Automated control of evaporative cooling in modular data centers

QRB Labs*

July 2026

1 Introduction: Airflow

A data center hosting high performance computers can be seen as a physical process which takes in energy in the form of electricity, uses it to do work in the form of computation, and as a side effect, generates waste heat. Removing this heat is one of the fundamental challenges of data center design.

For our data centers, consisting of shipping container-size modules, we rely primarily on airflow. As air passes by the computers, it absorbs their heat and carries it away. To help this basic process, we have three mechanisms, in order of priority:

- **Fans on the individual computers** pulling in cold air on one side, and/or pushing out hot air on the other side. The greater the airflow, the more heat is removed.
- Negative pressure, achieved using large **fans on the container exterior** pull air out on the hot side, increasing the airflow which helps increases the heat removed.
- Lowering the temperature of the incoming air, by using **water curtains** which create evaporative cooling which increases the amount of heat removed.

See Figures 1 and 2.

With containerized data centers, the environment can be analyzed and controlled in a modular way as well. Conclusions from 1 container can directly apply to N containers. Thus most of the information here will be presented in a scale-invariant way. Where the quantities depend on scale, we will specify it per module. One module is a 20-ft container data center consuming about 0.5MW of power.

*info@qrb-labs.com

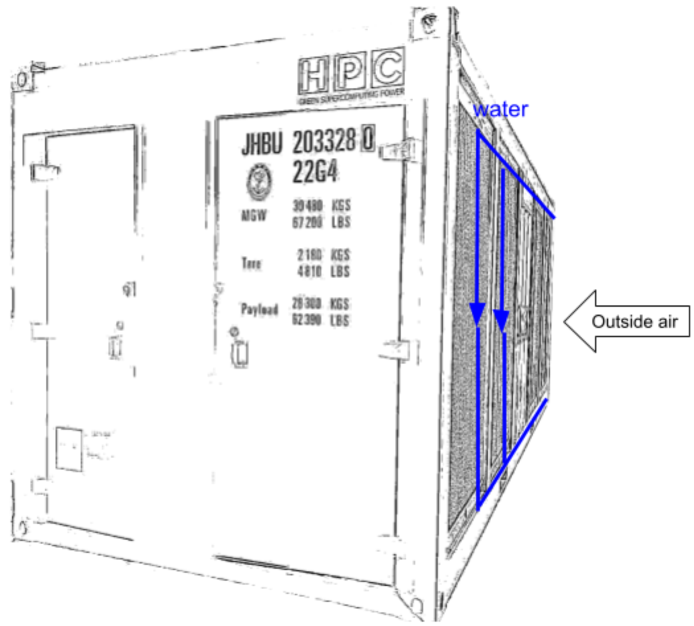


Figure 1: Data center container cold side exterior

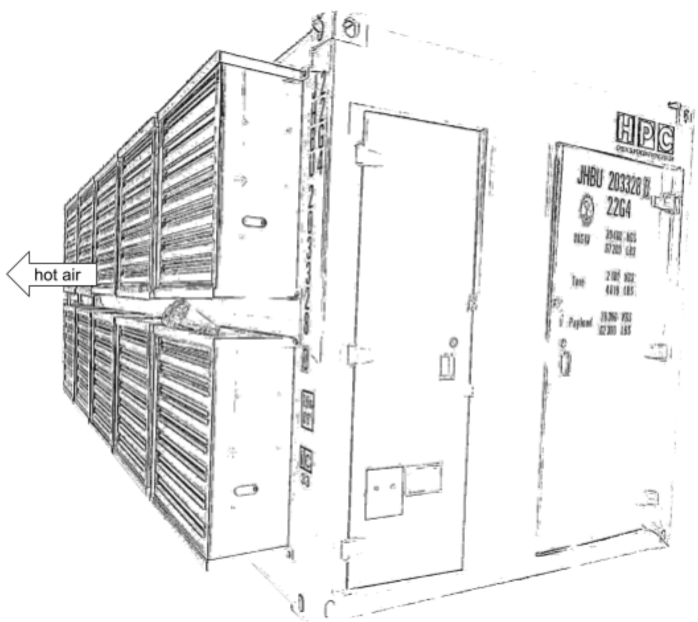


