ReKon: Recombinable Knowledge Units for Systems Integration Projects

Sandeep Purao1,2, Arvind Karunakaran1, Brian Cameron1
1 College of Information Sciences and Technology, Penn State University
2 School of Information, University of Washington

Abstract

We describe the ReKon platform. It contains granular knowledge units that can be combined, as needed, to support efforts to build systems integration solutions. They are created from more than 1,200 templates contributed by four IT consulting organizations. The granularity of knowledge units represents a direct response to the emergent nature of systems integration projects that defies a one-size-fits-all approach. The paper develops the underlying conceptual model and operations for ReKon, describes a prototype implementation, and reports results from a two-phase evaluation that point to its potential.

Keywords: Knowledge Units, Method Fragments, Method Engineering, Systems Integration

1. Introduction

We have reached a threshold. Software systems are no longer conceived and constructed as monoliths [1]. Instead, they are often envisioned as complex entities that facilitate interoperations among autonomous systems aka systems-of-systems [2]. The difficulties lie in efforts needed to convert these envisioned solution sketches into workable solutions [1, 3, 4]. It is a challenge that further exacerbates conventional problems of cost overruns and project failures [1, 5]. There are few prescriptive methodologies for constructing such entities. However, the tasks required for constructing these entities tend to be ill-specified, emergent and difficult to anticipate [6]. Addressing these [7, 8] requires a key ingredient: knowing what to do and how. Coarse-grain, abstract and prescriptive software engineering methodologies cannot meet this need. The research question we address is, therefore, the following: Is it possible to devise a platform that can provide granular knowledge assets that project members may combine, as needed, to address the unique needs of different projects to build systems-of-systems?

This paper develops such a platform. It contains Template Chunks extracted from more than 1,200 Project Templates made available for this research project by four IT consulting organizations. The paper describes the process followed for chunking and codification; the conceptual model that underlies the platform; and a prototype implementation. A two-phase evaluation with potential users demonstrates potential usefulness of the approach.

2. A Motivating Example

Consider the following scenario faced by John, Mary and Sam, for a systems integration effort at Painters R Us. It requires constructing a catalog of services from a scheduling application. Their concerns revolve around scalability, security and aligning the outcomes against current processes. None is directly addressed by the templates codified from earlier efforts. These templates, often running into scores of pages, contain instructions and worksheets for tasks such as requirements gathering, designing, and testing. The team realizes that their project is likely to benefit from these. They would ideally like to leverage parts of several templates best suited for this project. The sheer number of templates, and conflicting and overlapping choices, however, means that they are not sure how to move forward. The problems faced by John, Mary and Sam are not new. Templates have been traditionally used in several organizations. They represent, among other things, rules, routines and best practices accumulated from past projects [9, 10]. To understand their concerns better, we first review prior work from several streams.
3. Related Work

**Building Systems of Systems**: Systems of Systems (SoS) represent “a collection of … systems that (are brought) … together to obtain a new, more complex (entity) [6, 11].” The track record of building large and complex systems-of-systems continues to be sub-par [2, 3] with few pointers to antecedents of success [3, 4, 6, 12-14] in spite of prior work elsewhere [12-14]. Failures have been attributed to several reasons including unsuitability of approaches for the intended task [8, 15-17]. Anecdotes, lessons learned and best practices, however, continue to be accumulated because of the lengthy and complex nature of these projects – often codified as templates, forms, and checklists. Yet, the unique needs of new projects often tend to be out-of-sync with routines suggested by these project templates (like the problems faced by John, Mary and Sam). This mismatch between knowledge needs of a project and availability of knowledge from prior efforts (and its potential impact) remains unexplored [5, 6, 11].

**Knowledge Management**: Definitions of knowledge management (KM) vary based on views of knowledge such as ‘state of mind’, an ‘object to be stored and manipulated’ [18]. We define KM as ‘concerned with identifying and leveraging the knowledge in an organization’ [19]. Prior significant research related to KM enablers and reasons for failure [9, 18, 20-23] may be understood in terms of the framework from Nonaka and Takeuchi [26], who distinguish individual versus collective knowledge, and tacit versus explicit knowledge [24]. Knowledge creation is then described as a process that begins with the tacit knowledge of an individual [24-26], and moves forward as a spiral through the phases of socialization, externalization, combination and internalization (returning to individual tacit knowledge) [24-26]. In the context of projects to build systems-of-systems, the tacit and explicit knowledge tends to be ‘mutually dependent’ [18, 27] and can have reinforcing qualities that are not exploited if KM efforts focus on codifying explicit knowledge alone [18, 28].

**Method Engineering**: Method engineering (ME) asserts that methods should be constructed to meet a particular project need to overcome limitations posed by general approaches [29-33]. Following ME, a method consists of several fragments (such as DFD or ER) [34-36] identified from existing methodologies (the body of knowledge about methods) [29, 31, 32]. A method engineer creates a personalized method in response to the project needs [29]. ME, however, faces two significant concerns: granularity, and the inherently abstracted nature of knowledge about methodologies, limiting the usefulness of ME in practice. In situations like the one faced by John, Mary and Sam, the ME mandate requires them to return to prescriptive methodologies, ignoring the Templates, and a substantial effort that may still result in integrity issues.

4. Recombinable Knowledge Assets

A key distinguishing feature of this research from ME is the knowledge sources it leverages. It views project templates, instead of abstract methodologies (such as DFD or ER), as carriers of knowledge [25, 26]. Consider, for example, templates for specifying requirements or code review (see Figure 1).

![Figure 1. Project Templates for (a) Gathering Requirements, (b) Code Review](image-url)

These templates do not contain method fragments. Instead, each provides an instance of systemic knowledge, i.e., explicit knowledge systematized and packaged in forms such as documents,
specifications and manuals [25]. From a KM perspective, such systemic knowledge assets stand in contrast with tacit knowledge, which is obtained through work experience, i.e. experiential knowledge or articulated in symbolic form [25]. For this research, more than 1,200 such templates were provided by four IT consulting organizations. The research challenge, then, involves designing a platform that allows recombining knowledge units to facilitate building of SoS, where the knowledge needs tend to be unique and difficult to anticipate.

