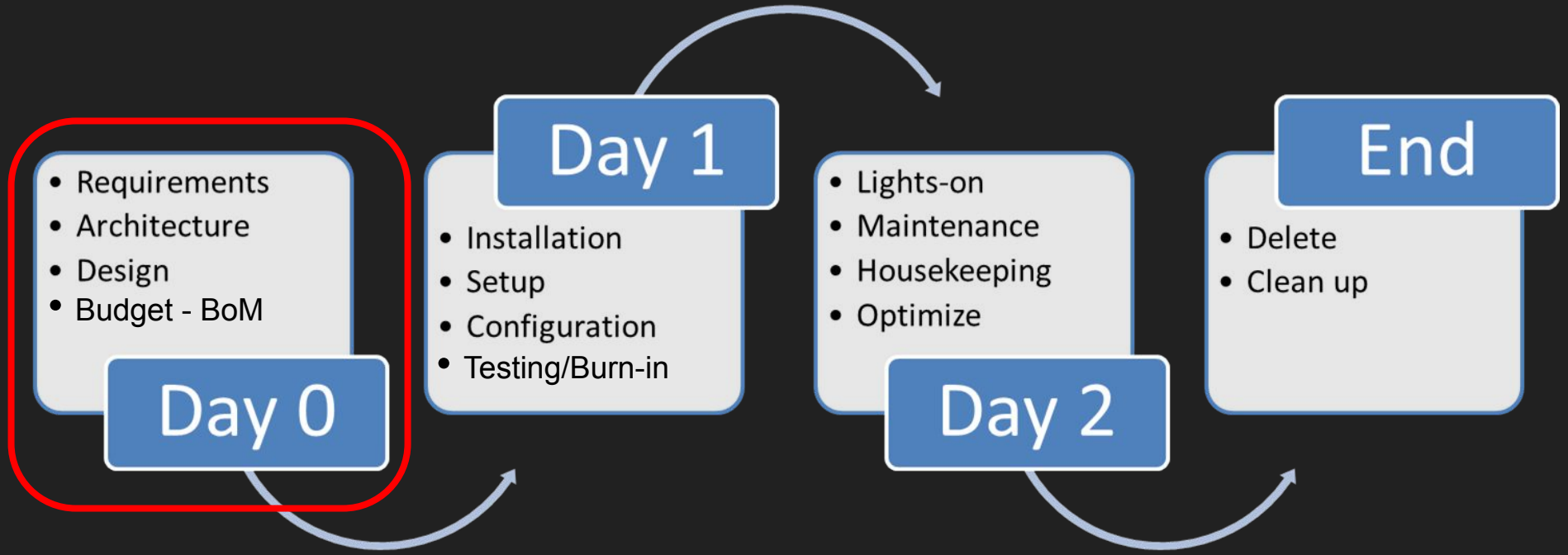
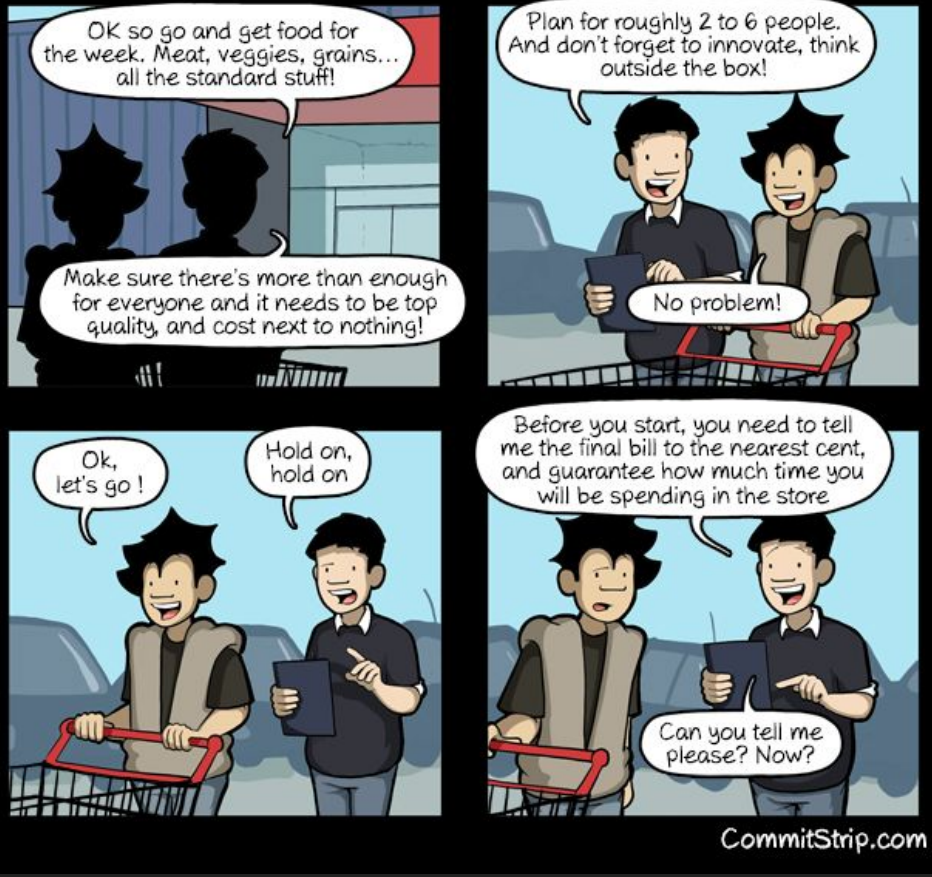
The Art of Capacity Planning