Figure 2: Data center container hot side exterior

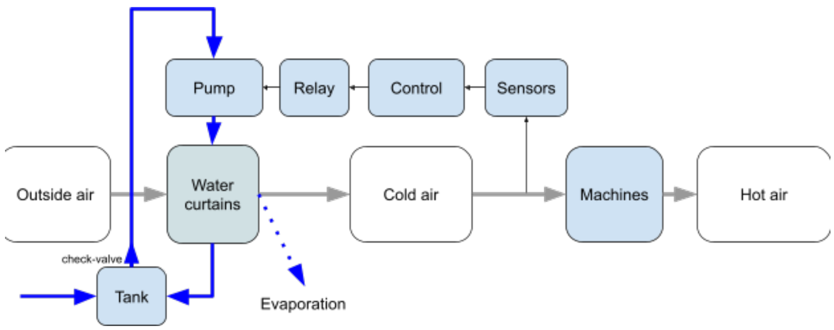


Figure 3: Feedback control of water

In this report, we focus on evaporative cooling. Specifically, our goal is to achieve the cooling objective with optimal water consumption.

2 Feedback control system

Ideally, net water consumption in the curtains should equal the evaporation mass needed to achieve the desired cooling. We get close to that in reality by running it in a closed loop, such that, modulo small leaks, any water that is not evaporated is recirculated via a water tank. This tank must be placed below the level of the floor of the containers. If not, water from the tank will flow backwards into the bottom of the curtains and leak out.

If the outside air is cool enough, e.g. below 25°C, the water curtains are not necessary. Third, too much humidity can harm the machines in the data center, so we don't want the water to simply be always on. To run the water circulation optimally, we use a feedback control system based on air temperature. See Figure 3.

3 Control circuit

The design and experimental hardware of the physical control system are shown in Figures 4 and 5.

- The actuator in this system is a **water pump**: when it is turned on, with power from a "main" source V3, the water circulates.
- Power to the pump is controlled by a "contactor" type **electromechanical switch**. When an electrical current flows in a coil between the control poles, A1 and A2, it creates a magnetic field which mechanically pulls closed a pair of contacts that allow the main power to flow to the pump.
- The current between A1 and A2, which is generated by voltage source V1, is controlled by a **digital relay**, which turns it on and off following a DC

