk I/O	Net I/O	C.S.	CPU	Mem	I/O	start	shut.	save	rest.
XEN-HV	8.1	15.5	51.1	7.7	41	0.6	35.7	42.4	21	2.4	17.9	16.2
XEN-PV	11.13	4.8	7.9	4.8	98	23.8	21.8	20.1	18.5	3.5	17.2	16.9
KVM	9.13	14.5	56.3	7.0	50	0.4	33.5	55.0	20	2.5	18	16.5

Overhead. According to Section 3.1, we measure the virtualization overhead of four workloads: CPU task, memory task, I/O task and Context Switch task. The metric results are calculated by the performance degradation percentage of the workloads running on the VAS against running on the physical host. Lower is better. As Fig. 5 shows, all VAS alternatives achieved less overhead on CPU and Memory tasks, but greater overhead on I/O and Context Switch tasks, because I/O and context switch tasks would cause more interactions of VMM. However, VAS-XEN-PV gained much better performance on I/O overhead test, especially on disk I/O test, although it produces the worst performance about the context switch overhead. The reason is that para-virtualization mechanism using modified I/O driver which could significantly decrease the number of VMM context switches.

Isolation. According to Section 3.3, we measure the isolation with the normal VM's performance degradation cause by a "bad" VM which produces extreme resources consumption. We tested three kind of performance isolation using different kinds of "bad" VMs — the CPU stressed, the Memory stressed and the I/O stressed. Hence, lower is better. For comparison, we also test the performance isolation of non-virtualization case, in which the normal workload are impacted by a stressed workload within the same single OS.

As Fig. 5 demonstrates, CPU isolation is generally better than the other two. Further, VAS-XEN-HV and VAS-KVM had really poor memory isolation, while VAS-XEN-PV illustrated quite bad I/O isolation.

Manageability. In this experiment we merely test the duration of general VM operation – VM start, shutdown, save and restore. Lower is better. We also the duration of the physical machine's reboot to illustrate the virtualization efficiency. The three VASes attained very close results of each metrics as displayed in Fig. 5.

5.3 Overall Decision Process

In this section we show the steps of the decision process using our VirtDM model.

Example 1. As an example shown in Fig. 4, it provides a hierarchy MCDM including 4-levels. Besides bottom and top level, the criteria level is composed of two levels: 1) metric-level which involves three metrics: overhead, isolation and manageability metrics; 2) the sub-metrics of the sub-metric level. All sub-metrics are cost-type and have been measured before decision. The metrics of the metric-level derive from the synthesization of the sub-metrics in the level directly below them.

Step 1. Normalizing the attribute data to be dimensionless.
According to the common normalizing methods, we normalize the raw metrics results data shown in Table 1 to be dimensionless by using Formula (7) and the cost-criteria linear conversion method.

$$r_{ij} = \min_{1 \leq i \leq m} x_{ij}/x_{ij}. \tag{7}$$

Further, we again normalize the weights by simple weighted mean method to satisfy the sum of the weights of all VAS alternatives in the alternative-level added up to 1. For example, the following equation is right for each sub-metric: $w(VAS1)+w(VAS2)+w(VAS3) = 1$. The normalized result is shown in Table 2.

Table 2. The normalized data

	Overhead[%]					Isolation[%]			Manageability[sec.]			
	CPU	Mem	Disk I/O	Net I/O	C.S.	CPU	Mem	I/O	Start	Shut.	Save	Rest.
XEN-HV	0.38	0.19	0.12	0.27	0.45	0.38	0.27	0.26	0.31	0.38	0.33	0.34
XEN-PV	0.28	0.61	0.77	0.43	0.19	0.01	0.44	0.54	0.36	0.26	0.34	0.33
KVM	0.34	0.20	0.11	0.30	0.36	0.61	0.29	0.20	0.33	0.36	0.33	0.33

