Cloud App Anatomy

Tanj Bennett

Applications and Services Group

Microsoft Corps

Cloud Apps Are Personal

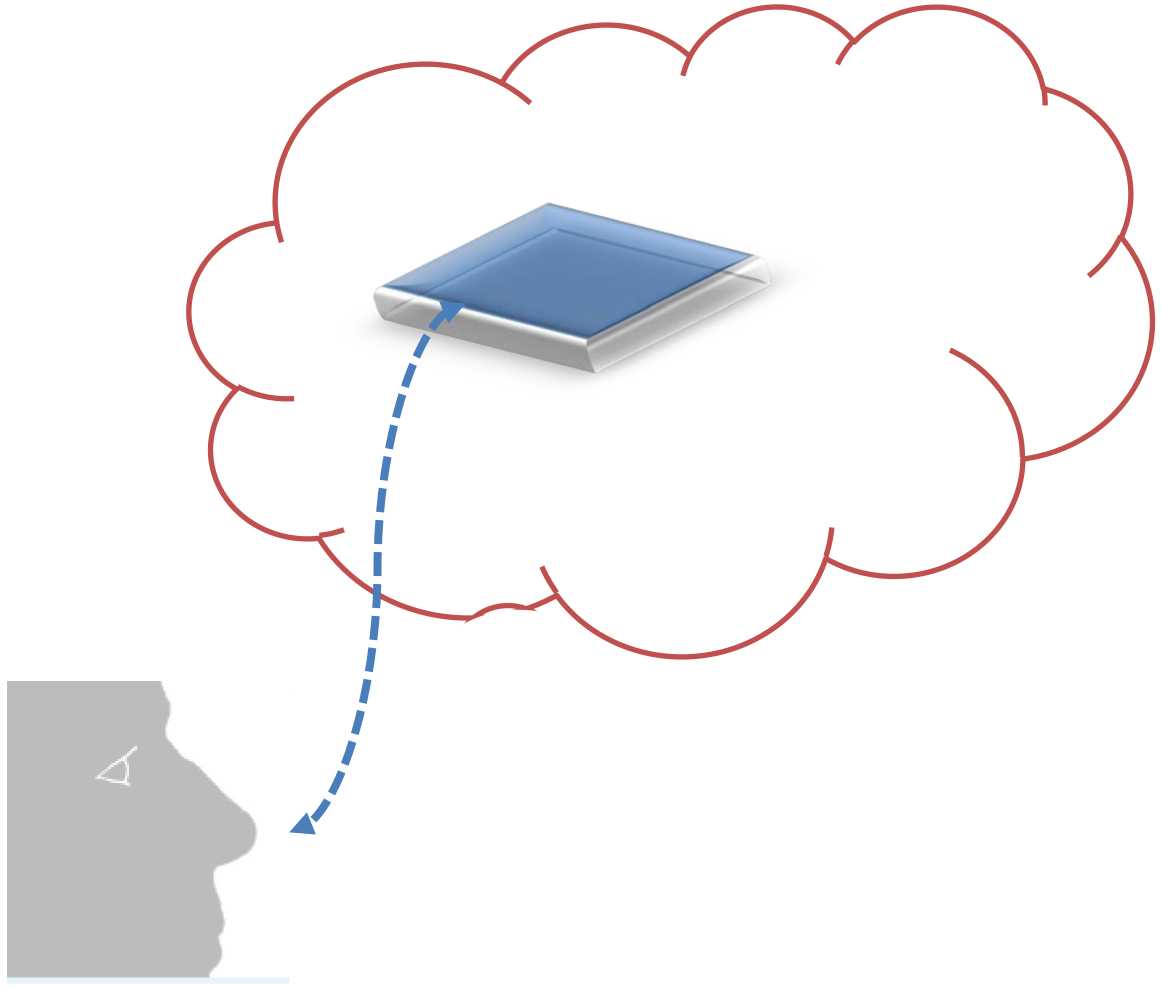
Personal applications have a display, means of input, and computational devices which execute them. My audience probably vaguely remembers a time when these were assumed to be in one machine.

Over time the display and input unified into mobile devices. As these mobile devices run on batteries, and because much of what we now like to do accesses the web, the computation migrated to data centers (the cloud).