4.1 The ReKon Model
The ReKon model builds on these foundations. It conceptualizes components needed for a systems-of-systems effort, for example, an interview protocol (logical knowledge unit) for conducting interviews (task) during gathering requirements (phase). Interview protocols available (physical knowledge units) in multiple Project Templates (project templates) can be separated and made available. Project members may access these (retrieval) and combine to create new templates (target template). Figure 2 formalizes the ReKon model.

<table>
<thead>
<tr>
<th>Knowledge Units and Systems-of-Systems Development Efforts</th>
</tr>
</thead>
<tbody>
<tr>
<td>p ∈ P Phases in a Systems of Systems Development Effort</td>
</tr>
<tr>
<td>t,u ∈ T Tasks in a Systems of Systems Development Effort</td>
</tr>
<tr>
<td>l,m ∈ L Logical Knowledge Units</td>
</tr>
<tr>
<td>need (l,t,p) Knowledge Unit l is needed for Task t, Phase p {0,1}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Templates and Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>s ∈ S Project Templates contributed by four Consulting Organizations</td>
</tr>
<tr>
<td>j, k ∈ K Physical Knowledge Units constructed from Project Templates</td>
</tr>
<tr>
<td>part (k,s) Physical Knowledge Unit k is part of Project Template s {0,1}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ReKon Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance (k,l) Physical Knowledge Unit k is an instance of Logical Knowledge Unit l {0,1}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>d ∈ D Templates constructed by combining Physical Knowledge Units</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operations on ReKon</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ k</td>
</tr>
<tr>
<td>k ∈ {k} Selection of the appropriate Physical Knowledge Unit instance k</td>
</tr>
<tr>
<td>d = concatenate (j, k), where j ∈ { j</td>
</tr>
</tbody>
</table>

The objective of ReKon platform is not merely storing and codifying templates in small units. The intent is to empower users to leverage fine-grained templates to respond to the emergent knowledge needs, i.e. allow recombination and appropriation to bridge the institutional and fluid domains [37]. The utility of ReKon lies in the knowledge units and operations. Implicit in the model is the manner in which meeting the knowledge needs of SoS projects is seen as KM instead of ME with a focus on leveraging systemic assets created in one/more organizations.

4.2 Creating Knowledge Units to Populate ReKon
To populate ReKon, templates (~1220) contributed by four leading IT consulting organizations were used. Each was ‘chunked’ and ‘classified’ in a matrix of ‘Phases’ and ‘Tasks’ constructed from PMBOK [14], a prescriptive methodology [38] and a characterization of systems-of-systems [1]. The process followed for deriving knowledge chunks from the templates involved multiple raters, lead by an...
experienced practitioner who was then working on a graduate degree. No direct bias was introduced in the coding process by any employees of the organizations who contributed the templates. A random sample of 122 documents (approximately 10% of the set) was chosen for initial classification. Prior to coding, the coders established common terms. After the first round, the classifications were compared to check consistency, differences were resolved via discussion, and the common terms were enhanced. For the second round, another coder was added, and coding was done on another random sample of 122 documents, and the process was repeated. Inter-coder agreement during the two rounds increased from 78% to 86%. The complete set (~1220 templates) was then divided and randomly assigned to coders who assessed the fit for each template to a particular cell in the matrix. Finally, the templates assigned to each cell were examined to select the best templates with rules such as size, thoroughness and availability of examples.

4.3 A Prototype Implementation

The knowledge units were used to populate and implement the ReKon platform. The prototype shows Tasks and Phases, and at their intersection, the relevant template chunks (Knowledge Units). For example, a client interview protocol represents a logical knowledge unit for conducting interviews (task) during gathering requirements (phase). Knowledge units were available for 36 cells. The total number of knowledge units was 92, with an average of 2.5 knowledge units for each non-empty cell. A browser-based implementation allowed users access to the Knowledge Units (see Figure 3).

5. Evaluation

Formative evaluation proceeded in two phases. A Pre-ReKon phase assessed a Standard (comprehensive) Project Template, aimed at establishing the need for ReKon. A Post-ReKon phase assessed intrinsic properties of knowledge units in ReKon such as granularity, size, appropriateness of classification, relevance of knowledge units, and usefulness for project needs. Subjects were recruited from an Advanced Enterprise Integration course, engaged in real-world integration projects. They used ReKon during their projects and to construct final project deliverables. The utility of ReKon was not directly assessed because the evaluation was formative and controls related to project or users were not possible. Selected results from the pre-ReKon assessment (n=28), summarized in Table 1, point to ambivalence among users.

Table 1. Pre-ReKon Assessment Reflecting on a Standard (Comprehensive) Project Template

<table>
<thead>
<tr>
<th>Question (Average Score, 1=Agree)</th>
<th>Representative Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Although I may not have used all sections, it is useful to have the complete template (1.42)</td>
<td>(+) ... useful to have the template ... so to have a guide (-) ... difficult to determine if a section is relevant.</td>
</tr>
<tr>
<td>It is better to have each section ... separately, so we can create [what] ... we need by combining the sections relevant to our project (2.42)</td>
<td>(+) Most ... will not use all sections ...and it may be easier to make your own document. (-) ... Having sections ... poses the risk of missing something</td>
</tr>
</tbody>
</table>

Post-ReKon Assessment: Table 2 shows selected results from the post-ReKon assessment. The first two (ability of a knowledge unit to satisfy knowledge needs, and relevance of the knowledge unit) were assessed based on a knowledge unit that the participants randomly chose. Many participants suggested that meta-data (e.g. “further explanation may be needed in the templates. ... to have a better
understanding”) would be useful. One suggested that a “...quick view feature that open it up in a tiny thumbnail to view” would let users locate appropriate information quickly. Several commented that some knowledge units need to be more granular as “some of the "chunks" ... should possibly be re-worked to make them easier to understand.” A few felt that it was more helpful “to have one complete document to look (at) and ...then fill in the sections that are relevant to our project.

