VERSATILE CPS FOR DATA CENTER COOLING PROPOSAL

John E. Rollinson
jerollin@andrew.cmu.edu
CMU 15-624

J Su
CMU 15-424

6 December 2019

ABSTRACT

Energy usage optimization is a very important area of research in tech industry. Different companies and research groups have applied different types of models and optimization algorithms to achieve desired results across a variety situations. However, many of these research focus on optimizing their cost function without directly ensuring the safety of their final control system on the servers in the data center. Our project focus on system that regards the safety of equipment and operation as most paramount feature and we have constructed a permissive controller model and prove its safety in order to then leverage `ModelPlex` to produce a runtime safety system that could be used in conjunction with a wide-variety of "optimized" controllers.

We have constructed two models and have formally proved in dynamic logic that the model with simpler physics and constraints always has a safe temperature under operation. The second model takes on many more complex physics and more real life factors like the efficiency of heat transfer and thermal properties of instruments in the system, and we have proved that the energy inside the system is always lower than max allowed energy. In this proof system, our model is ready to take on different optimization algorithm for the cooling controller and still proceed to prove the safety of the system base on property of the optimization algorithm.

***K*eywords** CPS,Energy Usage Optimization

1 Introduction

1.1 Motivation

Optimizing energy usage in data centers and for industrial machinery cooling has been an active area of research due to the large amounts of energy and money they require. In 2014, data centers in the U.S. consumed an estimated 70 billion kWh, representing about 1.8% of total U.S. electricity consumption[1]. And in 2018, about 163 billion kWh are used for industrial indoor cooling, which accounts for 12% of US industrial electricity consumption. Therefore, reducing the energy requirements for cooling is a major focus as it can account for a substantial portion of the energy usage of a data center and can be optimized without modifying the workload present in the data center.

While optimizing for energy efficiency, however, it is important to account for why this cooling is necessary in the first place. Excessive temperatures can cause permanent damage to multiple components of the servers in a datacenter. In particular, excessive heat at either the processor or hard drive of a server increases its immediate risk of failure as well as its risk of failure in the future, even after the excessive heat has been removed. In light of this property, maintaining a certain max temperature at each server is often an important part of minimizing expenditures or meeting the requirements of service-level agreements.

1.2 Related Works

In trying to control cooling costs, research generally focuses on introducing control decisions that lower (or shift) the overall energy consumption of the system when compared to a naïve solution. In particular, Yeo et al's research into reducing cooling costs for small to medium size data centers in [1] is representative of one approach where the performance of individual components is deliberately limited to reduce total cooling requirements. In contrast Mansousakis et al provide a different approach where they mix air intake sources to achieve optimal cooling efficiency.

A more recent approach, however, has been to optimize for cost. Wang, Wang, and Zhang present a model and evaluation for minimizing energy costs based off temporary changes in the price per watt [2]. In particular, they use multiple thermal masses to over-cool when energy is cheap and under-cool when it is more expensive. This allows them to shift their energy usage in time and reduce the overall cost of cooling.

2 Proposed Model

One trend throughout current work about data center cooling is the focus on energy efficiency through complex modeling (computational fluid dynamics) or machine learning. However, these efforts do not seek to formally prove properties of their models and controllers. Our goal is to model and provide a permissive runtime check system that ensures a monitored controller operates safely during its operation.

2.1 Assumptions

Given current state of formal proofs regarding data center cooling, we start with several simplifying assumptions. Although these assumptions lead to a significantly less accurate model then state-of-the-art models that rely on computational fluid dynamics, our model provides a foundational starting point for developing more accurate models over time.