Twitches required

Personal apps in the cloud remain interactive. Indeed, they ideally are every bit as responsive as they were when on a single machine. Humans have not changed. The same ideals of less than 50 ms for effortless discrete responses, and less than 20 ms for physical engagement (games, simulations, immersion) remain unforgiving. Today, we still expect most of the immersive UI to be executed at the device, so for most purposes cloud apps aim for the sub-50 ms target.

Cloud apps always answer to users, not job schedulers.

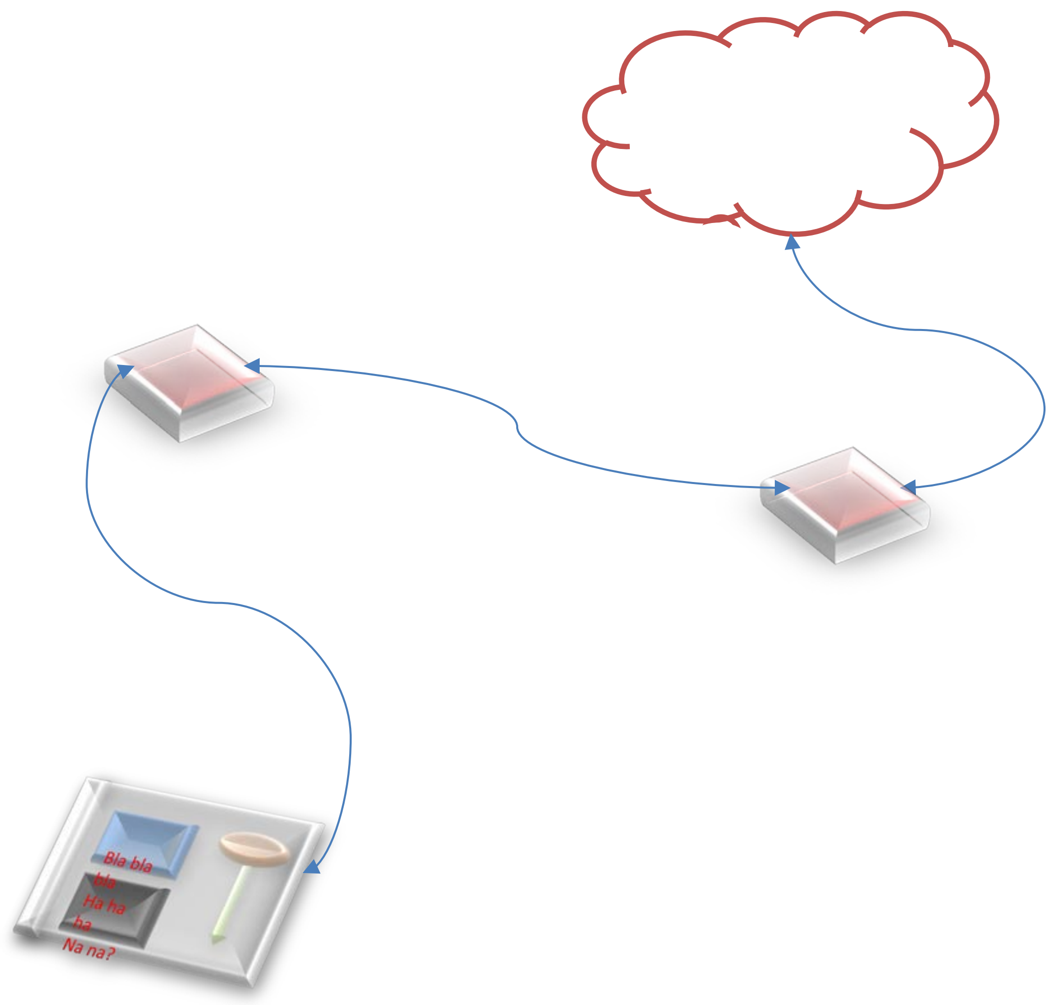


The need for speed

Latency has two components: delay due to distance, and delay due to stretching out the payload. The round trip delay by distance is roughly 1ms per 100km of separation between user and data center, over glass fiber. It seems likely that almost all humans could be within 10 msec of at least one data center within a decade.

Data stretch is the real driver for “wide band”. If we send 1 MB of response at 10 Mbps, that takes about a second. This is why 1 Gbps is so desirable – it gets us down to 10 msec for the stretch. This is much more important than the speed for media, which is likely to be under 50 Mbps even for 4k streaming.