Table 2. Post-ReKon Assessment

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Outcome (N=29)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Knowledge Needs satisfied by a Knowledge Unit (1 – All; 5 – Very few)</td>
<td>2.71 (SD = 1.01)</td>
</tr>
<tr>
<td>2. Relevance of Knowledge Unit (1 – All relevant ; 5 – None relevant)</td>
<td>2.82 (SD = 1.02)</td>
</tr>
<tr>
<td>3. Size of Knowledge Unit ( 1 – Too Small ; 5 – Too Long)</td>
<td>2.65 (SD=0.93)</td>
</tr>
<tr>
<td>4. Number of Knowledge Units in a Cell (1 – Too Many ; 5 – Too Few)</td>
<td></td>
</tr>
<tr>
<td>Instance 1 – Several units</td>
<td>4.27 (SD = 0.64)</td>
</tr>
<tr>
<td>Instance 2 – Some units</td>
<td>3.27 (SD = 1.1)</td>
</tr>
</tbody>
</table>

Together, these provided formative suggestions that we continue to incorporate. The results are also encouraging because they allow operationalizing rules such as size of knowledge unit (Criterion 3), and number of knowledge units (Criterion 4).

6. Conclusion

The results suggest that it may be possible to address pathologies such as unnecessary reuse [10] created by excessive emphasis on knowledge codification. This is particularly relevant for building systems-of-systems where knowledge needs tend to be unique and difficult to anticipate. We have described the ReKon platform (from a KM, instead of ME perspective). It has been implemented as a prototype and populated with the help of more than 1,200 Project Templates contributed by four IT consulting organizations. The platform is underpinned by a ReKon model that is simple and extensible. Formative evaluation shows that ideas underlying ReKon are likely to be valuable; further work continues to improve the platform. At the workshop, we hope to report on its ongoing progress and obtain feedback from colleagues.

Acknowledgement. The work reported has been funded by the National Science Foundation under award numbers 722112 and 722141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF). We acknowledge the assistance of Matthew Peters, Ben Karchner and Alex Pearre in Coding and Data Analysis. We also thank the Four IT Consulting Organizations who provided us their project-related templates.

References

27. Tuomi, I.: Data is more than knowledge: implications of the reversed knowledgehierarchy for knowledge management and organizational memory. (1999)
DATA AND REVENUE MANAGEMENT USING TIERED STORAGE

Anna Ye Du1, Sanjukta Das1, Ram Gopal2, Ram Ramesh1
1SUNY Buffalo, 2University of Connecticut
yedu@buffalo.edu, sdsmith4@buffalo.edu, ram@business.uconn.edu, rramesh@buffalo.edu

Abstract

Information lifecycle management is of critical importance to organizations with burgeoning data. As different groups of data assume different priorities within a firm, tiered storage emerges as a necessary and viable mechanism for data management. We consider the case of online video downloads and model data as a network where relationships between data nodes capture the impact on content usage and browsing habits. We devise a bi-level programming approach to solve the buyer’s and the seller’s problems by taking into consideration both direct and indirect hits to the specific data nodes. We model the seller as a revenue maximizer while the buyer is interested in maximizing the overall hits, especially revenue generating data content. We provide a detailed analysis of the choices made by these two entities and provide a roadmap for the empirical analysis to be accomplished in future.

Keywords: Information lifecycle management, tiered storage, data management, data networks

1. Introduction and Motivation

Tiered storage management is increasingly becoming a critical component of Information lifecycle management. This trend is driven by the exponential growth in data and significant variations in data priorities based on factors such as recency of data, frequency of use, relevance of the data to recent events of importance to the firm or other such factors. Tiered storage management necessitates moving groups of data between more expensive high quality online storage and a far less expensive option of archival storage. A recent survey (Castagna 2009) finds that the percentage of those respondents that “implemented a tiered storage architecture (excluding tape libraries)” increased from 49% in 2008 to 59% in 2009. The survey also shows that 42% of the respondents chose “classifying data so that it’s sent to the right tier” as their “biggest pain point related to their tiered storage system”. Indeed this is the focus of our research. Industry reports (such as Lebeck 2004, Arrington 2006 and others) do discuss policies for classifying data for tiered storage architectures. However, they fail to take into account the cost and Quality of Service (QoS) implications inherent in that trade-off. Our research fills this important lacuna.

We consider the case of online video downloads which have a tremendous storage appetite and whose relative importance of specific video content changes over time. Early academic research (Little and Venkatesh 1995 and Brubeck and Rowe 1995, among others) in developing the technology for Video on Demand (VOD) proposed the use of hierarchical architecture of tiered storage, where media objects are stored permanently on tertiary storage devices, and either moved to video servers on demand or delivered directly to viewers depending on the popularity of the videos. In this paper, we are motivated by the emerging business model of storage grids market. Enabled by such a market, a VOD provider can purchase storage capacity from a storage grids supplier based on demand. This buying practice requires the VOD provider to include a new factor, the storage pricing on various tiers, into its decision making on how to allocate the media objects in the tiered storage grids.

Further, we model data as a network where relationships between data nodes capture the impact on content usage and browsing habits. To the best of our knowledge, this has not been considered in the past literature. But this is important as the decision to place a data object in a tier hinges on its status and relationship vis-à-vis other data objects. Some of the data nodes (a node is analogous to a video in our setting) are revenue generating. We devise a bi-level programming approach to solve the buyer’s and the seller’s problems by taking into consideration both direct and indirect hits to the specific data nodes. We
model the seller as a revenue maximizer while the buyer is interested in maximizing the overall hits to its contents, especially revenue generating content. We provide a detailed analysis of the choices made by these two entities and provide a roadmap for the empirical analysis to be accomplished in future.

2. Model Setup

We consider a storage grid with two tiers, both online. Tier I has a faster average response time of $t^I$, and Tier II has a lower average response time of $t^II$ per GB. The storage grid provider, called the seller hereafter, offers storage in Tier II for free (this is done primarily for ease of exposition and our analysis continues to hold for cases where Tier II storage also has a price) and asks a price of $\pi$ for a unit of capacity in Tier I. The VOD provider’s data network is a graph $G = (V, E)$. The VOD provider is referred to as a buyer in our analysis. We denote the set of nodes as a vector $(v_1, v_2, ..., v_p)$. Each node $v_k$ needs capacity $c_k$ for storage. Each node $v_k$ can generate $h^I_k$ and $h^{II}_k$ hits by itself when it is stored in Tier I and Tier II respectively. This takes into account QoS degradations that might happen as a result of pushing the content to a tier of poorer quality. Note that these are direct hits coming from specific searches performed by the end users. We also model indirect hits as follows. When there is a recommendation link from node $i$ to node $j$, we say that there is a probability $p_{ij}$ that a user would browse node $j$ if s/he browses node $i$. The link is illustrated as below in Figure 1.