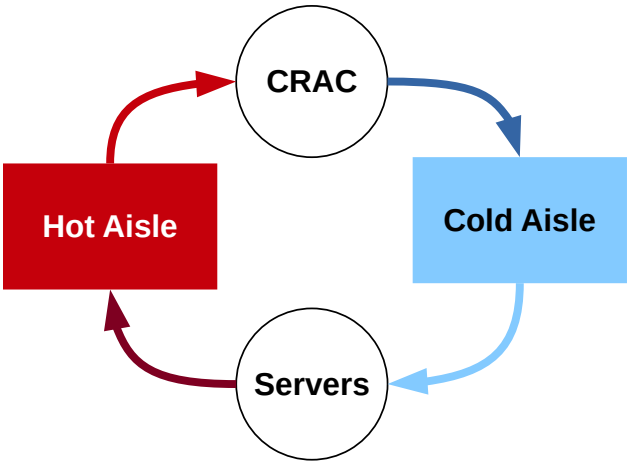

Figure 1: High-level view of the model.

Most of our assumptions revolve around factors that influence energy density and transfer within the system. Specifically, we assume:

1. The only source of energy transfer between components and the system as a whole occur at the cooling unit and the servers. This ignores energy transfer that naturally occurs between components since there are no perfect insulators, but it provides a tractable starting point for developing proof approaches.
2. The energy of our components is uniformly distributed throughout the entire mass of a component. While this is demonstrably false and the primary focus of CFD models for datacenters, a functioning model under this assumption could be adapted to be more accurate in the future. Specifically, some models address this issue by essentially splitting the datacenter into three separate regions where this assumption holds. Therefore, we focus only on this simple case.
3. The collective server load can be modelled as a single discrete heat source. This assumption simplifies the distributed nature of this modelling problem with respect to the individual of servers to a single instance. While this is another area in which state-of-the-art techniques focus on improving model accuracy and optimizing efficiency, we determined that this was also more appropriate for future work. Particularly, much of the focus on the distributed heat source model is focused on how to schedule and distribute the computational load which we viewed as an orthogonal problem to our proposed approach.
4. The specific heat of air in the system is constant. More accurate models account for the enthalpy of the air which is affected by multiple factors that we have not incorporated into our system (e.g. pressure and humidity). While this is more accurate, the air currently in the datacenter tends to remain relatively consistent as long as the cooling unit recirculates air rather than using external air (which may have a very different humidity level). This is why we focus on safety in that use case.
5. The mass of air in the aisles is constant and negligible elsewhere. Specifically, we assume that the cooling system causes air to circulate into and out of each component at the same rate and that there are no flow bottlenecks in the system.
6. The CRAC coefficient of performance is constant over a given range and those parameters are known at system start time. Although a linear or quadratic model with respect to the outlet temperature of the CRAC would be more accurate, the performance difference is generally small and this assumption simplifies finding an initial proof approach.

7. The power consumed by the servers is instantaneously transferred to the air passing from the cold aisle to the hot aisle. This assumption ignores the thermal mass of the various server components but should be safe as long as the energy capacity of the air greatly exceeds that of the servers. The more difficult part of this assumption is that the rate at which the servers can dissipate their heat is a function of the airflow through them and temperature of the cold aisle. We address this concern by having our safety parameter, T_s, be a function of those variables and implicitly accounting for this issue.

2.2 Control Model

We approach control in two phases: the environment's control and the cooling systems control decisions. The environment controls two inputs to the system: the external air temperature and the current server load. In contrast, the cooling system has control over the circulating fan's speed, the current power consumption of the cooling unit, and the intake source of the system (i.e. recirculated air from the hot aisle or external air).

2.2.1 Environmental Control

We model the environment's "control" through bounded non-determinism. This widens the model's applicability while still providing some tractability for our initial proofs. Specifically, we bound the rate of change of the external air temperature and only allow the server load to change within discrete time steps. We leave these controls strictly to non-determinism since a real-world system is likely to have little control over either of these conditions since external temperature is controlled by local weather, and the server load is tied to scheduling decisions that we have placed out of scope for this project.

2.2.2 Cooling Control

As mentioned above, we focus on three control decisions available to a cooling system. The recirculation rate we model as a continuous decision between 0 and some max flow rate for the system. This represents the max flow of air mass through the system. The power consumption of the cooling unit indirectly controls the rate at which energy is removed from the system during an evolution of our physics model. Finally, the intake source of the system is the final control aspect that we model. Given the complications associated with correctly modeling external air intake, we require that the system is always capable of recirculating air. This specifically addresses to potential problems: sudden changes in external air temperature cause by weather anomalies as well as significant humidity deviations between internal and external air that would invalidate some of our simplifying assumptions.