Trying to not run out of resources (at scale)

Infrastructure Operations Lifecycle



IT Project Estimates



What is capacity anyhow?

Capacity

How much of something is being produced, or is currently available

Demand

How much of something is being consumed

Forecasting

How must capacity be adjusted over time to meet demand without wasting resources

Examples?

What is capacity anyhow?

Design Capacity

The theoretical maximum efficiency of a system assuming no losses and exact performance of proposed system

Effective Capacity

The capacity at which a system is intended to run at long term including room for losses, performance overhead, and potential less than optimal design

Actual Capacity

The capacity at which a system actually runs at, with both planned and unplanned performance degradations or optimizations

Operating Efficiency

Efficiency = Actual output / Effective capacity
Utilization = Actual output / Design capacity

For optimal efficiency, capacity and demand need to be perfectly matched

Q: Why isn't this practical?

Too much capacity => underutilized/unused resources
Not enough capacity => unsatisfied customers

Q: So how do we resolve this?

Adjusting Capacity

Increasing Capacity

- Adding more hardware
- Increasing staffing to handle support
- Optimizing architecture to reduce demand
- Change demand to fit capacity

Decreasing Capacity

- Is it just the inverse of the above?
- Is it easier or harder than increasing capacity?

Adjusting Capacity

Bottleneck

> A resource that is already working at its full capacity such that it cannot handle any additional demand, and additional demand is required.

Examples of bottlenecks?

- Hardware (CPU, RAM, Storage, Network Capacity)
- Environmental (Space, Power, Cooling)
- Staffing (Experience, Training)
- Access to systems (Secure Environments, On-boarding)
- Budget ($$$)

Let’s talk strategy

Q: Should we:

- **Overprivision?**
- **Underprovision?**
- **Just meet demand?**

Q: Does the type of system matter?

Capacity Planning is closely tied to business goals and risk tolerance

Capacity Planning Methods

Capacity Leads Demand

- **Plan to have excess capacity such that it is always available when demand rises**

Capacity Matches Demand

- **Monitor demand closely and add capacity as forecasted demand predicts**

Capacity Follows Demand

- **Wait until demand exceeds capacity before adding further capacity**

Capacity Planning Timelines

Short Term (0 - 3mo)

Reactive Planning

Spikes in demand

Usually small changes

i.e. Spin up more instances to deal with flash sale event, increase size of disk on VM

Capacity Planning Timelines

Medium Term (3mo - 1yr)

- **Forecasted Planning**
  - **Adjusting to seasonal changes, month over month growth, etc**
- **Often larger changes**
  - **potential design changes**
- **Staff Changes**
  - **adding or reducing based on upcoming projects**

i.e. Adding more hosts to deal with the rise in holiday eCommerce, changing DB clusters, credential rotation and patching

Capacity Planning Timelines

Long Term (1yr+)

Significant changes
(often architectural)

Often difficult to implement and more difficult to revert
(technical debt)

i.e. New platforms, Moving to the Cloud, New DataCenter, etc

So how does this work in practice?