Step 2. Constructing the decision-making matrix.
For the convenient calculation, we extract the metrics from Table 2 to create three decision Matrices: O—Overhead decision matrix; I—Isolation decision matrix; M—Manageability decision matrix, where,

$$O = \begin{bmatrix} 0.38 & 0.19 & 0.12 & 0.27 & 0.45 \\ 0.28 & 0.61 & 0.77 & 0.43 & 0.19 \\ 0.34 & 0.20 & 0.11 & 0.30 & 0.36 \end{bmatrix}, I = \begin{bmatrix} 0.38 & 0.27 & 0.26 \\ 0.01 & 0.44 & 0.54 \\ 0.61 & 0.29 & 0.20 \end{bmatrix}, M = \begin{bmatrix} 0.31 & 0.38 & 0.33 & 0.34 \\ 0.36 & 0.26 & 0.34 & 0.33 \\ 0.33 & 0.36 & 0.33 & 0.33 \end{bmatrix}$$

.

Step 3. Identifying the weights for sub-metrics and metrics.
In this MCDM, without immeasurable metrics in the sub-metrics, the weights of the alternative-level relative to the metrics of the sub-metric level do not required to be determined by using preference in pair wise comparison method but immediately be identified from the measurements in the decision matrix. On the contrary, the identifications of weights using pair wise comparison method concentrate on the metrics of the sub-metric level relative to the metrics of metric-level. Each metric of the metric-level relative to the relevant sub-metrics need one iteration weight identification. It indicates that three metrics reflect three iterations. In addition, the weight of metrics of metric level relative to

the top level(objective level) needs one iteration. Hence, this MCDM exists four iterations using pair wise comparison method to determine weights.

This weight determination method associates decision-maker's preference with the pairwise comparison matrix. According to the requirement of the pairwise comparison method, we determine the weights with the help of eigenvector theory-based acceptance validation, and obtain the rational weight vectors, respectively.

We create four pairwise comparison matrices: $PO, PI, PMandPP$ as follows:

$$PO = \begin{array}{c} \\ CPU \\ Mem \\ DiskI/O \\ NetI/O \\ Cont.S \end{array} \begin{array}{c} \begin{array}{ccccc} CPU & Mem & DiskI/O & NetI/O & Cont.S \end{array} \\ \begin{bmatrix} 1 & 1 & 0.111 & 0.14 & 0.333 \\ 1 & 1 & 0.143 & 0.125 & 0.2 \\ 9 & 7 & 1 & 0.5 & 2 \\ 7 & 8 & 2 & 1 & 3 \\ 3 & 5 & 0.5 & 0.333 & 1 \end{bmatrix} \end{array}, PI = \begin{array}{c} \\ CPU \\ Mem \\ I/O \end{array} \begin{array}{c} \begin{array}{ccc} CPU & Mem & I/O \end{array} \\ \begin{bmatrix} 1 & 3 & 5 \\ 0.33 & 1 & 3 \\ 0.2 & 0.33 & 1 \end{bmatrix} \end{array},$$

$$PM = \begin{array}{c} \\ start \\ shut. \\ save \\ res. \end{array} \begin{array}{c} \begin{array}{cccc} start & shut. & save & res. \end{array} \\ \begin{bmatrix} 1 & 1 & 0.17 & 0.17 \\ 1 & 1 & 0.2 & 0.2 \\ 6 & 5 & 1 & 1 \\ 6 & 5 & 1 & 1 \end{bmatrix} \end{array}, PP = \begin{array}{c} PCM1 \\ ove. \\ iso. \\ man. \end{array} \begin{array}{c} \begin{array}{ccc} ove. & iso. & man. \end{array} \\ \begin{bmatrix} 1 & 3 & 9 \\ 0.33 & 1 & 4 \\ 0.11 & 0.25 & 1 \end{bmatrix} \end{array}$$

- (1) PO is used to identify the weights of sub-metrics: CPU, Disk I/O, Net I/O, Context Switch(Cont.S) relative to the metric overhead.
- (2) PI is used to identify the of sub-metrics: CPU, Memory, I/O relative to isolation metric.
- (3) PM is used to identify the weights of sub-metrics: start time, shutdown time(shut.), save time(save.), restore time(rest.) relative to manageability metric.
- (4) PP is used to identify the weights of metrics: overhead (ove.), isolation (iso.), and manageability (man.) in the metric-level relative to the objective top-level.