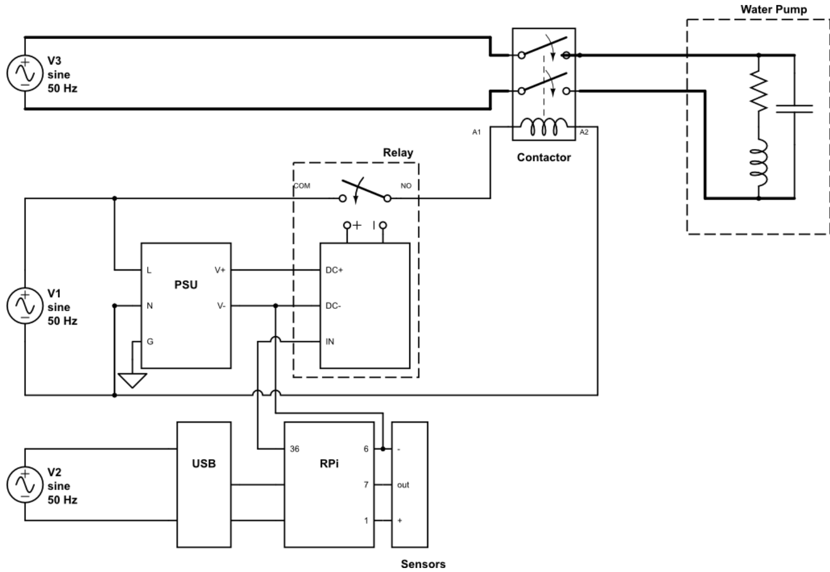


Figure 4: Control circuit

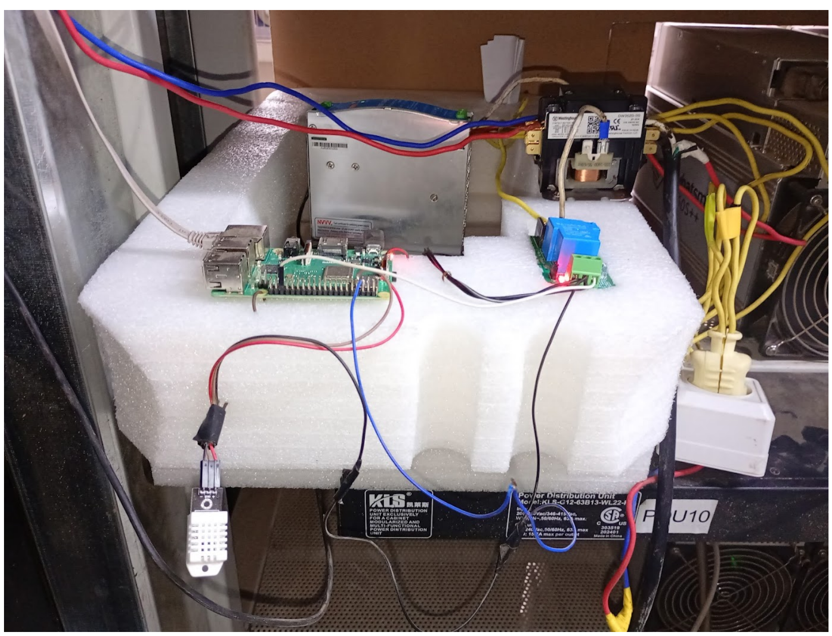


Figure 5: Experiment hardware. List of components in Table 1

signal.

- The DC signal is output from a Raspberry Pi (RPi) single board computer, by a **control algorithm**, using input from attached air temperature and humidity **sensors**.

A naive design would assume the digital relay can directly switch the pump motor: the motor would be driven directly off V1, while V3 and the contactor would be unnecessary. However, the pump motor is an inductive load. With an inductance L, voltage is proportional to the derivative of the current, $V = LdI/dt$. Thus, when it is switched on or off, there is a voltage spike. With a powerful pump, this spike can be large enough to damage the electronic components e.g. causing the relay to weld closed. Therefore it is advisable to separate the main power source V3 for the motor, and have it controlled by a heavy duty electromechanical switch which can absorb the spikes, as above. V1 and V2 are for low-power electronics and can be combined into a single power source if that is more convenient.

4 Physical system estimation

We start with simple experiments, in order to discover the system parameters. Specifically, we use fixed pulse control above a threshold temperature of 25 °C:

if $T > 25$ then turn ON for $a = 1$ min; wait $\tau = 30$ min; repeat

We ran two additional variants with $a = 1, \tau = 15$, and with $a = 2, \tau = 15$. Each variant of this experiment was run for 7 days.