It usually starts with one question:

- **“So, what is it you are trying to accomplish?”**

Then is usually followed by:

- **“Where are you going in the next year, what are your growth targets?”**
- **“What application / workloads?”**
- **“What happens if you end up on the front page of reddit?”**
  - **Is that possible?**
  - **Can you scale? / How bad would it be if you couldn’t?**

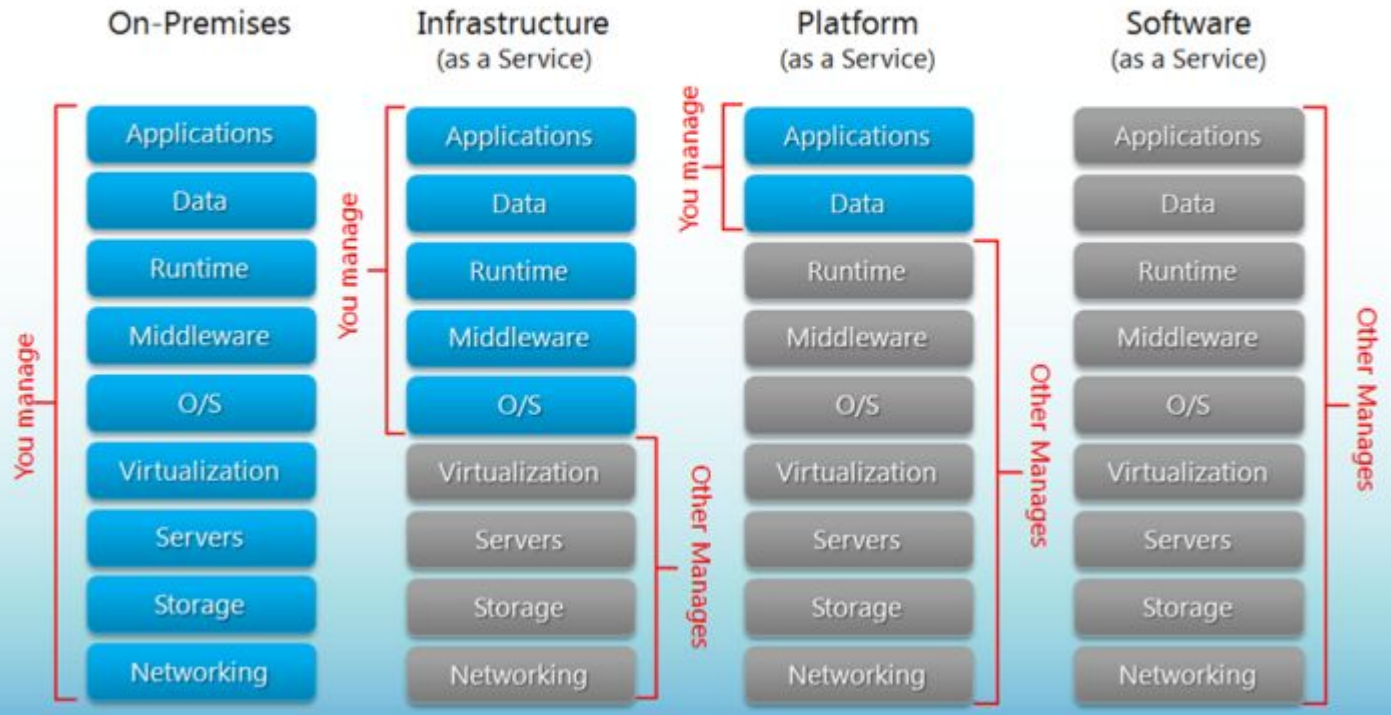
Time to run through an example

You are an engineer working for ABC company which is

- **A fairly large retail business (~1000 employees)**
- **Business is conducted both on premise and via eCommerce**
- **looking to add a new retail website to take over for their aging legacy systems**
- **Everything needs to be net-new as existing infrastructure is in need of replacement**
- **Presume networking requirements will be handled for you (this is coming up next week)**
- **Average concurrent clients are ~10,000**

Q: Do we need any more information?