This is fortunate, since ISPs can more easily oversubscribe burst traffic and their economic model requires oversubscription.



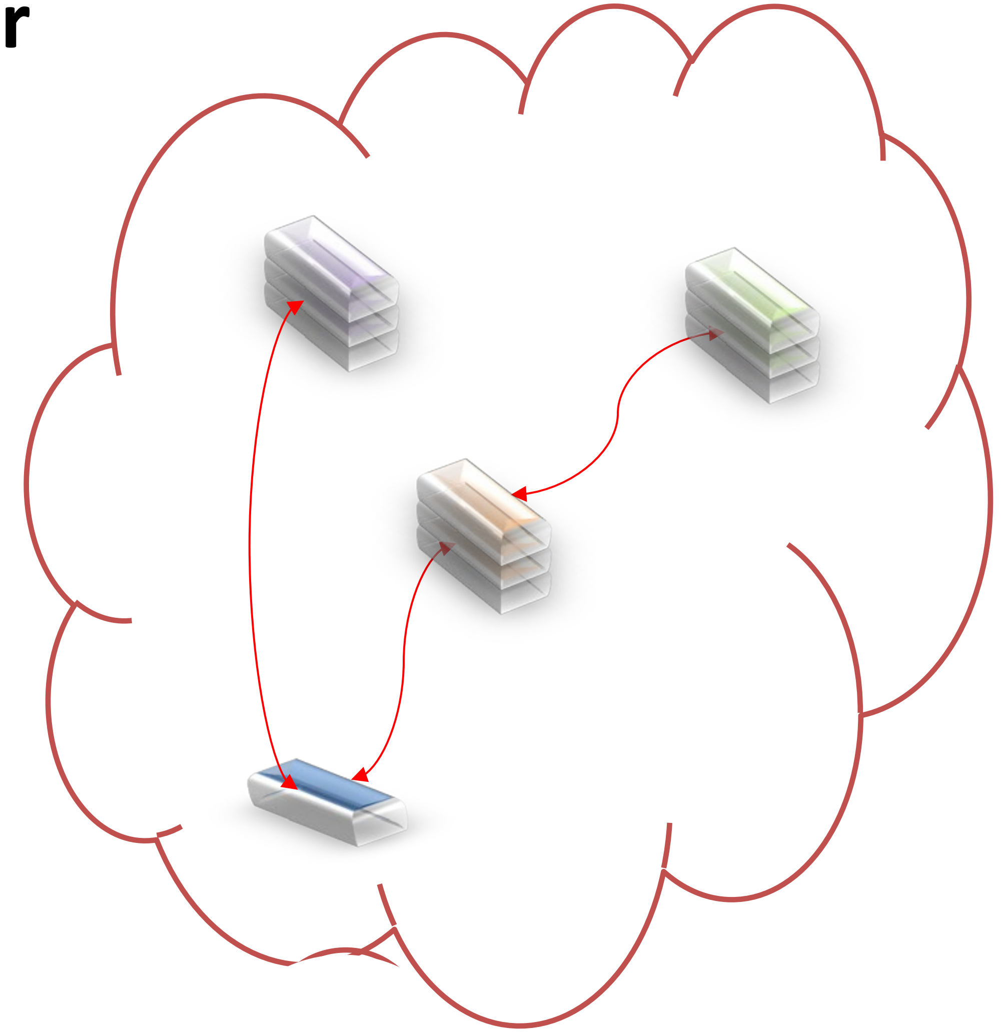
Work expands to fill a Data Center

Meanwhile, over in the data center a curious thing happened. Apps got a lot more complicated.

The pioneers were Search. All the vendors not only had to enlist large sets of machines for search but they increasingly had to run parallel operations like news, maps, advertising, and special topic updates. They developed systems for spelling correction, entity recognition, translation, local data, shopping, social graph.

These systems have been opened up as services which any application in the data center can call. Just as the search page is in reality an aggregation app mostly concerned with composition of the final page using information obtained from a family of services, other applications can compose these services as part of what they offer.

The data center allows far richer applications to be built than were possible within limited personal computers.