Observation 1 *Temperature drop is proportional to water flow duration. See Figures 6 and 7. Each additional minute of water circulation produces a drop of* ≈ 2 *°C. Note that this coefficient is scale invariant, as the air, water and heat quantities are all proportional to the size of the data center.*

More precisely, we observe that the relationship between temperature change and activation duration is affine, i.e.

$$a = \alpha \Delta T + a_0 \tag{1}$$

A physical explanation for the time lag a_0 is as follows: since the tank is below the level of the floor of the containers, and inflow to the curtains is at the top, the pipes from the pump to the curtains must have vertically rising sections. Thus when the pump is turned off, the water in that section drains back down into the tank with gravity. When the pump is turned on, it takes a_0 duration until the water fills the whole pipe and reaches the curtain again. Thus a_0 is the time to prime the pipes. This lag doesn't cause any net loss of water, but it does slightly increase the wear and tear on the motor so we want to minimize it. To do so, we install a check-valve, aka a one-way valve, at the output of the tank, so the water can't drain back down. Before the check-valve, we have

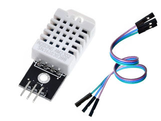
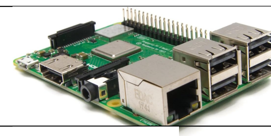
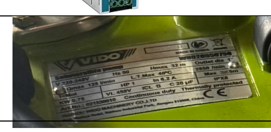
Component	**Description**	
Sensors	DHT22/AM2302 Digital Temperature and Humidity Sensor Module	
Computer	Raspberry Pi 3 Model B+ Board (3B+)	
Relay	FainWan Relay Module One Way 30A Optocoupler Isolation High Power Relay High/Low Level Trigger YYG-2 (12VDC)	
Contactor	Westinghouse HVAC Contactor, 2 Pole, 30 Amp, 240V Coil	
PSU	NDR-120-12 Power Supply NDR 120W, Input: 100-240VAC, Output: 12VDC, 10A	
Pump	Submerged Pump (125l/min, 0.75kW, 6.2A)	

Table 1: Hardware component specifications

Figure 6: Activation duration $a = 1$ min (top) and 2 min (bottom)

