

Data Mining for Modeling Chiller Systems in Data Centers

Debprakash Patnaik¹, Manish Marwah²,
Ratnesh K. Sharma², and Naren Ramakrishnan¹

¹ Virginia Tech, Blacksburg, VA 24061, USA

² HP Labs, Palo Alto, CA 94034, USA

Abstract. We present a data mining approach to model the cooling infrastructure in data centers, particularly the chiller ensemble. These infrastructures are poorly understood due to the lack of “first principles” models of chiller systems. At the same time, they abound in data due to instrumentation by modern sensor networks. We present a multi-level framework to transduce sensor streams into an actionable dynamic Bayesian network model of the system. This network is then used to explain observed system transitions and aid in diagnostics and prediction. We showcase experimental results using a HP data center in Bangalore, India.

1 Introduction

Over the last decade, data centers have grown from housing a few hundred multiprocessor systems to tens of thousands of servers in warehouse-sized buildings. However, widespread use of data centers has been accompanied with steep increases in power consumption and high costs, a matter of great concern to both owners and users of data centers. According to the EPA, US data centers have become energy hogs and their continued growth is expected to demand the construction of 10 new power plants by 2011 [1, 2]. Globally, datacenters currently consume 1–2% of the world’s electricity [3] and are already responsible for more CO₂ emissions than entire countries such as Argentina or The Netherlands.

Data centers constitute a mix of computing elements, networking infrastructure, storage systems along with power management and cooling capabilities (see Figure 1, left), all of which offer opportunities for improving energy efficiency and achieving more sustainable data centers. For instance, huge inefficiencies abound in average server utilization (believed to be at most 10–15%), and thus one approach to achieve greener IT is to use virtualization and migration to automatically provision new systems as demand spikes and consolidate applications when demand falls. Similarly, dynamic management of an ensemble of chiller units in response to varying load characteristics is another strategy to make a data center more energy efficient. There are even end-to-end methodologies proposed [5] that track inefficiencies at all levels of the IT infrastructure “stack” and derive overall measures of the efficiency of energy flow during data center operation.

A key problem is the unavailability, inadequacy, or in-feasibility of theoretical models or “first principles” methodologies to optimize design and usage of data centers.

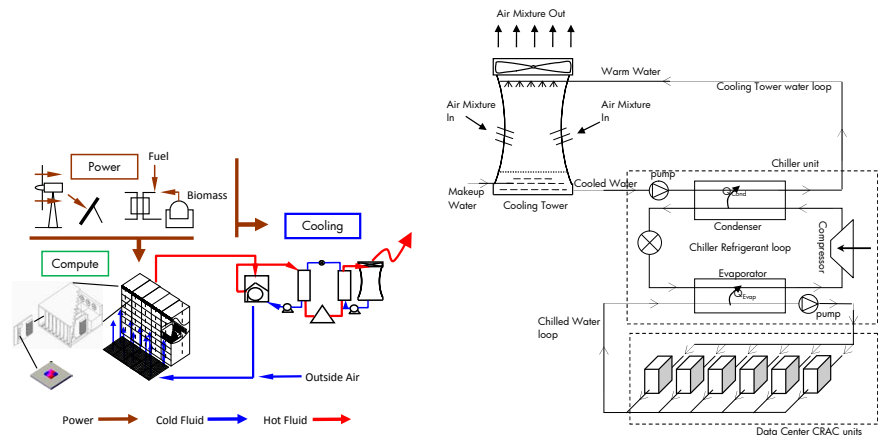


Fig. 1. (left) Elements of a data center [4]. (right) Schematic of a typical cooling infrastructure.

While some components can be modeled, (e.g., an operating curve for an individual chiller unit, a Computational Fluid Dynamics (CFD) based prediction of air flows through rows of racks for static conditions, temperature profiles to reveal hot spots in clusters [6]), these methods are computationally intensive and difficult to deploy in real time. Furthermore, what is missing is the ability to understand complex interactions between the multiple components of data centers. Needless to say, optimizing each component separately does not necessarily lead to a greener installation on the whole. Data-driven approaches to data center management are hence more attractive: by mining sensor streams from an installation, we can obtain a real-time perspective into system behavior and identify strategies to improve efficiency metrics.