![Figure 1: Indirect Hits between Data Nodes](image)

Let $H_i$ be the total number of hits to node $i$, and $h_i$ be the number of hits generated by node $i$ itself. Then $H_i = h_i + \sum_{j \in V} H_j p_{ji}$. The above expression ignores cycles in the data network. Let $x_i$ denote the percentage of $c_i$ stored in Tier I. Then $1 - x_i$ is the percentage of $c_i$ stored in Tier II. Then $H_i = h^I_i x_i + h^{II}_i (1 - x_i) + \sum_{j \in V} H_j p_{ji}$. If an object has to be stored in one tier in its entirety, then $x_i$ is binary. When $x_i = 1$, node $i$ is stored in Tier I, otherwise Tier II.

2.1. The Buyer’s Problem

We model three kinds of VOD providers, based on their objectives. The objective of the first kind of buyer is to maximize total hits on its website, akin to YouTube. The second type maximizes revenue in the absence of a budget constraint while the third type does so under a budget constraint. An example of a VOD who maximizes revenues earned from the distribution of videos is Amazon VOD. We put forward the decision problem for all three cases and discuss their respective implications below.

The first type of buyer (YouTube) faces the following optimization problem:

Max $\sum_{i \in V} H_i$

s.t. $H_i = h^I_i x_i + h^{II}_i (1 - x_i) + \sum_{j \in V} H_j p_{ji}$

$\sum_{i \in V} c_i x_i \leq C$

$0 \leq x_i \leq 1, \forall i \in V$

134 19th Workshop on Information Technologies and Systems
where $C$ is the budget constraint faced by the buyer for buying storage. Without this budget constraint, the buyer would put all its data objects in Tier I and the problem becomes trivial.

The second type of buyer (Amazon VOD, no budget constraint) has the following optimization problem:

Min $\sum_{i \in V} \pi_i x_i$

s.t. $H_i = h^I_i x_i + h^H_i (1 - x_i) + \sum_{j \in V} H_j p_{ji}$

$\sum_{i \in V} H_i \geq H$

$0 \leq x_i \leq 1, \forall i \in V$

where $H$ is the constraint on the desired total number of hits that the buyer would like to attract from the viewers. This is a constraint pertaining to QoS. Without this QoS constraint, the buyer would simply put all its data objects in Tier II and the problem becomes trivial again.

For the third type of buyer (Amazon VOD, budget constrained), we consider $S, S \subseteq V$, as a set of revenue-generating nodes with $r_i$ as the unit price for node $i$. The problem for the buyer is as follows:

Max $\sum_{i \in S} H_i r_i - \sum_{i \in V} \pi_i x_i$

s.t. $H_i = h^I_i x_i + h^H_i (1 - x_i) + \sum_{j \in V} H_j p_{ji}$

$0 \leq x_i \leq 1, \forall i \in V$

Here we do not have a separate QoS constraint. However, the QoS factor is accounted for in the first term of the buyer’s net profit maximizing objective function, $\sum_{i \in S} H_i r_i$. The net profit is determined by gross income less gross cost. Gross income is the income from the revenue-generating nodes. Thus the buyer has the incentive to increase the hits to those nodes by putting them and the nodes linking to them in Tier I. However, the gross cost of storing nodes in Tier I, $\sum_{i \in V} \pi_i x_i$ in the objective function, may limit the buyer to only putting some of those nodes in Tier 1. As this type of buyer has a number of interesting nuances to its problems, we analyze the different ways to price such buyers in Section 2.2.

2.2. Pricing Alternatives

First, consider the possibility of lump-sum transfers from the buyer to the seller. When the seller charges a lump sum price, he can design it in such a way that the buyer would put all its objects in Tier I and thus pay for all its storage demand. We study this problem by first identifying its lower bounds. The buyer’s worst case is when $x_i = 0, \forall i \in V$. Let $z = \sum_{i \in S} H_i r_i - \sum_{i \in V} \pi_i x_i$. Then the buyer’s worst case is that the buyer’s revenue is $W = z(x_i = 0, \forall i)$. On the other hand, the seller’s problem can be rewritten as:

$$\max \pi \sum_{i \in S} c_i x_i$$

s.t. $\sum_{i \in S} H_i r_i - \sum_{i \in V} \pi_i x_i = W + \text{(premium)} = W'$

Note that since the seller charges a lump sum price, its decision making variable is $\pi \sum_{i \in V} c_i x_i$. Then the seller’s problem can be further simplified as follows:

$$\max \sum_{i \in S} H_i r_i - W'$$
\[ s.t. \quad H_i = h_i^l x_i + h_i^u (1 - x_i) + \sum_{j \in \mathcal{J}} H_j p_j \]

\[ 0 \leq x_i \leq 1, \quad \forall i \in V \]

The buyer’s optimal solution becomes \( x_i^* = 1, \quad \forall i \in V \).

The solution for the lump sum price is no longer optimal for the seller when a unit price is charged instead for the storage on Tier I. A counter example to this is that a buyer will always put isolated less popular nodes in Tier II. The pair of the seller’s and the buyer’s problems becomes a bi-level programming problem when deriving unit prices. In the seller’s problem, the decision variable is the unit price \( \pi \) for storage in Tier I, and the buyer’s allocation variables \( x_i \)'s are given parameters to that problem. In the buyer’s problem, the decision variable is the allocation variable \( x_i \) and the seller’s unit price \( \pi \) are given parameters. To get an understanding of how to solve such a bi-level programming problem, we first take a look at a simple problem with just one node. When there is only 1 node, the buyer’s problem is

Max \[ H_1 r_1 - \pi c_i x_i \]

s.t. \[ H_1 = h_1^l x_1 + h_1^u (1 - x_1) \]

\[ 0 \leq x_1 \leq 1 \]

Substituting \( H_1 = h_1^l x_1 + h_1^u (1 - x_1) \) into the buyer’s objective function, we get

\[ \max(h_1^l r_1 - h_1^u r_1 - \pi c_i) x_1 + h_1^u r_1. \] When \( h_1^l r_1 - h_1^u r_1 - \pi c_i > 0 \), the buyer would put all of node 1 in Tier I, i.e. \( x_1 = 1 \), otherwise, the buyer would put all of node 1 in Tier II, i.e. \( x_1 = 0 \). Then the maximum \( \pi \) that the seller could charge is when \( h_1^l r_1 - h_1^u r_1 - \pi c_i = 0 \), i.e. \( \pi^* = (h_1^l - h_1^u) r_1 \). Accordingly, the buyer’s solution is \( x_1^* = 1 \).