2.3 Physics Model

Our physics model fundamentally focuses on the amount of energy stored in the hot and cold aisles of the datacenter. Specifically, we model the datacenter as two thermal masses (the aisles) and how energy is added and removed by circulating air through the system. In this system, energy leaves the cold aisle and enters the hot aisle by circulating air from the cold aisle through the servers, where energy is added to the system, into the hot aisle. Energy enters the cold aisle by circulating air from either the outside or the hot aisle through the cooling unit and into the cold aisle. The rate of circulation for both processes is the same.

All of the temperature values we use in the model are in Kelvin in order to allow us to easily assume that temperature values are strictly positive.

2.3.1 Constants

Within our model, we use several constants to capture external inputs to the system that should not change while the system is running.

- The specific heat capacity of air within the system, $c > 0$.
- The maximum time between control iterations, $step > 0$.
- The total mass of air in the cold aisle, $m_c > 0$.
- The total mass of air in the hot aisle, $m_h > 0$.
- The power draw of the servers when completely idle, $P_{idle} > 0$.
- The maximum power draw of the servers, P_{max} such that $P_{max} > P_{idle}$.
- The coefficient of performance for the CRAC unit, a_0.
- The minimum and maximum outlet temperatures at which the coefficient of performance is accurate, COP_{min} and COP_{max}.
- The maximum power draw of the CRAC's compressor unit, P_{cmax}.
- The maximum flow rate the CRAC is capable of generating, f_{max}.
- The maximum safe temperature of the cold aisle, T_{scmax} such that $T_{scmax} > T_c$ at all times.
- The maximum safe temperature of the hot aisle when the CRAC is running at full power, T_{shmax}. Specifically, $T_{shmax} = \frac{a_0 P_{cmax}}{c f_{max}} + T_{scmax}$ which captures the concept that at the maximum energy point for the system, the CRAC unit must be capable of cooling the hot aisle's air down to the cool aisle's current temperature.

2.3.2 Variables

We divide the variables in our system into three broad categories: environment variables, state variables, and controller variables.

Environment Variables We use the term environment variables to denote those parts of the system which the CRAC is neither directly nor indirectly influencing. Our model's two environment variables are P, the current power consumption of the servers, and t the current time for the system. While some datacenter cooling models would consider P to be a controllable variable, we have placed that outside the scope of our model.

State Variables We use the term state variables to denote the variables in the model which are indirectly controlled by the controller. In effect these are the temperatures of the hot and cold aisles, T_h and T_c, respectively. In particular, these variables are never assigned to and should only change according to our physics model during a run of the hybrid program described later.

Controller Variables These variables are the one that the controller directly manipulates (or is calculated directly from one of those variables). The two primary controller variables are the compressor power level, P_c, and the air circulation flow rate, f. Each of these variables directly influence the secondary variables of the outlet temperature, T_o determined by flow rate and compressor power, and current maximum safe temperature of the cold aisle, T_s which is determined by f. As a simplifying factor in our proof, we force the controller to always run the circulation fans at maximum speed which effectively make f constants and $T_s = T_{scmax}$.

2.3.3 Underlying Equations

The foundation of our model is modeling how energy changes in the hot and cold aisles over time. As a starting point for modeling energy in the system, we model the energy, E, in each aisle as

$$E = c\,m\,T \tag{1}$$

where c is our constant for the specific heat capacity of air, m is the appropriate mass constant for that aisle, and T is the current temperature of that aisle. Since c and m are constants, the change in energy of an aisle is

$$E' = c\,m\,T' \tag{2}$$

from which we can derive

$$T' = \frac{E'}{c\,m} \tag{3}$$

which captures the aisle's change in temperature over time. This is an important reframing of the equations since it is much easier to measure temperature directly than energy.