2 Data Centers

A typical data center is organized as racks of computing equipment arranged as rows. A large data center could contain thousands of racks occupying several tens of thousands of square feet of space. The cooling infrastructure of a data center typically looks as shown in Figure 1 (right).

The focus of this paper is the chiller ensemble in the data center cooling infrastructure. Each chiller unit is composed of four basic components, namely, evaporator, multi-stage centrifugal compressor, economizer and water-cooled or air-cooled condenser. Liquid refrigerant is distributed along the length of the evaporator to absorb enough heat from the water returning from the data center and circulated through the evaporator tubes to vaporize. The gaseous refrigerant is then drawn into the compressor. Compressed gas passes from the multi-stage compressor into the condenser. Cooling tower water circulated through the condenser tubes absorbs heat from the refrigerant, causing it to condense. The liquid refrigerant then passes through an orifice plate into the economizer. Flashed gases enter the compressor while the liquid flows into the evaporator to complete the circuit. Some terms used in the context of chillers are given below.

IT cooling load. This is the amount of heat that is generated (and thus needs to be dissipated) at a data center. It is approximately equivalent to the power consumed by the equipment since almost all of it is dissipated as heat. It is commonly specified in kilowatts (KW).

COP. The coefficient of performance (COP) of a chiller unit indicates how efficiently the unit provides cooling, and, it is defined as the ratio between the cooling provided and the power consumed, i.e.,

$$\text{COP}_i = \frac{L_i}{P_i} \quad (1)$$

where L_i is the cooling load on the i th chiller unit and P_i is the power consumed by it.

Chiller utilization. This is the percentage of the total capacity of a chiller unit that is in use. It depends on a variety of factors, mainly, the mass flow rate of water that passes through a chiller and the degree of cooling provided, that is, the difference between the inlet and outlet temperatures ($T_{in} - T_{out}$). For a particular T_{out} , an administrator can control the utilization at a chiller through power capping or by changing the mass flow rate of water.

Ensemble of Chiller Units The number of chiller units required depends on the size and thermal density of a data center. While one unit may be sufficient for a small data center, several units operating as an ensemble may be required to satisfy the cooling demand of a large data center. Figure 2 shows an ensemble of chiller units that collectively provide cooling for a data center. Out of the five units shown, three are air-cooled while the remaining two are water-cooled.

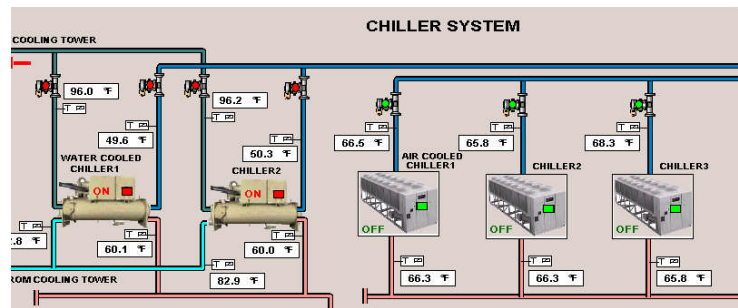


Fig. 2. Five chiller units work in tandem to provide cooling for a large data center.

Operational Challenges Although operating curves for individual chiller units exist, no model is available for operation of an ensemble, especially one consisting of heterogeneous units. Additionally, shift and/or drift of response characteristics with time further complicate their management. The operational goals are to satisfy the cooling requirements while minimizing the total power consumption of the ensemble and maximizing the average lifespan of the units. While multiple factors impact the lifespan of

a chiller unit, an important one is: rapid and large oscillations in utilization value. High amplitude and frequent variations in utilization due to varying load or some failure condition result in decreased lifespan, and, thus, need to be minimized.