For a network of nodes, similarly, the buyer’s objective function can be rewritten as \( \sum_{i \in \mathcal{I}} f_i(\pi) x_i \), for example, \( f_1(\pi) = h_1^l r_1 - h_1^u r_1 - \pi c_i \) for the single-node case above. Consider another example. Suppose the buyer’s objective function is \( \sum_{i \in \mathcal{I}} f_i(\pi) x_i = (5 - \pi) x_1 + (24 - 3\pi) x_2 + (12 - 2\pi) x_3 \), then the seller’s revenue under different unit prices are as in Table 1.

<table>
<thead>
<tr>
<th>( \pi ) ($/GB)</th>
<th>Seller’s revenue ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

In general, \( \sum_{i \in \mathcal{I}} f_i(\pi) x_i \) is a non-monotonic function. Suppose there exists a critical value of \( \pi \) that makes \( f_i(\pi) x_i = 0 \) for some \( i \). Then between two adjacent critical prices, the function \( \sum_{i \in \mathcal{I}} f_i(\pi) x_i \) is linear and either monotonically increasing or decreasing. This feature enables us to compare the value of \( \sum_{i \in \mathcal{I}} f_i(\pi) x_i \) at all these critical prices, and find the global solution.
2.3. Impact of Popularity and Other Factors on Choice of Tiers

To get a better understanding of how the buyer allocate nodes among the Tiers based on both the popularity factor and the economic factor, including pricing and object sizes, we study a simplified network structure with two nodes, one of which is revenue-generating.

![Two Node Network Example](image)

Figure 2: Two Node Network Example

Then, \( H_1 = h_1^I x_1 + h_1^{II} (1 - x_1) \) and \( H_2 = h_2^I x_2 + h_2^{II} (1 - x_2) + H_1 p_{12} \)

\( = h_1^I x_2 + h_2^{II} (1 - x_1) + (h_1^I x_1 + h_1^{II} (1 - x_1)) p_{12} \). If \( S = \{1\} \), the buyer’s objective function is \( H_1 r_1 - \pi x_1 - \pi x_2 = ((h_1^I - h_1^{II}) r_1 - \pi c_1) x_1 + h_1^{II} r_1 - \pi c_2 x_2 \). Then the optimal pricing for the seller is determined by \( (h_1^I - h_1^{II}) r_1 - \pi c_1 = 0 \) \( \Rightarrow \pi^* = \frac{(h_1^I - h_1^{II}) r_1}{c_1} \). The buyer’s allocation decision is \( x_1 = 1 \) and \( x_2 = 0 \). If \( S = \{2\} \), the buyer’s objective function is \( H_2 r_2 - \pi c_1 x_1 - \pi c_2 x_2 = ((h_1^I - h_1^{II}) p_{12} r_2 - \pi c_1) x_1 + ((h_2^I - h_2^{II}) r_2 - \pi c_2) x_2 + h_2^{II} r_2 + h_1^{II} p_{12} r_2 \). Then the two critical points for the seller’s prices are \( (h_1^I - h_1^{II}) p_{12} r_2 - \pi c_1 = 0 \) \( \Rightarrow \pi'' = \frac{(h_1^I - h_1^{II}) p_{12} r_2}{c_2} \), and \( (h_2^I - h_2^{II}) r_2 - \pi c_2 = 0 \) \( \Rightarrow \pi''' = \frac{(h_2^I - h_2^{II}) r_2}{c_2} \).

If \( \pi'' \geq \pi''' \), \( \frac{h_1^I - h_1^{II}}{h_2^I - h_2^{II}} p_{12} \geq \frac{c_1}{c_2} \) \( (1) \)

Then at \( \pi'' \), the buyer’s allocation decision is \( x_1 = 1 \) and \( x_2 = 0 \), and the seller’s revenue is \( \pi'' c_1 x_1 + c_2 x_2 = (h_1^I - h_1^{II}) p_{12} r_2 \); and at \( \pi''' \), the buyer’s allocation decision is \( x_1 = 1 \) and \( x_2 = 1 \), and the seller’s revenue is \( \pi''' (c_1 x_1 + c_2 x_2) = (h_2^I - h_2^{II}) r_2 \left( \frac{c_1}{c_2} + 1 \right) \). If \( (h_1^I - h_1^{II}) p_{12} \geq (h_2^I - h_2^{II}) \left( \frac{c_1}{c_2} + 1 \right) \),

\( \frac{h_1^I - h_1^{II}}{h_2^I - h_2^{II}} p_{12} \geq \frac{c_1}{c_2} + 1 \) \( (2) \)

The optimal price \( \pi^* \) should either be equal to \( \pi'' \) or \( \pi''' \). Putting (1) and (2) together, when \( \frac{h_1^I - h_1^{II}}{h_2^I - h_2^{II}} p_{12} \geq \frac{c_1}{c_2} + 1 \), the seller’s optimal price is \( \pi'' \), and when \( \frac{c_1}{c_2} \leq \frac{h_1^I - h_1^{II}}{h_2^I - h_2^{II}} p_{12} < \frac{c_1}{c_2} + 1 \), the seller’s optimal price is \( \pi''' \). If \( \pi'' < \pi''' \), i.e.

\( \frac{h_1^I - h_1^{II}}{h_2^I - h_2^{II}} p_{12} < \frac{c_1}{c_2} \) \( (3) \)
Then at \( \pi^* \), the buyer’s allocation decision is \( x_1 = 1 \) and \( x_2 = 1 \), and the seller’s revenue is \( \pi^* (c_1 x_1 + c_2 x_2) = (h_1 - h_1^I) p_{12} \left( \frac{c_1}{c_2} + 1 \right) \); and at \( \pi^{**} \), the buyer’s allocation decision is \( x_1 = 0 \) and \( x_2 = 1 \), and the seller’s revenue is \( \pi^{**} (c_1 x_1 + c_2 x_2) = (h_2 - h_2^I) p_{12} \left( \frac{c_1}{c_2} + 1 \right) \). If \( (h_1^I - h_1^I) p_{12} \left( \frac{c_1}{c_2} + 1 \right) \geq (h_2^I - h_2^I) \frac{c_1}{c_2} \),

$$\frac{h_1^I - h_1^I}{h_2^I - h_2^I} p_{12} \geq \frac{1}{1 + c_2 / c_1} \quad (4)$$

Putting (3) and (4) together, when \( \frac{1}{1 + c_2 / c_1} \leq \frac{h_1^I - h_1^I}{h_2^I - h_2^I} p_{12} < \frac{c_1}{c_2} \), the seller’s optimal price is \( \pi^* \), and when \( \frac{h_1^I - h_1^I}{h_2^I - h_2^I} p_{12} < \frac{1}{1 + c_2 / c_1} \), the seller’s optimal price is \( \pi^{**} \). Conditions (1) – (4) therefore give us the optimal pricing policy for the seller under this scenario.