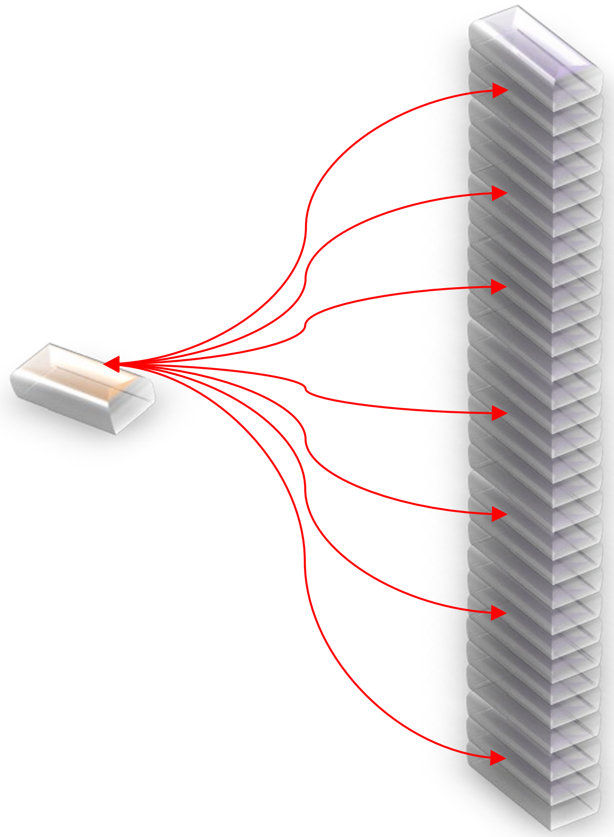
A 99.9 Percentile Problem

Today, a cloud app routinely calls upon thousands of different servers to compose an answer. Data centers are ridiculously parallel, and the resources they have are humongous. Datasets of tens of petabytes are ordinary.

Unfortunately, execution times are variable, but we still have that 50ms deadline to beat, maybe 40ms of which can be used inside the DC. If we use a thousand servers then we will end up waiting for the 99.9th percentile responses. All over the place, since pretty much any cloud-scale resource has at least hundreds of servers. Many services are layered 2 or 3 deep.

As a result uniformity of transport and execution has become the focus. Networks must deal with incast storms and cannot use strategies which assign random delays. Hypervisors must watch machine loading and avoid slowdowns and queues. Resources are aggressively scattered to avoid hotspots.

Latency outranks throughput.

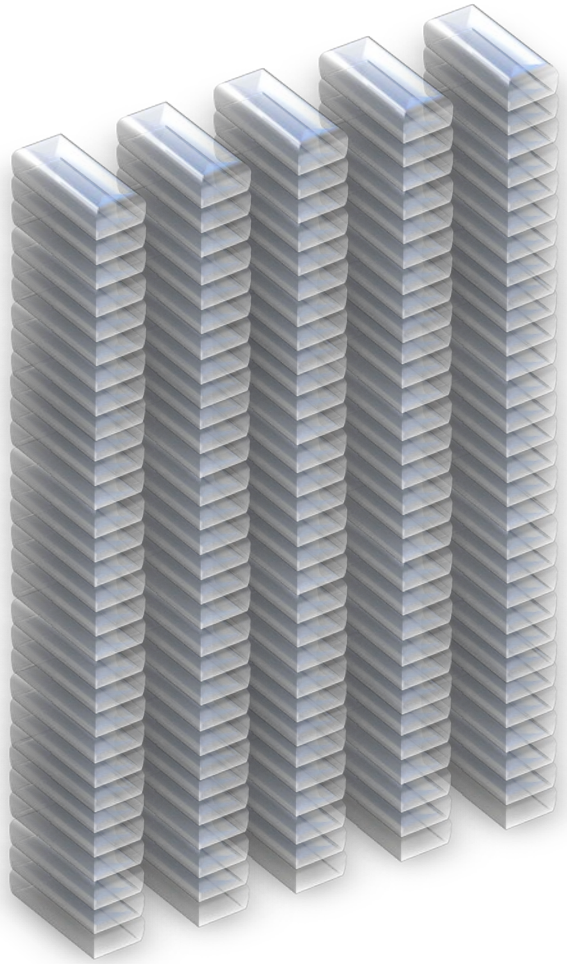


Listen to the users

Millions of them. They are impulsive. You don't hear from them for hours, then they want a response in milliseconds. Any given session is long boredom punctuated by moments of panic. So, the software is not organized in sessions requiring conventional thread contexts.