3 Prior Work

Earlier work in data center management adopt a “first principles” approach by conducting detailed CFD-based modeling of air and temperature flows [7]. The computational infeasibility of such large-scale simulations coupled with the relative ease of gathering real-time data from sensors has led others, most notably within HP labs, to alternatively explore the use of data analysis techniques. For instance, modeling of rack-level temperature data specifically in relation to CRAC layout has been undertaken in [8, 9]. Optimization opportunities at multiple levels of smart center architecture have also been studied in [5]. More recent work [10] focuses on sensor data mining to identify anomalous and deviant behavior. Other related work includes the InteMon system from CMU [11, 12] that dynamically tracks correlations among multiple time series [13] but only studies a few time series. While all these works constitute important strides in analytics, in order to support high-level knowledge discovery capabilities, we must raise the level of abstraction at which we study and infer patterns from sensor data streams. In particular, we must generalize from correlations and anomalies in time series datasets to being able to model the entire complexity of relationships underlying system variables [14] and, in this manner, promote life-cycle modeling of IT systems. More importantly, sustainability considerations and prior domain knowledge about life-cycle modeling must drive and inform the design of data mining algorithms. Other methods [15] have been proposed to capture the dynamics of time-series by modeling them as Markov chains and then applying clustering to the models to discover different cluster dynamics.

We have recently demonstrated the usefulness of data mining in characterizing utilization of the cooling infrastructure of a data center [16]. In this study, our primary goal was to link the multivariate numeric time series data, e.g., temperatures recorded from chiller units, power utilization etc. to sustainability characterization. We decomposed this goal into cluster analysis and event encoding to obtain abstract symbolic representation of the data, followed by motif mining, and sustainability characterization. Thus, this approach uses motif patterns as a crucial intermediate representation to aid in data reduction. The present work builds upon these successes to develop an actionable model of the data center’s cooling infrastructure. Similar motivations can be found in [17] who use a (dynamic) Bayesian network model of a production plant to detect faults and anomalies. Unlike this work, however, we show how such networks can be efficiently learned using frequent episode mining over sensor streams. Furthermore, our networks are defined over clustered representations of the time series data rather than raw sensor signals.

4 Methods

Our proposed framework for analyzing the cooling infrastructure is illustrated in Figure 3. First, data reduction is performed in a few different ways. The raw time series

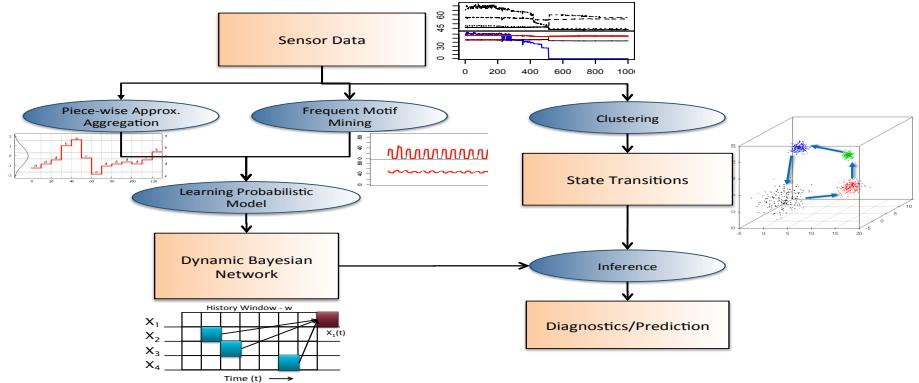


Fig. 3. Framework for modeling data center cooling infrastructure.

data is compressed using piece-wise aggregate approximation following discretization using equal frequency bins. This helps capture the average dynamics of the variables. A higher-order aspect of the time series involves repeating or oscillatory behavior. This is inferred by mining frequently repeating motifs or patterns in the time series, as described in our earlier work [16]. This information is integrated with the average behavior by recording the windows in which a time series exhibits motif patterns. Finally, it is also pertinent to mine relationships involving “control” actions that were taken during the operation of the chiller units. In this paper, we focus on ON/OFF actions.

A graphical model in the form of a Dynamic Bayesian Network (DBN) is learnt from the above data. This model captures the dependencies between the variables in the system over different time lags. Here, we focus on the target variable of utilization and seek to identify suitable parents for modeling in the DBN. Unlike classical methods to learn BNs [18], we demonstrate a powerful approach to learn DBNs by creating bridges to the frequent episode mining literature [19]. To apply the learned DBN, we define states of the system by clustering together the combined utilization of the chiller units. This allows the operation of the chiller ensemble to be represented as a sequence of state transitions. We now use the dependencies and (conditional) independency relationships found in learning the graphical model to find the most probable explanation behind the state transitions. This framework can then be applied for activities like data center diagnostics, performance improvement, load balancing, and preventive maintenance.