Note that when \( S = \{2\} \), the condition \( \frac{h_1^I - h_1^I}{h_2^I - h_2^I} p_{12} \geq \frac{c_1}{c_2} + 1 \) implies that the hits from node 1 (non-revenue generating) are so high and the storage size for node 2 (revenue generating) is so large, that it is optimal for the seller to set a high price to induce the buyer to buy Tier I storage only for node 1 rather than setting a lower price to incentivize Tier I storage purchase for both nodes. In other words, sometimes it is optimal for the buyer to put a non-revenue generating node in a tier with higher QoS at a higher cost, and drive traffic to a revenue-generating node stored in a Tier with lower QoS at a lower cost.

3. Conclusion and Future Research

We proposed an analytical framework for pricing tiered storage architecture in the presence of networked data and have provided insights pertaining to factors such as QoS and the popularity of connected nodes. We utilized a bi-level programming approach to solving this problem. We are currently conducting an empirical study to validate our analytical discussion by analyzing the network of Amazon VOD. Data collection pertaining to this is underway and will be bolstered by trace-driven simulations and sensitivity analysis pertaining to the various parameters such as transition probabilities across pages. We will have the empirical analysis completed and ready to be presented at the Workshop.

References:


DATA CENTER WORKLOAD CONSOLIDATION BASED ON TRUNCATED SINGULAR VALUE DECOMPOSITION OF WORKLOAD PROFILES

Thomas Setzer
Technische Universität München
setzer@in.tum.de

Abstract

In today’s data centers, typically thousands of enterprise applications with varying workload behaviors are hosted. As energy usage is one of the key cost drivers in data centers, workload consolidation is increasingly used to host multiple applications on a single server, sharing and multiplexing a server’s capacity over time. To minimize the number of required, energy-consuming servers, IT managers need to decide which applications should be combined on which server. For that purpose, typically application workload levels are predicted for a planning period such as a month in a defined granularity (e.g., over 5-minute intervals). Then integer programs are used to minimize the amount of required servers, while for each interval constraints ensure that the aggregated workloads of applications assigned to a server must not exceed a server’s capacity. As such problems are NP-hard and computationally intractable for data centers with hundreds of servers and fine-grained workload data, approximations are applied to find at least a good solution, often abandoning the chance to find the optimum. In this paper we propose a novel approach based on applying Singular Value Decomposition to the workload data to reduce the dimensionality of the problem by capturing workload features in order to make the problem computationally tractable. We interpret the coordinates of the time-series projections along the first right singular vectors as indicators for workload levels and complementarities and propose a model to solve the consolidation problem with these few indicators only. We evaluate the model using industry data.

Keywords: Workload Management, Virtualization, Consolidation, Multivariate Data Analysis, SVD

1. Introduction

In today’s data centers, typically thousands of enterprise applications like ERP modules or databases with complex and varying workload behaviors are hosted. Server virtualization based workload consolidation is increasingly used to raise server utilization levels. Server virtualization allows for hosting multiple virtual servers (or virtual machines (VM)) including application plus underlying operating system on a single physical server (target). A target’s capacity is then shared and multiplexed over time amongst VMs. As specifically energy costs account for 30–50% of the total data center operation costs, IT managers need to decide which VMs should be combined (consolidated) on which target to minimize the number of required, energy-consuming targets (Filani et al. 2008).

Existing consolidation decision models typically first predict VM workload over a planning period such as a day or a month in a defined granularity (e.g., maximum workload over 5-minute intervals) based on past observations. Usually, workloads show recurring patterns on a daily or weekly basis. For example, payroll accounting is performed at the end of the week, while workload of an OLAP application has a daily peak in the morning when managers access their reports. More advanced consolidation models leverage these cycles by first determining representative e.g. daily VM workload profiles describing the workloads expected in each time interval (e.g. maximum over a 5-minute interval) for different resource types such as CPU and memory. Second, an integer program (IP) attempts to assign those VMs together on targets whose workloads are complementary, i.e. peaks are at different times of the day to smoothen and increase overall target workload in order to reduce the number of targets. One constraint per resource and interval ensures that the aggregated workload of VMs assigned to a target must not exceed the target’s capacity.
As an example consolidation model we describe the Static Server Allocation Problem considering Varying Workload (SSAPv) model published by Bichler et al. (Bichler et al. 2006). Suppose that we are given $J$ VMs $j, j \in \{1, \ldots, J\}$ to be hosted by $I$ or less target servers $i, i \in \{1, \ldots, I\}$. Different types of resources $k, k \in \{1, \ldots, K\}$, may be considered and each target has a certain capacity $s_{ik}$ of resource $k$. $y_i$ is a binary decision variable indicating if target $i$ is used, $c_i$ describes the cost of a target (e.g. energy costs over a planning period), and the binary decision variable $x_{ij}$ indicates which VM is allocated to which target. The planning period is divided into time intervals indexed by $t=\{1, \ldots, \tau\}$. Let further $u_{jkt}$ describe how much capacity $j$ requires of $k$ in $t$. Techniques how to derive $u_{jkt}$ are described in (Bichler et al. 2008). The resulting consolidation problem is formulated in equation (1).

$$
\begin{align*}
\text{min} & \sum_i c_i y_i \\
\text{s.t.} & \quad \sum_{i \in I} x_{ij} = 1 \quad \forall j \leq J \\
& \quad \sum_{j \in J} u_{jkt} x_{ij} \leq s_{ik} y_i \quad \forall i \leq I, \forall k \leq K, \forall t \leq \tau \\
& \quad y_i, x_{ij} \in \{0, 1\} \quad \forall j \leq J, \forall i \leq I
\end{align*}
$$

The objective function minimizes server costs, the first constraint ensures that each VM is allocated exactly once, and the second constraint ensures that the aggregated workload of multiple VMs does not exceed a target’s capacity.