In order to operationalize this differential equation, we need to develop a more concrete formula for E' which is the change in energy of the aisle. Based off our simplifying assumptions we can assume that energy only enters and exits each aisle due to the circulating air. Since we assume energy (and hence temperature) are evenly distributed in each aisle, we can model the rate of energy flow out of each aisle as

$$E'_{out} = -c\,f\,T \tag{4}$$

where f is the flow rate in air mass per unit time and T is the current temperature of the aisle. This also means that the incoming energy can be written as

$$E'_{in} = c\,f\,T_* \tag{5}$$

where the new variable T_* represents the temperature of the incoming air immediately before reaching the aisle.

Cold Aisle Temperature From this setup, we can now derive the specific differential equations for each aisle as well as the outlet temperature. For the cold aisle, we T and T_* are directly equivalent to T_c and T_o, respectively. From the above general equations, we can derive

$$T'_c = \frac{f(T_o - T_c)}{m_c} \tag{6}$$

to model the change in the cold aisle's temperature since we model its inflow temperature, the CRAC's outlet temperature, directly.

Hot Aisle Temperature The hot aisle is slightly different since we are not directly modelling the outlet temperature of the servers. However, by observing that power is the derivative of energy and that we assume the power consumed by the servers is transferred to the passing air as heat energy, we can revise Equation 5 to

$$E'_{in} = c\,f\,T_* + P_* \tag{7}$$

where the new variable P_* is the rate at which energy is dissipated into the flowing air. In the case of the hot aisle, this is P, the current power consumption of the servers. Putting this together yields

$$T'_h = \frac{f(T_c - T_h)}{m_h} + \frac{P}{c\,m_h} \tag{8}$$

as the differential equation that describes changes in the hot aisles temperature.

$$P := *;$$
$$?(P \geq P_{idle});$$
$$?(P \leq P_{max});$$

Figure 2: Hybrid program for server power consumption.

CRAC Outlet Temperature The final remaining temperature which we need to model is the CRAC's outlet temperature T_o. Since we assume there is no significant airmass between the hot and cold aisles, T_o will be directly related to the inlet temperature of the CRAC which is the hot aisle, T_h. Specifically, it will be

$$T_o = T_h - \frac{a_0 \, P_c}{c \, f} \tag{9}$$

because the second term is accounts for the affect that the energy extracted from the system by the compressor has on the air flowing through the CRAC. In other words, $a_0 \, P_c$ is the amount of energy per second continually distributed over air with a heat capacity of c and flow rate of f. Taking the derivative of this yields $T_o' = T_h'$ since the term for energy removed is constant with respect to the ODE which yields

$$T_o' = \frac{f(T_c - T_h)}{m_h} + \frac{P}{c \, m_h} \tag{10}$$

as the final differential equation for modelling the temperature change of the CRAC's outlet.

2.4 Hybrid Program for the System

We can formalize the model above by rewriting the equations and control aspects into the syntax of hybrid programs.

2.4.1 Environment's Control

The first part of our program captures the environment setting a new power consumption value for the servers. Figure 2 shows how we can rewrite this portion of the model to be a hybrid program. This hybrid program allows the power consumption of the servers to take on any possible value that is between the servers' minimum and maximum power consumption levels.

2.4.2 CRAC Control

The next part of our hybrid program models the CRAC controller as shown in Figure 3. The first three lines of the program capture the controller logic that upper-bounds how much power the CRAC compressor can draw by the machines limit and lower-bounds it by matching the power injected into the system by the servers. Note that this is deliberately a very conservative system in order to allow us to develop our proof approach to the system. It should also be noted that this is guaranteed to have at least one run as long as an initial constraint of the system is that $P_{max} < P_{cmax}$ which is to say that the datacenter's cooling is not over-provisioned.