4.1 Motif Discovery

In this work, time series motifs are defined as contiguous sub-sequences that follow the same pattern and repeat several times over the entire time series. Each time series $T = \langle t_1, \dots, t_m \rangle$ in our data is an ordered collection of real values that a particular variable takes at different sampling times. In order to discover motif in a time series, we first discretize the time-series and use the discretization levels as symbols to encode the time series. This sequence of discrete symbols is then analyzed to detect the change points. We raise the level of abstraction further by doing a run-length encoding of the symbol sequence and noting where transitions from one symbol to another occur. This

gives us a sequence of transition events for input to serial episode mining as illustrated below:

$$\begin{aligned} \text{Symbol Sequence} &: d d d a a a a c c c c b b b b b \\ &\Downarrow \\ \text{Event Sequence} &: \langle (d-a, 5), (a-c, 10), (c-b, 14) \rangle \end{aligned}$$

Frequent episode mining is now conducted over this sequence of transitions. We look for serial episodes with inter-event constraints that repeat sufficiently often. The structure of a serial episode is given below.

$$\langle E_1 \xrightarrow{(0, d_1]} E_2 \dots \xrightarrow{(0, d_{n-1}]} E_n \rangle \quad (2)$$

Here E_1, \dots, E_n are the event-types in the episode α . Each pair of event-types in α is associated with an inter-event constraint. For example the pair $E_1 \rightarrow E_2$, is associated with $(0, d_1]$ such that in an occurrence of α event E_2 occurs no later than time d_1 of event E_1 . The mining process follows the level-wise procedure ala *Apriori*, i.e., candidate generation followed by counting. The candidate generation scheme is based on matching the $n - 1$ size suffix of one n -node frequent episode with the $n - 1$ size prefix of the another n -node frequent episode at a given level to generate candidates for the next level. The same frequent episode mining algorithm is presented in our earlier work [16]. There it is applied in a slightly different context.

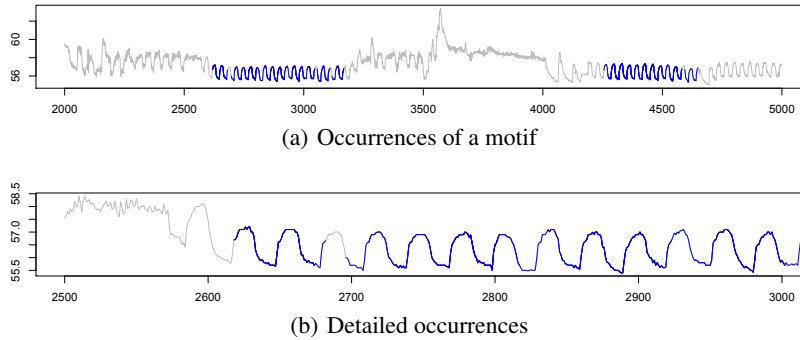


Fig. 4. Illustration of a repeating motif in the utilization of a water cooled chiller unit

4.2 DBN Structure Learning

In this section we discuss probabilistic graphical models and provide the intuition behind building a graphical model for the variables in the chiller ensemble. We focus on Bayesian networks, specifically dynamic Bayesian networks where the random variables are time-stamped.

A Bayesian network (BN) is a graphical model denoted by $B = (G, P)$ where G is a directed acyclic graph (DAG) and P is a set of conditional probability distributions.

The graph $G = (V, E)$ consists of a set of nodes V representing the random variables $\{X_1, \dots, X_N\}$ in the system and a set of directed edges E . Each directed edge in E , denoted by $i \rightarrow j$, indicates that random variable X_i is a parent of random variable X_j . The conditional probabilities in P are used to capture statistical dependence relationships between child nodes and parent nodes. In particular, given a random variable X_i in the graph, we denote by $\text{par}(X_i)$ the set of random variables that are parents of X_i . The statistical dependence between X_i and its parent nodes $\text{par}(X_i)$ is captured by the conditional probabilities $P(X_i|\text{par}(X_i))$.