As the problem is strongly NP-hard, it cannot be solved optimally for larger instances, in particular as the number of constraints grows linearly with $\tau$ multiplied by $K$ (Garey et al. 1979). Therefore, usually intervals are coarsened to reduce the number of constraints, e.g., hourly workload intervals are used by taking maxima over 12 5-minute intervals. However, coarsening intervals reduces the problem size but also the ability to exploit workload complementarities and therefore impacts the solution quality. Additionally, there are inherent inefficiencies with time intervals: for a certain period of time an interval might be coarse for VMs with volatile workload during that period, while it might be unnecessarily fine-grained for other VMs with smoother workload during that period (v.v. in other periods).

In this paper we propose a consolidation model based on multivariate statistics to circumvent the computational problems resulting from fine-grained workload data as well as the trade-off between fine- and coarse-grained time resolution. In section 2 we apply truncated Singular Value Decomposition (SVD) to the original workload matrix (with workload time series as row vectors) and project the time series onto data points in the space spanned by the first right singular vectors of the SVD. In section 3 we give an interpretation of these points, where coordinates along the first right singular vector indicate workload levels, and subsequent coordinates indicate workload complementarities. Subsequently we develop a mathematical model to solve the consolidation problem with only the few indicators derived. In section 4 we evaluate the model using industry data. Related work is discussed in section 5. In section 6 conclusions are drawn and future work is discussed.

2. Dimensionality Reduction of Workload Data

The $K\tau$-dimensional tuples describing VM workload time series can be represented as points in a $K\tau$-dimensional space, where a VM workload level of a resource $k$ in $t$ is indicated as a coordinate along a dimension $(k, t)$. To reduce dimensionality, these points need to be projected into an $E$-dimensional space so that $E<<K\tau$. We apply truncated SVD for that purpose as it is applicable to non-square and not full-ranked workload matrices and fast SVD approximations exist.
Let $R$ be the original $J$ by $K$ matrix of $J$ VMs, with time series (per $k$) of length $\tau$ as row-vectors (elements of $R$ are $u_{jk\ell}$). Let $U \Sigma V^T$ be $R$’s factorization using standard SVD, where $R$’s singular values $\sigma_k$ in $\Sigma$ are ordered in non-increasing fashion, $U$ contains the left singular vectors, and $V^T$ contains the right singular vectors. The intuition of this factorization is that the right singular vectors are the axis of a new space, the associated singular values are scaling factors for these axis, and the row-vectors in $U$ represent the coordinates of VM workloads in the new space. As an illustration, consider workloads $u_{jk\ell}$ of VMs $j$ for $\tau=2$ (maximum during daytimes ($t=1$) and nighttimes ($t=2$)) for one resource $k=1$. The resulting data points are shown in Figure 1.

![Figure 1: Workload Time Series Projections](image)

For each $j$, coordinates $u_j^1$ are calculated by perpendicular projection of the points onto $u^1$, the first right singular vector. These coordinates show the best 1-dimensional approximation of the data because $u^1$ captures as much of data variation as possible by one direction. VM coordinates $u_j^2$ regarding the second right singular vector $u^2$ ($u^2 \perp u^1$) captures the maximum variance after removing the projection of the data along $u^1$ (in this 2-dimensional example, $u^2$ captures all of the remaining variance; in general the number of singular vectors equals $R$’s rank).

What makes SVD practical is that variation below a particular threshold $E$ can be ignored as the singular values associated with the right singular vectors sort them in “goodness” order with explained variation from most to least. This is the idea of truncated SVD where only the first $E$ column vectors of $U$ and the first $E$ row vectors of $V^T$ are considered.

3. Dimensionality Reduction of Workload Data

A Principal Direction of Workload and Capacity Limits

As a regression line running through the data points, $u^1$’s direction approaches dimensions $(k, t)$ with high aggregated workload where overload situations are likely to appear. Hence, we interpret $u^1$ as major workload direction. Consider the scenario depicted on the left-hand side of Figure 2.

![Figure 2: Major Workload direction and Complementarity](image)
The coordinate \(u^j_1\) of a VM \(j\) along \(u^1\) fully describes \(j\)’s workload as \(\sigma_j = 0 \forall e > 0\). Here, the problem can be solved as a variant of the bin-packing problem, with \(u^j_1\) as object sizes, and the projection of the target capacity limits as bin sizes. We determine the bin sizes as follows: for each of the \(K \times \tau\) original dimensions the capacity constraint for resource \(k\) of target \(i\) is \(s_{ik}\) (for all \(t\)). Hence, for each target we obtain hyperplanes which form a convex polyhedron indicating its capacity limits (the grey lines in the pictorials show the hyperplanes of a target \(t=1\); a rectangle in the 2-dimensional case).

As a point (e.g. the aggregated \(u^j_1\)-coordinates of combined VMs) outranging this rectangle indicates target overload, the capacity limit is the intersection point \(P_i\) of \(u^1\) and a hyperplane of target \(i\). Hence, \(i\)’s bin size equals \(|P_i|\), the Euclidian norm of the vector from origin to \(P_i\).

**B Workload Complementarities**

However, usually \(\sigma_2, \sigma_3, \ldots\) are non-zero and \(u^1\)-based workload estimation is inaccurate. In the scenario depicted on the right-hand side of Figure 2, additional VMs A-D with equal \(u^j_1\) but different \(u^j_2\) coordinates are considered. \(u^j_2\) captures “distances” to \(u^1\), i.e., \(u^1\) workload approximation errors. Workload in \(t=1\) (\(t=2\)) is overestimated (underestimated) by \(u^j_1\) for VMs with positive \(u^j_2\) (A and B); the opposite for VMs with negative \(u^j_2\) (C and D). Hence, when combining VMs with positive and negative \(u^j_2\) - for example A and D – A’s workload is overrated in intervals where B’s workload is underestimated and v.v., which reduces a target’s aggregated \(u^j\) workload estimation error. For example, when combining A and D, A’s higher workload in \(t=1\) is compensated by D’s lower workload in \(t=1\) in order to avoid overload due to \(u^1\) approximation errors. Therefore, VMs \(j\) with \(u^j_2\)-coordinates that add to zero can be considered as complementary.