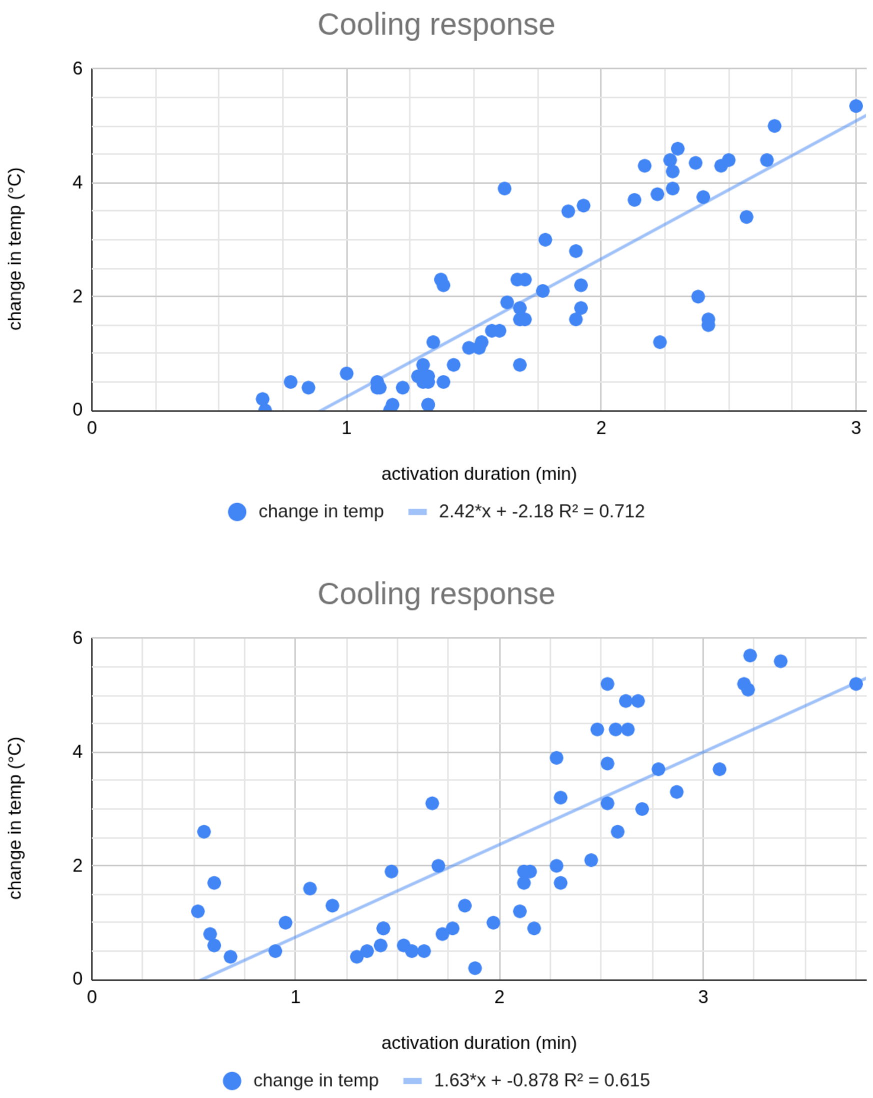


Figure 7: Activation duration a vs temperature change ΔT, before (above) and after check-valve (below)

$\alpha = 60/2.42 = 25$ s/°C and $a_0 = 60 \times 2.18/2.42 = 54$ s. After this improvement, we get

$$\alpha = 60/1.63 = 37 \text{ s/°C} \tag{2}$$

and

$$a_0 = 60 \times 0.878/1.63 = 32 \text{ s} \tag{3}$$

reducing the priming time to half. We can't get it to zero because, when the pump is off, the horizontal sections of pipe will empty into the water curtains and must also be primed. See Figure 7.

Now, accounting for the offset a_0, the observation agrees with theory. From the fundamental energy balance for adiabatic evaporative cooling [3], temperature change should be linear in the mass of water evaporated:

$$\Delta T = \frac{m_w L}{m_a C_p} \tag{4}$$

where m_w is the mass of water evaporated, m_a is the mass of air, $L = 2,440 kJ/kg$ is the latent heat of vaporization, and $C_p = 1.006 kJ/kg/°C$ is the specific heat of air.

Observation 2 *Net water consumption is about 20 liters per minute per container. Since it is in a closed loop, we can assume the amount of evaporation is equal to that consumption.*

Using our empirical value from Observation 2 in Equation (4), *for every 1 second of water circulation*, we have $m_w = 20/60$ kg, and the heat removed through evaporation is

$$Q = m_w L \approx 800 \text{ kJ}$$

That means the cooling power is 800kW. But the data center only gets 500kW total power per container, so we have much more cooling potential than heat produced. If we left the water circulation on full time, then, evaporation would stop as we reach maximum humidity. A constant high level of humidity of course is harmful for the equipment. Thus even if water supply was unlimited and/or zero cost, we would still want to optimize the amount we use.

Observation 3 *Width of the pulse response: when the water is turned on, the temperature drops until it reaches a bottom about 7-8 minutes later, and then it rises back to the original level after 15 minutes. This pulse response width does not depend on the duration a of the control pulse. (Figure 6)*

To explain this, from Equation (4), consider:

$$m_a = \frac{m_w L}{\Delta T C_p} = \alpha * m_w \frac{L}{C_p} \approx 30,000 \text{ kg}$$

That is the amount of air needed to give a quantity of heat Q to the evaporation. Of course we do not have 30,000 kg/s of air flow (that would be like a hurricane!).

Our typical containers have 10 fans each, with diameter 1m, and power 0.55kW or 1.1kW per fan. The air flow produced can be calculated using the pressure difference, mass of air and the power of the motor – see e.g. [1] – and in this case, translates to about 74 kg/s. This implies that it takes about

$$30,000/70 \approx 400 \text{ s}$$

for the outside air, after being cooled by evaporation, to mix with the existing air in the data center and reach the new temperature. This mixing time does not depend on the size of the pulse a, only on the speed at which the air moves through the container. So in theory, it should take about 7 minutes for the cooled air to mix in, and another 7 minutes to mix out. And that agrees closely with Observation 3!