To model a discrete-time random process $\mathbf{X}(t), t = 1, \dots, T; \mathbf{X}(t) = [X_1(t)X_2(t) \dots X_N(t)]$ as studied here, we use the more expressive formalism of dynamic Bayesian networks. In particular, we focus on time-bounded causal networks, where for a given $w > 0$, the nodes in $\text{par}(X_i(t))$, parents for the node, $X_i(t)$, belong to a w -length history window, $[t - w, t)$. Note that parent nodes cannot belong to the current time slice t for $X_i(t)$.

This assumption limits the range-of-influence of a random variable, $X_k(t)$, to variables within w time-slices of t and also indicates that the random variables $X_i(t)$ and $X_j(t)$ are conditionally independent given their corresponding parent sets in the history window. Further, we also assume that the underlying data generation model is stationary, so that joint-statistics can be estimated using contingency tables.

The learning of network structures involves learning the parent set, $\text{par}(X_i(t))$, for each $X_i(t)$, $i = 1, \dots, N$. In this work we assume that there are no spurious independencies in the data, i.e., if a random variable $X_j(t - \tau)$ is a parent of $X_i(t)$, then the mutual information $I(X_i(t); X_j(t - \tau)|\mathcal{S})$ conditioned on a subset $\mathcal{S} \subseteq \text{par}(X)$ is always greater than zero. Moreover time-bounded causality enables us to learn the parents of each node $X_i(t)$ independent of any other node in the same time slice. We use a greedy approach to learn the parent set of each node $X_i(t)$. And proceed by adding a node which has the highest conditional mutual information to the parent set of $X_i(t)$ as shown in Eqn 3:

$$\text{par}^{i+1}(X_i(t)) \leftarrow \text{par}^i(X_i(t)) \cup \arg \max I(X_i(t); X_j(t - \tau)|\text{par}^i(X_i(t))) \quad (3)$$

The search is continues until the number of nodes in the parent set is k (where k is an user-defined parameter) or the conditional mutual information drops to zero. The structure learning is then followed by maximum likelihood estimation of the conditional probability tables.

4.3 State Transition Modeling

In analyzing complex systems it is usual to try and summarize the operating characteristics into a finite set of states where the systems spends most of its time. The utilization information of the chiller ensemble is important from both the aspect of efficiency and sustainability. Therefore we define the states of our system in terms of the combined utilization of all the chiller units.

In order to obtain a finite set of states, we first perform a k -means clustering on the utilization vectors and use the cluster labels as symbols to encode the multi-variate time series of the combined utilization. Thus the multivariate series of utilizations is now encoded as a single long state sequence.

Over the state sequence the points of interest are the times where the system moves from one state to another. The exact transition times can be affected by the clustering scheme used but on the average they capture changes in the operating characteristics.

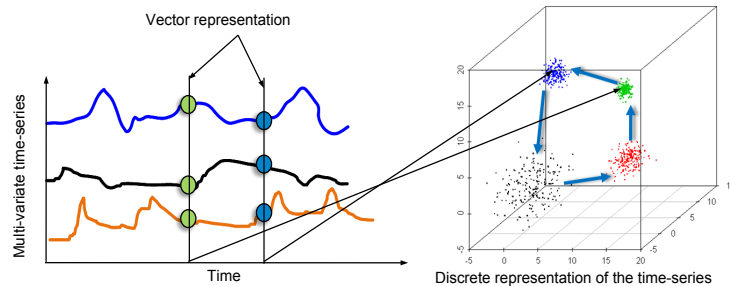


Fig. 5. Illustration of rendering multivariate time series data as a sequence of state-transitions using clustering. The arrows are drawn between corresponding vectors on the time series plot and the 3-d plot.