**C Model Formulation**

On the other hand, combining VMs with positive \(u^j_2\) like A and B on a target further intensifies \(u^1\) workload underrating in \(t=2\). Let \(z_{\Omega}\) be the absolute sum over \(u^j_2\) values of VMs assigned to a target. To avoid target overload when using a bin-packing formulation, \(z_{\Omega}\) must be added to the aggregated \(u^j_1\) coordinates of assigned VMs to ensure sufficient capacity in all time intervals. The resulting IP entitled Thin Workload Consolidation Model (ThinWCM) is shown in equation (2).

\[
\min \sum_{i \in I} c_i y_i \\
\text{s.t.} \\
\sum_{i \in I} x_{ij} = 1 \quad \forall j \leq J \\
\sum_{j \in J} (u^j_1 x_{ij}) + z_{\Omega} \leq |P_j| y_i \quad \forall i \leq I \\
\sum_{j \in J} (u^j_2 x_{ij}) - z_{\Omega} = 0 \quad \forall i \leq I \\
y_{ij}, x_{ij} \in \{0,1\} \quad \forall j \leq J, \forall i \leq I \\
z_{\Omega} > 0 \quad \forall i \leq I
\]

Again, the objective is to minimize server costs and the first constraint ensures that each VM is allocated exactly once. The second constraint ensures that the aggregated \(u^1\) workload estimate of VMs assigned to a target plus \(z_{\Omega}\), their aggregated \(u^2\)-coordinates do not exceed the target’s capacity limit. The third constraint calculates \(z_{\Omega}\) required in the second constraint (replacing the third constraint by two linear constraints is straightforward). Although not shown for reasons of clarity, variation along \(u^3, u^4, \ldots\) is
considered just as variance along $u^2$. For each $u^e$, $2 < e < E$, we introduce a constraint to determine $z_{ie}$, and add each $z_{ie}$ to $z_{2}$ in the second constraint. As a conservative estimator for the remaining variance in $u^{E+1}$, $u^{E+2}$, … for each $j$ we add the sum of $j$’s absolute coordinates $u^e_i$, $e > E$, to $u^E_j$, and ignore further complementarity in $u^e$, $e > E$.

4. Experimental Analysis

From a professional data center we obtained data describing 5-minute averages for CPU and memory workload of hundreds of VMs over multiple months. Most of these workloads exhibit rather deterministic daily patterns without a significant trend. Thus, we consider daily workload profiles. We consider scenarios from 10 to 160 VMs to be consolidated, where each scenario consists of 10 arbitrarily chosen VM subsets. We assumed targets with identical capacity.

In our experiments we analyze ThinWCM regarding solution quality (no. of targets) and computational time to solve the model using SSAPv as a benchmark. We set $E=5$ as over 90% of total workload variance was described in the directions of the first 5 eigenvectors. As SSAPv with 5-minute intervals (SSAPv 5 minute) is intractable for larger problem instances, we solve SSAPv additionally for 1-hour (SSAPv hourly) and 1-day intervals (SSAPv daily). For 1-hour intervals we derive maxima over 12 5-minute intervals and for 1-day intervals workload is represented by its maximum. As mentioned before, the ability to exploit complementarities decreases with increasing interval lengths. Using SSAPv daily for each scenario an upper bound $I$ for the number of targets was obtained; solutions of SSAPv 5 minute indicated lowest bounds. Calculations were performed on a 2.4Ghz Intel Duo, 4GB RAM using R for SVD calculation and Lp_solve v.5.5 (with defaults) as solver. Figure 3 shows the aggregated experimental results.

![Figure 3: Aggregated Experimental Results](image)

In the diagram on the left-hand side, for each model variant the average number of required targets per scenario is displayed as bar height. Missing bars indicate that no solution was computable within four hours. In most experiments, ThinWCM derived the optimal solution and dominated SSAPv hourly (and obviously SSAPv daily). The graph on the right-hand side of Figure 3 shows, on a logarithmic scale, the average computational time per scenario required to solve a model (for ThinWCM, time to compute the SVD is included). The exact model SSAPv 5 minute could be solved for up to 100 VMs, while SSAPv hourly could be solved for up to 120 VMs. SSAP daily and ThinWCM could be solved within a minute even for 160 VMs, with a much higher solution quality when applying ThinWCM instead of SSAPv daily.

5. Related Work

While there has been a lot of work on capacity planning in IS, little work has focused on efficient server consolidation. Closest in spirit to our work is the work by Bichler et al. (Bichler et al. 2008) and by Rolia et al. (Rolia et al. 2003), both use integer programs to exploit workload complementarities and statistically multiplex resources over time to minimize the amount of targets while ensuring sufficient
capacity in each time interval. They apply approximations such as time-slot coarsening and meta-heuristics such as Genetic Algorithms (GA) to make their solutions computationally tractable. (Rolia et al. 2005) and (Cherkasova et al. 2006) describe an approach based on statistical multiplexing using GA that penalize low target utilizations and target overload to minimize the number of targets. (Seltzsam et al. 2006) also forecast workload profiles to multiplex server resources. (Urgaonkar et al. 2002) analyse best-fit and worst-fit heuristics to bundle complementary services on common servers.

In contrast to our work, we did not find approaches in the literature that apply multivariate statistics like SVD to reduce the dimensionality of the consolidation problem in order to transfer and solve the problem in a low-dimensional space.

6. Conclusion and Outlook

In this paper we introduced ThinWCM, a server consolidation model based on truncated SVD of workload data to derive indicators for VM workload levels and complementarities. In first experiments with industry data, ThinWCM found the optimal solution in most cases and solved much larger problems than decision models with a comparable solution quality.

To the best of our knowledge, there is now previous work on how to apply multivariate statistics in order to solve IT problems such as server consolidation more efficiently.

In our future research we plan to evaluate larger sets of workload traces and we will explore additional heuristics for server consolidation. Furthermore, as today IT service management suffers from the complexity of handling vast amounts of high-dimensional data, we plan to apply multivariate statistics to dynamically control and visualize data center workload with a few indicators only. In particular, we plan to predict trends and detect workload anomalies that require intervention like moving a VM to another target before an anticipated overload situations occurs.

References