5 Control algorithm

Goal:

1. keep temperature below a threshold $T^* = 25°C$,
2. while minimizing water consumption and humidity.

5.1 P control

To optimize the control, i.e. achieve the temperature target as well as possible with minimal water use, we can vary the amount of water as a function of temperature. Observation 1 suggests proportional control. Specifically, $T - T^*$ represents the error, and the desired temperature change is our control signal

$$u = (T - T^*)^+ \tag{5}$$

which translates to a water circulation duration a given by Equation (1), which we can rewrite as

$$a = \alpha u + a_0$$

This control action can be taken every τ. For proportional control, ideally, the control frequency should be faster than the impulse response, so that the signal can actively shape the response. Based on Observation 3, that implies $\tau < 7$ min. However, in our setup, the pump is submerged, and too frequent activation can reduce the longevity. So we limit ourselves to $\tau = 15$ min.

5.2 PD control

Knowing our control is a bit slow relative to the system it is trying to control, an improvement is to make the control proportional not just to the current temperature but the predicted temperature τ minutes in advance, i.e.

$$u = T + \frac{dT}{dt}\tau - T^* \tag{6}$$

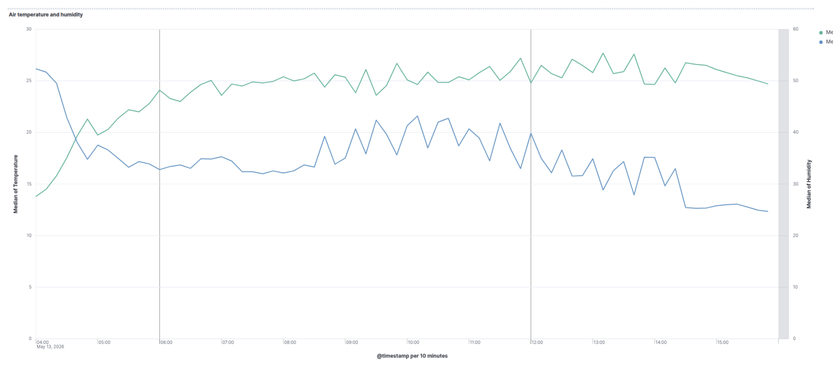


Figure 8: PD control result

which makes it proportional and derivative control. That succeeds in controlling temperature down to the threshold, as we see in Figure 8. But as we can see it bounces back up in between activations, and the average is therefore above the threshold.

5.3 PID control

To eliminate this bias at steady state average, we add an integral term,

$$u = K_p(T - T^*) + K_d \frac{dT}{dt} + K_i \int (T - T^*) dt \tag{7}$$

making it proportional integral and derivative control. See [2]. The constants can be set based on physical interpretation.

- The proportional term $K_p = 1$ implies that if the error is 1 degree, we want the correction to be 1 degree.
- The derivative coefficient we set $K_d = \tau = 15\ min$, meaning if we predict an error of 1 degree in 15 minutes we apply 1 degree of correction now.
- The integral term $K_i = 1/(60\ min)$ implies that a past error of 1 degree cumulated for one hour adds 1 degree of correction.

More details on PID controller parameter interpretations can be found in [5].

Implementation note: Temperature and humidity are sampled every 1 minute. For robust real-time estimate of the derivative of temperature $\frac{dT}{dt}$, we take the slope of the temperature trend least-squares regression, over a window of 1 hr. For the integral term, $\int(T - T^*)dt$, we take a left Riemann sum over the same window. See [4] for source code.