An interesting question to ask now is: What causes the state transitions to take place. In the context of our modeling, this question translates to: what factors in the system cause the utilization of the chiller to go up or down? This can be answered by observing the changes that take place in the system around the state transitions. The problem with this approach is the lack of availability of sufficient data around the change points. We propose an alternative approach where we decompose the question by asking what causes each of the changes in the chiller units. From the graphical model already learnt we know the set of variables in the system that directly affect the utilization. These variables belong to the parent set of the utilization nodes. The task of finding the most probable causes for a transition amounts to evaluating the following probability distribution:

$$\Pr(\mathcal{S}) = \prod_{i \in \text{UtilizationVariables}} \Pr(\mathcal{S}_i | X_i(t) = a_i, X_i(t-1) = b_i) \quad (4)$$

where $\mathcal{S}_i = \text{par}(X_i(t)) \setminus X_i(t-1)$ and $\mathcal{S} = \cup \mathcal{S}_i$. The most likely values that \mathcal{S} takes can be considered the best explanation of the transition. Here t is the time at which a state transition occurs, a_i, b_i are the discrete values the utilization variable takes before and after the state-transitions. These can be approximated by the cluster centers of each cluster used to define a state.

5 Results and Discussion

We applied our methods to chiller data obtained from a large HP production data center covering an area of 70,000 square feet. Its cooling demand is met by an ensemble of five chiller units. The ensemble consists of two types of chillers: three are air-cooled and the remaining two are water-cooled. The data is collected from October 21, 2009 to November 13, 2009, totaling over 576 hours of data consisting of over 47 variables.

Table 1. A few important system variables in the chiller ensemble data.

System Variable	Description
AC_CH(<i>i</i>)_EVAP_E_TEMP	Temperature of water entering air-cooled chiller <i>i</i>
MAIN_HEADER_TEMP_1_F	Water temperature at distribution header
WC_CH(<i>i</i>)_IN_TEMP	Temperature of water entering water-cooled chiller <i>i</i>
WC_CH(<i>i</i>)_OUT_TEMP	Temperature of water leaving water-cooled chiller <i>i</i>
AC_CH(<i>i</i>)_RLA	Percentage utilization of air-cooled chiller <i>i</i>
WC_CH(<i>i</i>)_RLA	Percentage utilization of water-cooled chiller <i>i</i>

5.1 Graphical Model

For learning the graphical models, the raw time series data was aggregated over windows of 15 minutes and augmented with results from motif mining conducted over the same data. The motif information is in form of a binary valued sequence for each time series. The binary values indicate whether or not motif occurrences were seen in each 15 minute window. A finer grained representation of the motifs will be explored in the future.

Since utilization of a chiller mostly determines its energy consumption, we have shown that portion of the learned dynamic Bayesian network in Figure 6. Three air-cooled chiller utilization variables: AC_CH1_RLA, AC_CH2_RLA, AC_CH3_RLA; and two water-cooled ones: WC_CH1_RLA and WC_CH2_RLA are shown together with their parents and grandparents. In the DBN, almost every node has itself from the previous time slice as one of its parents, which is expected since the data is continuous valued time series data. In order to assist in creation of a causal Bayesian network, partial domain information was incorporated through constraints such as limiting nodes that could become parents. Three utilization variables show direct dependency on water temperature. This is, again, expected since the aim of the chiller ensemble is to maintain water temperature close to the set point.

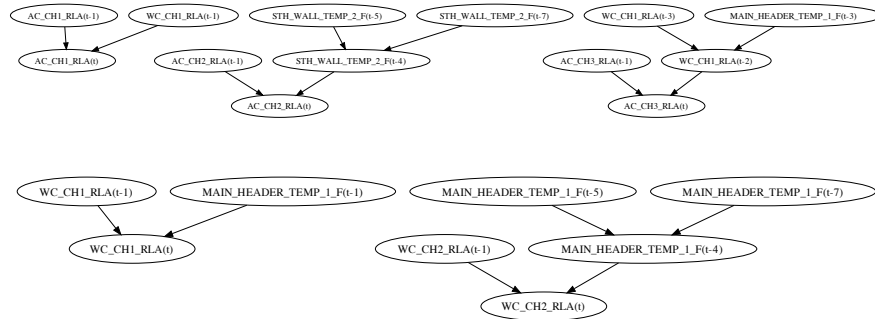


Fig. 6. Graphical Model learned from the discretized time series data. Shown are the parent nodes for only the utilization variables of the five chiller units.