Separation of Responsibilities



So where to begin?

Top down approach

- **Start with the application**
- **Move down the stack to data-systems => OS => VMs => Servers => Storage**

Bottom up approach

- **Start with what you have => then figure out what additional resources you need as you go up the stack**

Q: Which approach should be used in this case?

Example Application Requirements

Each application instance

- **Can serve 1000 clients**
- **Requires 2 CPU cores at 1.0GHz**
- **Requires 1.5GB ram**
- **Requires 25GB disk space**
- **Backed by database that grows by 5mb per client per day**
- **Runs on Windows or Linux**

VM Requirements

OS requirements

- What OS are you using?
  - Is there additional overhead?

Virtualization requirements

- What hypervisor are you using?
  - Is there additional overhead?
- Does this overhead always impact your Effective Capacity?

Hardware Requirements

Finally, down to ‘Bare Metal’

- **So, what do we need for bare metal hardware?**
- **CPU/RAM/Storage/etc**

Q: What are we not thinking about?

Lets run through an example

Anatomy of a Server Rack

How do we know how many servers will fit in a rack?

- **U/RU - Rack Unit (1.75in)**

What needs to go in a rack?

- **Servers**
- **Networking Routers/Switches**
- **PDU/UPS**
- **Console, Accessories (optional)**

What about cabling? Cooling Systems?

How much power do we need?

Depends on the Server in question

- **Can range from a few hundred watts to over a thousand per server.**

Servers will usually come with redundant hot-swap PSUs

- **Check the ratings to determine how much power the server will draw**

Calculating total power consumption (Simplified)

- **Convert Power Readings/Specs to Watts**
- **PSU Peak Load (W)**
  x Number of Servers
  x Number of Redundant Circuits (Usually 2)

Power at Scale

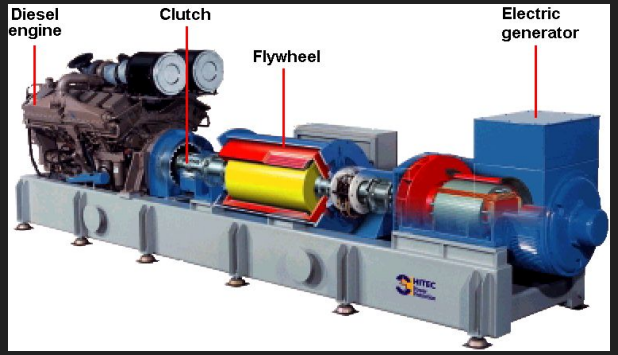
Power Distribution Unit (PDU)

- **distributes reliable network power to multiple devices**

Types of PDUs

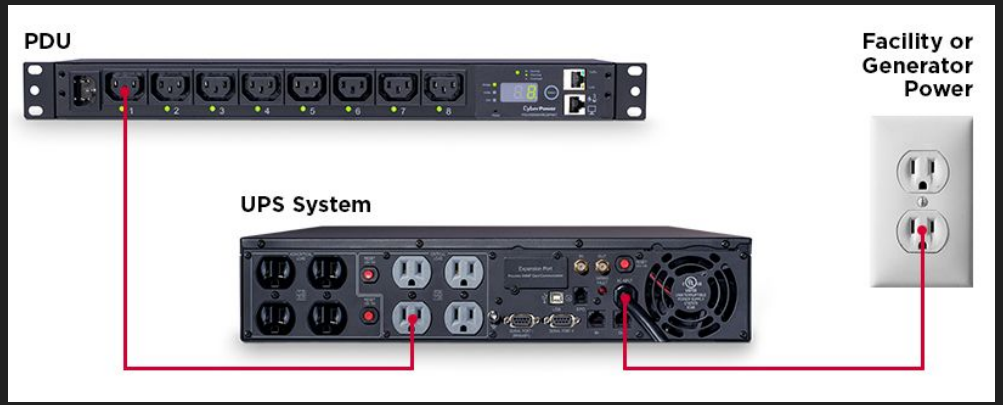
- **Basic (aka “Dumb”)**
- **Metered**
- **Monitored**
- **Switched**
- **Bypass Switch**
- **Auto Transfer Switch (ATS)**