The next two lines of Figure 3 set the outlet temperature according to the chosen power level. The formula in the program is a simple derivation of Equation 8. The final line of the hybrid program is another conservative bound on P_c that ensures we do not output unsafe temperatures into the cold aisle. This last line was not ideal but was required to ensure that under light loads we did not let the system shift too much energy into the cold aisle despite having the capability to keep the aisle cold enough.

$$
\begin{aligned}
&P_c := *;\\
&?(P_c \leq P_{cmax};\\
&?(a_0 * P_c \geq P);\\
&T_o := *;\\
&?(c * f * T_h - a_0 * P_c = c * f * T_o);\\
&?(T_o \leq T_s);
\end{aligned}
$$

Figure 3: Hybrid program for modelling the CRAC controller.

$$
\begin{aligned}
&t := 0;\\
&\{\\
&\quad T_c' = (f * (T_o - T_c))/m_c,\\
&\quad T_h' = (f * (T_c - T_h))/m_h + P/(c * m_h),\\
&\quad T_o' = (f * (T_c - T_h))/m_h + P/(c * m_h),\\
&\quad t' = 1\\
&\quad \& \, t \leq step\\
&\quad \& \, T_o \leq COP_{max}\\
&\quad \& \, T_o \geq COP_{min}\\
&\}
\end{aligned}
$$

Figure 4: Hybrid program for modelling the physics of the system.

2.4.3 Physics

The physics portion of our model is described by the hybrid program in Figure 4. The only significant addition from what we described earlier is the added domain constraints of $COP_{min} \leq T_o \leq COP_{max}$. This should not affect the proof of the system but would cause a resulting run-time verification system to throw an exception when the controller leaves the model's region of accuracy.

3 Proof Overview

We were unable to close our proof. Although, it is likely that are model is still incomplete in that it is probably lacking some initial conditions and loop invariants, we believe our overall approach is on the correct path to proving properties about the model.

3.1 Approach

Our general approach to the proof is to require the system to start with less than the maximal safe amount of energy and that all of the parameters make sense and show that each loop iteration does not affect our desired safety properties. Since the goal was to showed continual safety, our initial conditions, loop invariants, and final safety conditions all mirror each other closely. The major focus

of our proof approach was to identify the key differential invariants that allow the loop invariants to hold and this is where we encountered the most challenges.

Target Differential Invariants

As part of our proof approach, we targeted proving three high-level invariants that we believed would allow us to prove the overall properties of the model. Specifically, we looked to prove that the (1) overall energy in the system remained below the equilibrium point with the servers running a maximum power, (2) the hot aisle is always warmer than the cold aisle, and (3) the cold aisle stays below the safe temperature.

Total Energy Below Equilibrium This invariant captures the nature of the differential equations that there is a natural equilibrium point in the system when the servers are running at maximum power. Specifically, our invariant was

$$c\,m_h\,T_h + c\,m_c\,T_c \leq c\,m_h\,T_{shmax} + c\,m_c\,T_{scmax} \tag{11}$$

which effectively states that the current total energy in the system must be less than the energy in the system at this equilibrium point.

Cold Aisle Cooler than Hot Aisle This invariant is designed to capture the idea that the model guarantees that the cold aisle is always cooler than the hot aisle. More formally,

$$T_c \leq T_h \tag{12}$$

should always be true. Intuitively, the relations of input and output temperatures for each aisle guarantee this since the cold aisle's input is always less than or equal to the hot aisle's temperature and the hot aisle's input is always greater than the cold aisle's temperature.

Cold Aisle Cooler than Safe Temperature Point Our final target invariant was the target safety condition itself which could be formally written as

$$T_c \leq T_s \tag{13}$$

to capture that the cold aisle's temperature is always a safe temperature for the servers' inlets.

3.2 Details

The initial steps of our proof are to unfold the initial formula and apply a loop invariant that matches our desired safety conditions (basic assertions that temperature values are greater than 0 and that $T_c \leq T_s$). From there we are able to unfold the inner program and start working against the differential equations portion of our model. We then proceeded to tackle our target invariants in the order presented.

3.2.1 Total Energy Invariant

Because we started with a conservative controller that only removes energy from the system or maintains the current energy level, this invariant was currently easy to prove. Specifically, we were able to differential cut Equation 9 into the domain constraints of the ODE. This was sufficient to prove this invariant.