The aggregator software typically fills about 10,000 servers at each Microsoft data center today and growing rapidly. More important is the kind of software. There is a combination of memory/SSD cached database for all the context objects, and a system of context free asynchronous code which represents the logic of the application.

The asynchronous approach is necessary because of the high degree of parallelism. If you try to write synchronous code to manage scatter-gather from a thousand servers, it will be a mess. The context free nature of the code allows for impulsive start and response. It is a dataflow system where code proceeds when the data is ready.



Availability Rules

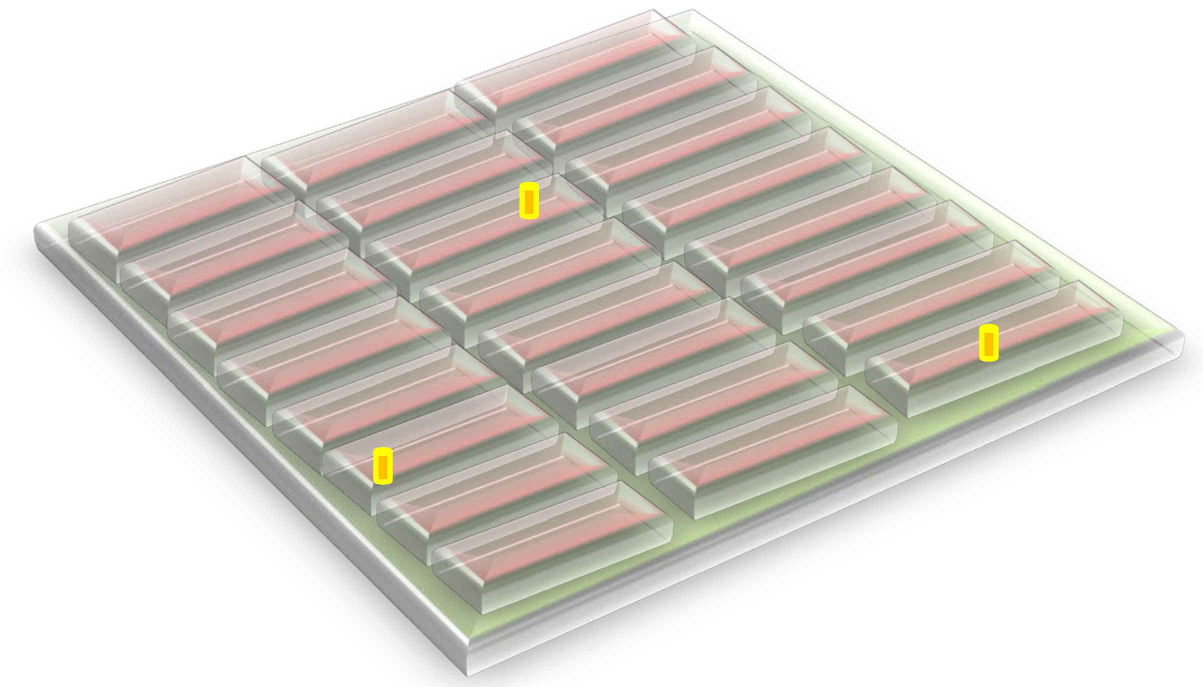
When the cloud service is unavailable mobile clients stop working. The shift to using the resources of the cloud to deliver novel functionality means there is no mobile substitute. A goal of 99.999% availability is typical.

Hardware is unreliable when thousands of machines are used. Services must adapt to failures fast. If the failure lasts more than a few seconds an replacement is provisioned before a second or third failure is likely.

Data centers generally have high levels of redundancy. There are usually two or more sources of power plus a UPS system, and the UPS is increasingly fine grained. Networks have multiple redundant paths.

The reliability focus is holistic for the whole center. Individual machines are good but inexpensive.

Replication with fault independence confers durability in the DC, regardless of media.



Recap

Cloud apps require some shift in priorities.

- Drive to get faster until responses are reliably below 50 msec (at least)
- Non-availability is never an option
- Availability done right is durability
- Services span thousands of machines
- Applications seen by users are typically aggregating multiple services in the data center
- Uniform low latency is a very strict requirement
- Services need to failover very fast
- Any given machine can and will fail
- Networks are fast to avoid stretch latency

Much of this is built into the infrastructure of each vendor’s clouds today. The sophistication of the infrastructure will increase. The variety of resources/services that can be used by cloud applications will continue to increase.