Figure 9 shows the results with this algorithm, where indeed we eliminate the long term bias thanks to the integral term. With the updated parameters after

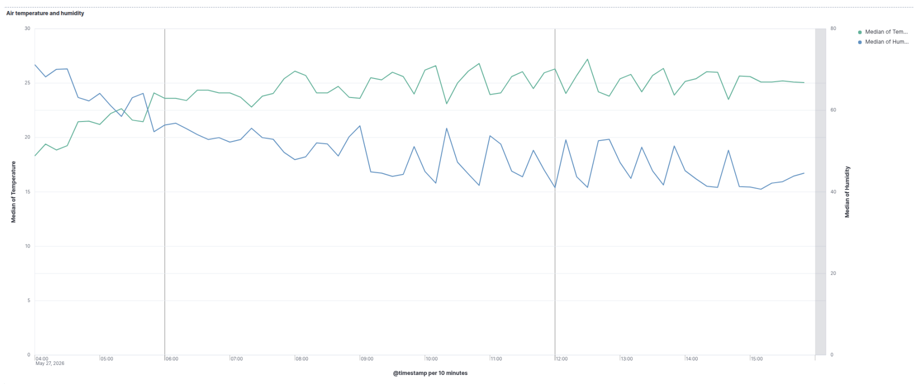


Figure 9: PID control result

Figure 10: PID control with check valve

adding the check-valve, PID control achieves temperature control with slightly less total pump activation time. Figure 10 shows the activation duration and effects over a particularly hot 2-hour period.

6 Water usage

Figure 11 shows a longer period, spanning the transition from a hot dry season to a cool rainy season. Water consumption varies depending on the external conditions, as the control system successfully maintains a steady internal environment. Our final algorithm (PID), which starts at the halfway point of Figure 11, shows efficiency improvements over the simpler algorithms. Less water is used for the same performance. However due to seasonal changes during our experiments, it is difficult to quantity the improvement with all else being equal.

Figure 12 shows water use efficiency per day. The plot illustrates the experimental

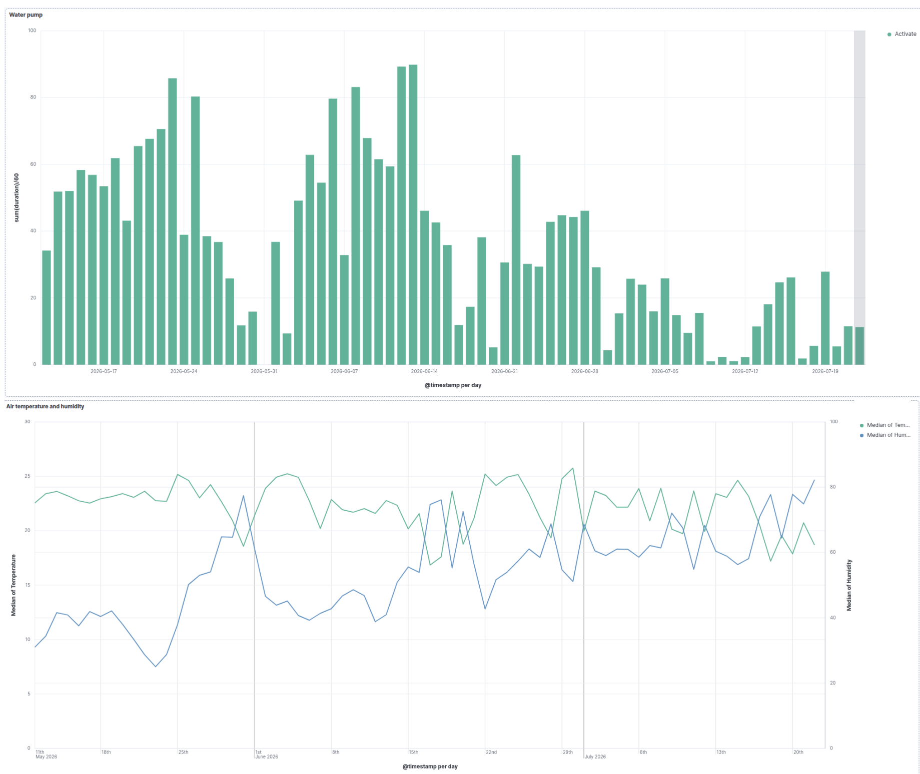


Figure 11: Daily water use (top) and median temperature and humidity (bottom)