More interesting relationships are revealed by two air-cooled units (1 and 3) that directly depend on the utilization of water cooled chiller. This would indicate that these

two chillers ramp up or down based on how the water cooled chiller (which has higher capacity) is operating.

5.2 State Transitions

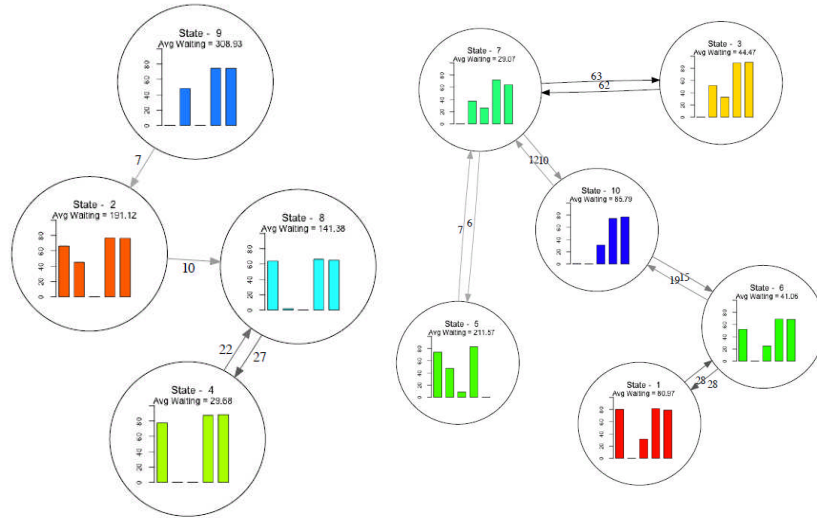


Fig. 7. Transitions in the state-based representation of the combined utilization of the chiller ensemble. Edges representing 5 or fewer transitions are not shown. And edges representing more than 5 transitions are labeled by the actual counts.

The operational states of the system, as shown in Figure 7, are obtained by clustering together the combined utilization of all the chillers in the ensemble. Here we use k -means clustering with $k = 10$ and organize the states in decreasing order of total power consumption, which is computed using Equation ???. Since the objective of this work is to operate a chiller ensemble more energy efficiently, we color code the system states to reflect their power consumption. The color red indicates the maximum power consumption (about 3298 KW) while the color blue indicates the least consumption (2148 KW). Also shown, for each state, are the average utilization values of the five chillers as a histogram with first three (starting from the left) being air-cooled ones and the last two being water-cooled units. The time spent by the system in each state is also listed (maximum in state 9, while least in state 1). The arrows show the transitions between states, with the gray-scale indicating the frequency of the transition (darkest implying most often).

The system states are mainly characterized by the number and kind of chiller units operating, their utilization levels and power consumption. Some states are quite similar, e.g., states 3 and 7, which have the same chiller units operating with not much difference in their utilization levels. Other states show marked difference, e.g., state 10 has three

chillers operating (one air-cooled and two water-cooled) while state 6 has four working units (two air-cooled and two water-cooled). Note that consequently state 10 consumes less power than state 6. A data center chiller operator will be interested in understanding the variables that influence transitions from state 10 to 6.

Transition: State 10 → State 6 When the chiller ensemble makes a transition from state 10 to state 6, air cooled chiller-1 turns on. The graphical model can be queried to provide the most probably explanation for this change. Using the model from Section 5.1, we estimate values of parent of utilization node when these state transitions take place, as listed in Table 2. Note that utilization levels of the parent (WC_CH1_RLA) are high when these transitions take place. These and other similar insights would facilitate more energy efficient management of the chiller resources.

Table 2. List of most-likely value assignments of the parent-set of node AC_CH1_RLA i.e. utilization of air-cooled chiller 1.

$\text{par}(X_i(t))$	Delay	Value	$\text{Pr}(\text{par}(X_i(t)) X_i(t), X_i(t-1))$
WC_CH1_RLA	1	(75.37, 77.38]	0.27
WC_CH1_RLA	1	(72.36, 75.37]	0.18
WC_CH1_RLA	1	(77.38, 79.71]	0.18

In the future, we also plan to incorporate system alarms as nodes in the graphical model with the objective of discovering the variables that most significantly influence a particular alarm. This causal inference would provide valuable information to a data center chiller operator on corrective steps needed to handle the alarm. Furthermore, operator actions would be added to the model to enable discovery of dependencies between state transitions and such actions. This would allow an operator to query what actions could be taken (if any at all) to move the system from a less energy efficient state to a more efficient one.

