Cloud Technology Strategies

A survey of needs and opportunities in the Cloud industry

Steve Webster • Google Datacenter Software • 2021-05-05

What's a "strategy"?

The kernel of a strategy contains three elements:

1. A *diagnosis* that defines or explains the nature of the challenge. A good diagnosis simplifies the often overwhelming complexity of reality by identifying certain aspects of the situation as critical.
2. A *guiding policy* for dealing with the challenge. This is an overall approach chosen to cope with or overcome the obstacles identified in the diagnosis.
3. A set of *coherent actions* that are designed to carry out the guiding policy. These are steps that are coordinated with one another to work together in accomplishing the guiding policy.

Strategy example: Lord Nelson v. the French fleet

Situation: Lord Nelson, leading the British fleet, encounters the enemy. The French fleet is larger and more modern, but their captains and crews are less experienced than the British.

Guiding principles: Leverage British superior skill in seamanship, gunnery, and leadership. Avoid broadsides with the enemy. Disrupt their lines to impede signaling from the flagship.

Actions: “Straight at them” - arrange the British ships in straight columns to present a smaller incoming target, and rapidly engage French head on, creating a melee. Every British captain is free to fight as they see fit.

Trends

Renting computers >>> Owning computers

Situation: Compute and networking needs are growing fast for every organization. Computing hardware depreciates in 3-5 years -- often not a good investment.

Guiding principles: Businesses avoid capital expenses (CapEx) of low-return IT investments. Convert into high-return, predictable, scalable operational expenses (OpEx) instead.

Actions: Users rent computing capacity from cloud providers, who bear the CapEx costs but amortize them over many customers. Industry scale win-win for OpEx and efficient CapEx.

End of Moore's Law

Situation: the physics of photolithography are reaching their detail limit, slowing Moore's Law pace from historical ~18 month chip density doubling to >8 years. General purpose compute is too expensive / low performance to meet the planet's growing compute needs.

Guiding principles: Use hardware acceleration for high-value workloads. Create compute architectures that maximize machine and datacenter reconfigurability and utilization. Build engineering teams and manufacturing partnerships to rapidly bring custom silicon to market.

Actions: GPUs, AWS Nitro (VM offloads), Google TPU (ML training and inference) and Argos (video accelerator), etc. Co-design hardware / software solutions.
Research radical replacements for Von Neumann architectures and silicon chips: molecular computing, quantum computing, etc.

Data bottlenecks

Situation: massive growth in data set sizes driven by ML training, data mining, high-bandwidth content creation and sharing, etc. Compute utilization often limited by computer bus and network bandwidth, increasing costs.

Guiding principles: create machine and network architectures that relieve data bottlenecks. Position data nearer to users.

Actions: increase Cloud “edge” capacity, use in-package high bandwidth memory, create architectures that blur distinctions between computer bus and network for more effective disaggregation.

Reliability

Cloud users' expectations of reliability

Situation: reliability of cloud infrastructure is a competitive advantage because more reliable infra means customers can buy less of it to achieve same overall uptime (and simplify their deployments)

Guiding principles: balance reliability and infrastructure cost to meet customer needs while optimizing profitability and provider reputation. Cloud vendors use their own infrastructure where possible.

Actions: monitor cloud reliability at least as closely as customers' own workloads. Create systems with predictable failure modes, balanced with overall performance.

Growing system complexity

Situation: cloud systems are being driven towards greater complexity and layering due to the end of Moore's Law, diverse workloads, efficiency demands, etc. This creates unintentional dependency loops, failure cascades, and unpredictable failure modes.

Guiding principles: Engineer away, otherwise accept and hide complexity. Isolate systems behind simple, long-lasting interfaces wherever possible.

Actions: Data centers are enormously complex warehouse-scale computers, but virtualization and layered systems make them appear to be simple and consistent. Design products for apparent simplicity: e.g. Spanner is a highly complex multi-master global datastore with guaranteed latency, but it looks like a sharded SQL database to users.

Plan for failure

Situation: at large scale, production systems will, not may, fail unpredictably.

Guiding principles: do not deploy production systems whose failure blast radii are not well understood. Where blast radius is necessarily global (authentication, quota, etc), harden these systems against failure.

Actions: rigorously understand and document system failure modes and dependencies. Design systems that defend their boundaries from bad data. Expend extra effort to make client systems resilient to upstream failure. Establish enforceable and measurable service contracts and SLOs.

Security

Situation: cloud providers run hostile / adversarial workloads on shared computers, increasing risk to themselves and customers.

Guiding principles: Isolate all workloads without unduly impacting performance. Find zero-day vulnerabilities first.

Actions: Have zero trust at system boundaries. Run all workloads in sandboxes. Track “dirty” media (including firmware) and establish root-of-trust procedures for making it “clean.”