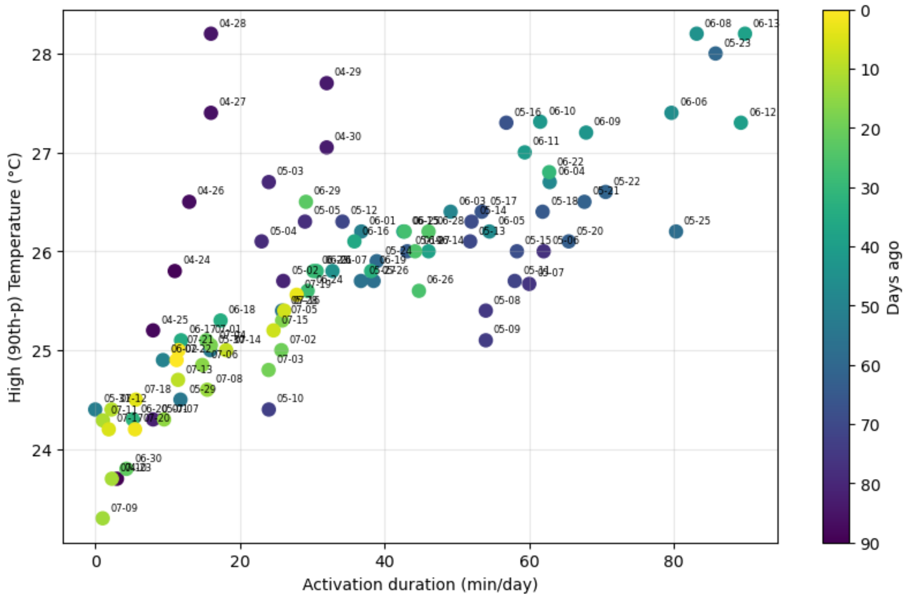


Figure 12: Daily water use efficiency

period (purple/blue dots) to the later PID control period (yellow/green dots).

Table 2 shows a summary of usage statistics. Around 60-80% of the circulating water is evaporated and hence actively contributing to cooling, while the rest is recirculated in the closed loop. Net water consumption is in the hundreds of liters per MW per day, same order of magnitude as the average American household.

Metric	**Value / Range**
Water circulation duration	40 ... 500 min/week
Net water consumption	2 ... 8 m^3/week/container
	300 ... 2,000 l/day/MW
Evaporation share	60% ... 80%
Recirculation share	40% ... 20%
Air flow rate	70 kg/s/container
Evaporative heat removal rate (when active)	800 kW/container

Table 2: Summary statistics

References

[1] Frank P. Bleier. *Fan Handbook: Selection, Application, and Design*. McGraw-Hill, New York, 1998. Relationship between motor kW, static pressure, total pressure efficiency, and CFM deliverable.

[2] Dean K. Frederick and A. Bruce Carlson. *Linear Systems in Communication and Control*. John Wiley & Sons, New York, NY, 1971.

[3] Frank P. Incropera, David P. DeWitt, Theodore L. Bergman, and Adrienne S. Lavine. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, Hoboken, NJ, 6th edition, 2007. Chapter 1: Conservation of Energy & Chapter 6: Simultaneous Heat and Mass Transfer.

[4] QRB-Labs. Feedback control logic (`feedback_control.py`). `https://github.com/QRB-Labs/qrb/tree/main/rpi/`, 2026.

[5] Zurich Instruments. Principles of PID controllers. White paper, Zurich Instruments AG, July 2023. Accessed via Zurich Instruments Application Resources.