In this MCDM, we assume that datacenter administrators are to make decision of choosing a high performance VAS with better I/O performance. Therefore the metric disk I/O and Net I/O are given a high preference value in the matrices.

Based on each comparison matrix, we calculate all relevant weight vectors by eigenvector theory and calculate the approximate weights by geometry mean method. All the pairwise comparison matrices get through the consistency validation.

Finally, the relevant valid weights are expressed as follows:
(1)$Wo = (0.048, 0.044, 0.311, 0.435, 0.163)^T$ is the overhead weight vector,
(2)$Wi = (0.634, 0.260, 0.106)^T$ is the isolation weight vector,
(3)$Wm = (0.075, 0.081, 0.422, 0.422)^T$ is the manageability weight vector,
(4)$Wp = (0.681, 0.250, 0.069)^T$ is the synthetic weight vector.

Step 4. Combining the weights and result analysis.
We calculate the combined weights of overhead (W_1), isolation (W_2), as well as manageability (W_3) as follows.

$$(1)W_1 = O{\cdot}Wo = \begin{bmatrix} 0.38\ 0.19\ 0.12\ 0.27\ 0.45 \\ 0.28\ 0.61\ 0.77\ 0.43\ 0.19 \\ 0.34\ 0.20\ 0.11\ 0.30\ 0.30 \end{bmatrix} \cdot \begin{bmatrix} 0.048 \\ 0.044 \\ 0.311 \\ 0.435 \\ 0.163 \end{bmatrix} = (0.254, 0.499, 0.248)^T$$

$$(2)W_2 = I \cdot Wi = \begin{bmatrix} 0.38\ 0.27\ 0.26 \\ 0.01\ 0.44\ 0.54 \\ 0.61\ 0.29\ 0.20 \end{bmatrix} \cdot \begin{bmatrix} 0.634 \\ 0.260 \\ 0.106 \end{bmatrix} = (0.338, 0.178, 0.484)^T,$$

$$(3)W_3 = M \cdot Wm = \begin{bmatrix} 0.31\ 0.38\ 0.33\ 0.34 \\ 0.36\ 0.26\ 0.34\ 0.33 \\ 0.33\ 0.36\ 0.33\ 0.33 \end{bmatrix} \cdot \begin{bmatrix} 0.075 \\ 0.081 \\ 0.422 \\ 0.422 \end{bmatrix} = (0.337, 0.330, 0.333)^T.$$

The final result of the decision-making process is concluded by the following Formula:

$$V{=}(W_1, W_2, W_3){\cdot}Wp{=} \begin{bmatrix} 0.254\ 0.499\ \ 0.248 \\ 0.338\ 0.178\ \ 0.484 \\ 0.337\ 0.330\ 0.0.333 \end{bmatrix} \cdot \begin{bmatrix} 0.681 \\ 0.250 \\ 0.069 \end{bmatrix} {=} (0.281, 0.407, 0.313)^T.$$

It concludes the combined vector V which represents the VAS priority. It indicates the rank order: $0.281 < 0.313 < 0.407$, which is corresponding to $VAS1 < VAS3 < VAS2$. Thus, the second VAS alternative, namely, XEN-PV, is the best choice for our given MCDM problem in the case.

6 Conclusions and Future Work

In this paper, we design and implement the VirtDM model to serve the VAS decision making in a datacenter. We define a fine-grained, in-depth, and human friendly metrics system to cover essential performance characteristics of a VAS. We employ classic MCDM methods to ease the quantification of people's preference. VirtDM will measure different metrics, normalize their results, calculate their weights fit into people's preference and finally give an overall decision from given VAS candidates.