Velocity

Deploy in fast, small increments

Situation: Predictions of cloud demand have large error bars because customers don't know themselves what they'll want to buy in the future.

Guiding principles: building unused cloud capacity ahead of demand is wasteful, but failing to meet cloud customer demand is worse. Satisfy demand while the error bars are still close together.

Actions: Create data centers, deployment techniques, and products that can be quickly deployed in small capacity increments. Design for drop-in product interchangeability & rapid development.

Flexible, optimal planning

Situation: Future cloud demand is difficult to predict. Products, locations, capacities, and go-to-market strategies are all affected.

Guiding principles: Make plans as if those plans will necessarily change in flight. Be prepared to handle arbitrary dimensions on the optimization space as complexity increases.

Actions: Create planning systems that can continually and optimally plan and re-plan all aspects of cloud deployment as circumstances, demand, and predictions change. Integrate planning with the product development pipeline.

Predict the future by causing it

Situation: Short-term prediction of cloud workload mix and demand is inaccurate, leading to inefficiency. But long term trends (e.g. IaaS → PaaS) can be planned for.

Guiding principles: create new products and technologies that create or anticipate demand. Design them to start small but scale fast.

Actions: Google TensorFlow software + TPU hardware, AWS Graviton2 CPUs, various "platform" products like BigQuery, Spanner, ElasticSearch, etc.

Efficiency

Efficiencies at industrial scale

Situation: enormous, expensive industrial infrastructure (power, cooling, buildings) is required for cloud computing.

Guiding principles: minimize the infrastructure needs by maximizing utilization -- make the infrastructure as “busy” as possible. Use large-scale control systems to realize operational efficiencies.

Actions: Oversubscribe all dimensions of datacenter operation where utilization is low and predictable: power, cooling, etc. Take advantage of nonlinearities in power or water usage (electric motors, CPUs, fancoils, etc.) to realize efficiency gains by spreading load evenly across resources.

Take advantage of flexible guarantees

Situation: some cloud users are willing to trade guaranteed performance for lower cost

Guiding principles: identify opportunities to trade long-tail guarantees of performance for higher overall utilization and/or burst performance.

Actions: most cloud providers have some form of a “preemptible VM” that has weak guarantees of availability but a low cost. These kinds of workloads are “sheddable” in the event of a load spike, enabling a load-shaping mechanism that enables higher utilization overall.

Storage durability, performance, locality

Situation: Cloud data storage needs are growing exponentially. Storage performance and durability are intertwined with datacenter infrastructure and networking.

Guiding principles: co-design storage and data center infrastructure products to optimize performance at scale. Predict the arrival of new storage technologies and plan accordingly.

Actions: Much of the work in this area is proprietary, because storage has a simple interface that allows radical complexity and optimization behind it.

Sustainability

Data centers use lots of power and water

Situation: data centers are industrial sites that consume 3MW to >100MW of power continually, and up to 5400 liters of water per MWh for cooling. Site selection is largely driven by availability of water and power.

Guiding principles: Minimize PUE (power utilization effectiveness) overhead. While reliability is of paramount importance, power and water efficiency are close behind.

Actions: Engineer high-efficiency air and water based cooling systems. Take advantage of high thermal gradients and equipment that can run at high temperatures. Use waterless cooling, immersion cooling, and liquid cooling where appropriate.

Carbon reduction is a priority

Situation: most energy available on the grid is carbon-emitting (fossil fuels), but carbon-free energy availability is growing rapidly. Availability typically varies (sun shines, wind blow) throughout the day and seasons.

Guiding principles: Consume carbon-free energy whenever possible.

Actions: Design global schedulers to move workloads in space and time to consume low-carbon energy. Shed loads to avoid consuming carbon energy where practical.

Google has committed to 100% carbon-free energy by 2030, other cloud providers are also moving to renewable energy sources.

Renewable grids are less stable grids

Situation: renewable energy generation is growing rapidly, displacing legacy fossil fuel and nuclear generation. But renewables are less reliable (no wind, cloudy day), and data centers need reliable energy.

Guiding principles: use renewable energy where and when possible. Design solutions to help stabilize the grid as an active energy consumer.

Actions: integrate datacenter load control and energy storage systems with the local grid to provide load shaping, load deferral, and grid services to stabilize the grid and encourage renewable deployment.

Energy storage is starting to look attractive

Situation: Regulated emissions constraints and unstable renewable grids make some locations unattractive for data center placement.

Guiding principles: Energy storage attached to the data center or local grid can be used for carbon-free backup power, grid stabilization, diesel generator replacement, etc.

Actions: MSFT announced fuel cells, Apple has local solar farms, Google grid-scale batteries, etc.

Thanks for your time! Any questions?