3.2.2 Cold Aisle Cooler than Hot Aisle

This invariant proved to be significantly more complex than the first invariant. In particular, the both aisle temperature could be converging and differential cuts and invariants are not enough. We were able to prove this invariant by introducing two differential ghosts. Specifically, we introduced

$$y' = \left(\frac{f}{2\,m_h} + \frac{f}{2\,m_c}\right) y$$
$$z' = -\left(\frac{f}{2\,m_h} + \frac{f}{2\,m_c}\right) z$$

into the ODE in order to prove $T_c \leq T_h$ by proving $(T_h - T_c)y^2 \geq 0$ and $z \cdot y = 1$. We derived the differential equation for y' by hand after closely analyzing the derivative of $(T_h - T_c)y^2$ and leveraging Equation 9 to substitute out the instance of T_o in T_c'. We were then able to use $a_0\, P_c \geq P$ from our controller's definition (see Figure 3) to further simplify and reduce the derivatives to 0. We then chose z' such that it would show $zy = 1$ was an invariant of the equations.

3.2.3 Cold Aisle Temperature is Safe

We were ultimately unable to prove this invariant and this is where our proof work became blocked. Intuitively, we tried to work through showing that if $T_o \leq T_s$ then this would establish $T_c \leq T_c$. However, while we believe these are true of the model as described we were unable to develop an approach that worked. In particular, the confounding situation for us is that $T_o \leq T_c$ is not valid. In cases where the controller has overcooled the system, there are valid states where $T_o > T_c$ and this makes it difficult to use T_o to prove a maximal value for T_c. Our intuition is that we need to leverage the hot aisle relationship and the relationship between T_o and T_h to reframe the problem (similar to what we did for $T_c \leq T_h$), but we have yet to find a set of invariants that provide traction for this.

3.2.4 Remainder

We believe that this final differential invariant is likely the hardest part of the proof remaining and that the rest of the proof should close easily after proving the overall safety consideration.

4 Discussion

4.1 Applications

We have constructed and proved two models that can be used for data center cooling system in this project. Since performance and power draw from data center is predictable and bounded,we want to have data center model that,with given controller,is able to provide a proof in dynamic logic that a system is guaranteed to be safe while running at the control efficiency.
Our system can be modified and apply to any energy circulation system that the safety of equipment and operation is paramount.
In the first model with rather simpler physics and constraints that the air entering the server racks is always safe; this can potentially used in single source and small scale cooling. For example, we can have a run-time safety system with dynamic logic inside a water cooling system for a desktop at home. While it has similar physics to data center, but we will have only one heat sink and will encounter less extremum.
The second model is a two-isle system that is similar to many current using data centers and it takes considerations of real life factors like the efficiency of heat transfer and thermal properties of instruments in the system, and we have proved that the energy inside the system is always lower than

max allowed energy.While many data-center focus on efficiency and can afford some overheating and failure of equipment, that's why there was a lack of formal dynamic proof system for the safety of data center, but as quantum computing raising in the scene and many other equipment has become too valuable to lose in the operation due to too much energy in the system, our approach can be used as a preliminary model for the guaranteed safe cooling solution.

4.2 Limitations

We have faced many challenges and limitation with the current project both in the modeling and proof process with the time we have.

1. Properly handling the delayed response between control decisions in the AC unit and impact on the server racks due to the thermal mass we model in the cold aisle. While approximations such as the linear approximation we used in our simple initial model are useful, our ultimate goal is to achieve a more precise lower bound for safety.
2. Handling the compounding affect of modeling the hot aisle as a thermal mass. In our initial attempts to model it, we found it very difficult to convert our intuitive reasoning about the relationship between hot and cold aisles. Specifically, that the temperature of the hot aisle, T_h, is always higher than the temperature of the cold aisle, T_c. Informally, we have been able to convince ourselves that this is true based off the heat transfer equations and asymptotic behavior, but we have found it very difficult to gain traction on the right invariant because of the challenge of correctly formulating the relationship between the CRAC's outlet temperature and the cold aisle. It seems that bounding this relationship properly is the key to gaining traction on proving the invariants of the more general system.
3. While our current model could potentially take on controller with similar bound and still give out safety proof for the properties, in situation that add variables and modifying the physics, our current proof might need find completely different invariant and approach to prove the safety.
4. While working on this project, we came to appreciate the complexity of modelling systems with inter-related feedback loops. In particular, the level of complexity for proving a single-aisle system is significantly less than that of a two-aisle system. Our failure to understand this complexity difference is likely a major reason we were unable to achieve our original goals as our desired goals were, in hindsight, very ambitious.