Power at Scale



Uninterruptible Power Supply (UPS)

- Provides emergency power to a load when the input power source fails
- This could be full outage, brown-out, or just ‘bad power’
- Range in size from single computer to entire datacenter in size



So what about all that heat?

Servers take a ton of power => which in turn creates a ton of heat

Heat Dissipation Methods

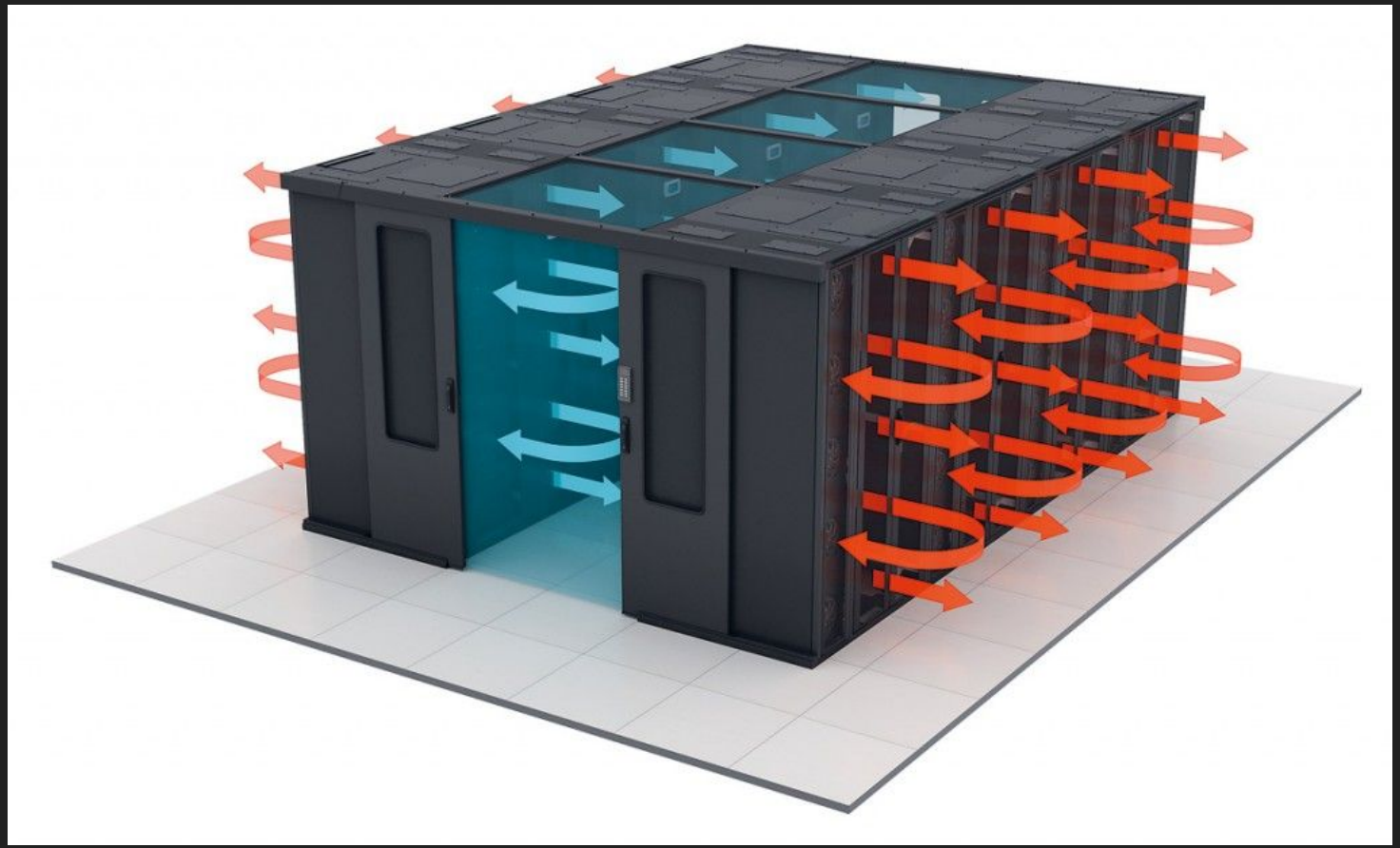
- **Air Cooling**
- **Liquid Cooling (inc. Chilled Water)**
- **Free Cooling?**