However, many aspects of VirtDM are far from satisfying. For example, our metrics system is fair rough and omits some metrics difficult to measure, e.g. the efficiency of the automatic policies of consolidation or migration. Further, our model are primarily based on AHP method and other MCDM methods maybe more sophisticated and more appropriate. These deserve our further investigation and effort to improve.

References

1. Bittman, T., Webinar, G.: Server virtualization: From virtual machines to clouds (2010)

2. Blachman, N., Peek, J.: How google works. (GoogleGuide) (retrieved April 20, 2007)
3. Uhlig, R., Neiger, G., Rodgers, D., Santoni, A., Martins, F., Anderson, A., Bennett, S., Kagi, A., Leung, F., Smith, L.: Intel virtualization technology. Computer 38(5), 48–56 (2005)
4. Apparao, P., Iyer, R., Zhang, X., Newell, D., Adelmeyer, T.: Characterization & analysis of a server consolidation benchmark. In: Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 21–30. ACM (2008)
5. Padala, P., Zhu, X., Wang, Z., Singhal, S., Shin, K.: Performance evaluation of virtualization technologies for server consolidation. HP Labs Tec. Report (2007)
6. Makhija, V., Herndon, B., Smith, P., Roderick, L., Zamost, E., Anderson, J.: Vmmark: A scalable benchmark for virtualized systems. VMware Inc., CA, Tech. Rep. VMware-TR-2006-002 (September 2006)
7. Matthews, J., Hu, W., Hapuarachchi, M., Deshane, T., Dimatos, D., Hamilton, G., McCabe, M., Owens, J.: Quantifying the performance isolation properties of virtualization systems. In: Proceedings of the 2007 Workshop on Experimental Computer Science, p. 6-es. ACM (2007)
8. McDougall, R., Anderson, J.: Virtualization performance: perspectives and challenges ahead. ACM SIGOPS Operating Systems Review 44(4), 40–56 (2010)
9. Huber, N., von Quast, M., Brosig, F., Kounev, S.: Analysis of the Performance-Influencing Factors of Virtualization Platforms. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6427, pp. 811–828. Springer, Heidelberg (2010)
10. Kundu, S., Rangaswami, R., Dutta, K., Zhao, M.: Application performance modeling in a virtualized environment. In: 2010 IEEE 16th International Symposium on High Performance Computer Architecture, HPCA, pp. 1–10. IEEE (2010)
11. Ye, K., Che, J., Jiang, X., Chen, J., Li, X.: vtestkit: A performance benchmarking framework for virtualization environments. In: The Fifth Annual China Grid Conference, pp. 130–136. IEEE (2010)
12. Moller, K.: Virtual machine benchmarking (2007)
13. Saaty, T.: Decision-making with the ahp: Why is the principal eigenvector necessary. European Journal of Operational Research 145(1), 85–91 (2003)
14. McVoy, L., Staelin, C.: lmbench: Portable tools for performance analysis. In: Proceedings of the 1996 Annual Conference on USENIX Annual Technical Conference, pp. 23–23. Usenix Association (1996)
15. specvirt_sc (2010), http://www.spec.org/virt_sc2010/
16. Huang, D., Ye, D., He, Q., Chen, J., Ye, K.: Virt-lm: a benchmark for live migration of virtual machine. In: Proceeding of the Second Joint WOSP/SIPEW International Conference on Performance Engineering, pp. 307–316. ACM (2011)
17. Wang, P., Chao, K., Lo, C.: On optimal decision for qos-aware composite service selection. Expert Systems with Applications 37(1), 440–449 (2010)
18. Tarighi, M., Motamedi, S., Sharifian, S.: A new model for virtual machine migration in virtualized cluster server based on fuzzy decision making. Arxiv preprint arXiv:1002.3329 (2010)
19. Hwang, C., Yoon, K.: Multiple attribute decision making: methods and applications: a state-of-the-art survey, vol. 13. Springer (1981)