4.3 Future Work

In addition to fully proving our current model, there are several avenues for future work for our project that would greatly increase the practicality of the model. Each of these avenues improves the applicability or accuracy of the model and should allow monitored controllers more flexibility in the control choices they make.

4.4 Allow Energy into the System

Our current controller always requires the CRAC to extract at least as much energy from the system as the servers are putting into the system. This is a very conservative model that is not sustainable in the real world because we will either cool to a point where the coefficient of power for the CRAC is no longer accurate or be able to observe heat transfer through unmodeled sources. This improvement is probably the minimum requirement for making the model usable in real datacenters.

4.4.1 Multiple Temperature Zones

Computational fluid dynamic models have shown that the uniform mass of air is not an accurate assumption for most server rooms. While some controllers currently overcome this by treating the temperature of the cold aisle as the hottest observed temperature, more accurate models divide the cold aisle into three or more zones where the uniform temperature assumption is accurate enough in practice. Adding this to the model would improve its ability to effectively monitor a wider range of CRAC control units.

4.4.2 Improved Energy Model

A major source of energy savings in current datacenter cooling systems is the ability to switch between recirculating air through the CRAC or venting the hot aisle to the outside and using outside air for the CRAC's intake. Properly modeling whether this is the correct choice requires modeling the difference in enthalpy (and hence specific heat capacity) of both the inside and outside air which is affected by other variables such as pressure and humidity. Additionally, using a more accurate model for the coefficient of performance for the CRAC unit which is generally a function of the outlet temperature would further boost the accuracy of the energy model.

However, these changes would invalidate large portions of our existing proof approach as it turns several constants in our model into variables. This would likely require significant effort to develop a revised proof approach.

4.4.3 Distributed Server Model

The final area for future work that we currently see is to expand our single mass of servers model into a distributed hybrid system where the servers can operate independently. This is likely of little value in the current model but could prove interesting in the multiple temperature model briefly discussed above as it could allow different power properties in the different zones.

5 Conclusion

The datacenter cooling system we studied and modeled is a useful starting point towards developing a system model that could ensure safe datacenter operations. The single-aisle cooling model has generally been enough to maintain safe temperatures in a datacenter and we have successfully proved a simple controller that could be expanded to a more permissive controller for use as a monitor.
We have also proved that our more realistic two-aisle model of a data center cooling system at least maintains some invariants that we believe are critical to proving its ultimate safety. Although we did not meet our original goal of a full proof, we have created a starting point that could be used to bring formal verification to datacenter cooling. This verification would be allow for complimentary safety monitors for current state-of-the-art machine learning models that may fail to operate safely in unpredictable and potentially catastrophic ways.

6 Deliverables

- `KeYmaera X` **file:** Simple model (`SimpleCool.kyx`, no hot aisle) proved in `KeYmaera X`. Proof of discussed model (`DataCool.kyx`) is a work-in-progress with proof on safety energy in the system.
- `ModelPlex` **monitor:** Not created (blocked on proof).

7 Division of Work

John Rollinson and J Su performed approximately 60% and 40% of the work, respectively.

References

[1] Sungkap Yeo, Mohammad M. Hossain, Jen-Cheng Huang, and Hsien-Hsin S. Lee. ATAC: Ambient Temperature-Aware Capping for Power Efficient Datacenters. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 1–14. ACM, March 2014.

[2] Yefu Wang, Xiaorui Wang, and Yanwei Zhang. Leveraging Thermal Storage to Cut the Electricity Bill for Datacenter Cooling. In *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*, HotPower '11, pages 8:1–8:5, New York, NY, USA, 2011. ACM.