How much cooling capacity do we need?

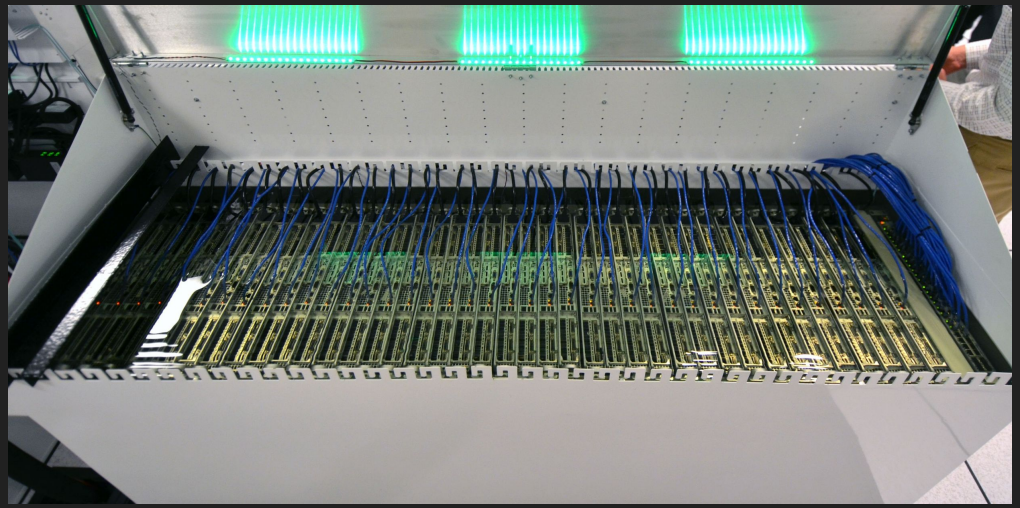
- **Heat is measured in BTU (1 Watt X 3.412)**
- **1 Refrigeration Ton = 12,000 BTU**
- **Usually cooling solutions are measured in BTU/h that can be dissipated**

CRAC - Computer Room A/C

Hot/Cold Aisles



Water Cooling



Water Chillers

Cooling Towers

Free Cooling?

Free Cooling?

Questions?

We just covered a lot of material - What questions do you have?

Key Ideas

- What is Capacity? (Design, Effective, Actual)
- How do you calculate Efficiency/Utilization?
- What are the difficulties in Adjusting Capacity? (Bottlenecks, Technical Debt, etc)
- Capacity Planning Methods (Leads, Matches, Follows) - tradeoffs
- Given App requirements (and relevant stats regarding lower layer requirements/overhead) - calculate required hardware capacity
- Given server/equipment stats - calculate power and cooling requirements
- UPS vs PDU (why are they important)
- Different Cooling Methods

References

- Operations management. ed. / S. Paton; B. Clegg; A. Pilkington; J. Hsuan. Basingstoke : McGraw-Hill, 2011. p. 207-237.
- https://www.vxchnge.com/blog/data-center-cooling-technology
- https://en.wikipedia.org/wiki/British_thermal_unit
- https://www.amazon.com/Guerrilla-Capacity-Planning-Tactical-Applications-ebook-dp-B004CRTNAA/dp/B004CRTNAA/ref%3Dmt_kindle