6 Conclusions and Future Work

We have presented an expressive framework for reasoning about data center chillers using data mining techniques. Our methods raise the abstraction level from raw sensor streams to networks of relationships between higher order variables. While in this paper we have focused on determining the model structure and dependencies, in the future, we plan to build a diagnostic and advisory system. We will enrich the time series with alarm, sustainability and user action information with the objective of achieving the following goals: (1) inference of the probable cause of a chiller alarm, e.g., an alarm indicating that the water supply temperature has consistently been above the set point; (2) provide best course of action to handle an alarm, with actions learnt from past data; (3) monitor sustainability metrics, such as energy use or carbon footprint, and recommend state changes towards more sustainable operation. Although our current studies have focused on one particular subsystem of a data center, we posit that our methods are general and can be targeted toward other complex system modeling tasks.

References

1. Koomey, J.: Power conversion in servers and data centers: A review of recent data and developments. In: Applied Power Electronics Conference. (Feb 25 2008)
2. Kaplan, J.M., Forrest, W., Kindler, N.: Revolutionizing data center efficiency. Technical report, McKinsey Report (2008)
3. Vanderbilt, T.: Data center overload. *New York Times* (June 8 2009)
4. Watson, B., et al.: Integrated design and management of a sustainable data center. In: InterPACK '09: Proceedings of ASME InterPACK, New York, NY, USA, ASME (July 2009)
5. Sharma, R., et al.: On building next generation data centers: Energy flow in the information technology stack. In: Compute 2008, Bangalore, India (Jan 2008)
6. Cameron, K.W., Pyla, H.K., Varadarajan, S.: Tempest: A portable tool to identify hot spots in parallel code. In: ICPP '07: Proceedings of the 2007 International Conference on Parallel Processing, Washington, DC, USA, IEEE Computer Society (2007)
7. Patel, C., et al.: Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications. In: ASME IPACK'01, Kauai, HI (July 2001)
8. Bautista, L., Sharma, R.: Analysis of environmental data in data centers. Technical Report HPL-2007-98, HP Labs (June 2007)
9. Sharma, R., et al.: Application of exploratory data analysis (eda) techniques to temperature data in a conventional data center. In: ASME IPACK'07, Vancouver, BC (2007)
10. Marwah, M., Sharma, R., Bautista, L., Lugo, W.: Stream mining of sensor data for anomalous behavior detection in data centers. Technical Report HPL-2008-40, HP Labs (May 2008)
11. Hoke, E., Sun, J., Faloutsos, C.: Intemon: Intelligent system monitoring on large clusters. In: 32nd International Conference on Very Large Data Bases. (Sep 2006) 1239–1242
12. Hoke, E., et al.: Intemon: Continuous mining of sensor data in large scale self infrastructures. *Operating Systems Review* **40**(3) (2006) 38–44
13. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming pattern discovery in multiple time series. In: 31st International Conference on Very Large Data Bases. (2005) 697–708
14. Morchen, F.: Unsupervised pattern mining from symbolic temporal data. *ACM SIGKDD Explorations* **9**(1) (2007) 41–55
15. Ramoni, M., et al.: Bayesian clustering by dynamics. *Mach. Learn.* **47**(1) (2002) 91–121
16. Patnaik, D., et al.: Sustainable operation and management of data center chillers using temporal data mining. In: Proc. 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining. (2009) 1305–1314
17. Nielsen, T.D., Jensen, F.V.: Alert systems for production plants: A methodology based on conflict analysis. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (2005) 76–87
18. Friedman, N., Nachman, I., Pe'er, D.: Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In: 5th Conf. on Uncertainty in Artificial Intelligence UAI (1999). (1999) 206–215.
19. Patnaik, D., Laxman, S., Ramakrishnan, N.: Discovering excitatory networks from discrete event streams with applications to neuronal spike train analysis. In: IEEE Intl. Conf. on Data Mining (ICDM'09). (December 2009)