The Great Availability Caching Fallacy...

Through the 1970's the number of individual airline reservations systems in the world grew rapidly from a small handful to well over one hundred. Initially, the systems focussed on selling only the seats of the airlines owning and operating them, but pressure grew relentlessly to sell each others seats for interlined itineraries. The systems were connected to each other, primarily via the SITA network, but the communications facilities were primitive, slow, unreliable and expensive. Hence, a bright idea surfaced: Availability Caching.

In its initial form, called AVS (AVailability Status), this capability spread like wildfire through the reservations systems from the late 1970's. Basically, the principle was that each airline informed all its partners of the flights which had seats available to be sold and these could be sold freely by all until such time as a subsequent message arrived saying stop selling (because the flight is nearly full). The AVS messages between the systems were basically asynchronous standard format telexes; i.e. not real-time, but store-and-forward messages appearing on queues and which the systems processed automatically. The trigger thresholds telling partners to sell or to stop were fairly coarse and allowed for the time delay of the primitive network. However, in those days yields were high, critical load factors were low and having some unsold seats on every flight was not an issue.

As we progressed through the 1980's and into deregulation, competition grew, yields dropped and critical load factors went up and up. Selling every possible seat on every possible flight by every possible means became an economic imperative. In parallel, the inter-system communications facilities improved in leaps and bounds, both in terms of reliability and speed. Hence, the use of AVS grew explosively, the trigger levels were set ever more finely and the whole inter-connected complex of reservations systems came to rely on the expectation that AVS was actually being transmitted and processed in real-time. Most of the time, it actually was, but not all the time. Compounding this shortcoming was the fact that the sheer volume of AVS messages was becoming a bottleneck in the system. Every time a few seats were sold on a flight, causing the availability to go below the trigger level, an AVS went out saying stop (one message, but with 20, 40, 60, etc. addressees getting a copy to be processed). Later there were some cancellations on that flight (the travelling public was getting ever more fickle), causing the availability to exceed the trigger level, and out went an AVS saying sell to all those same addressees. This was further compounded by the introduction of numeric availability in the late 1980's. Simple yes/no AVS messages were augmented by Numeric AVS (dubbed AVN): "I have 9 left on this flight", then "I have 6 left on this flight", then "I have 2 left on this flight" and later "oops, now there are 7 left on this flight". By this stage AVS was threatening to swamp the whole system. I remember one occasion where the American Airlines Sabre system had a major schedule change and their faithful ally, Air New Zealand, ended up with so many AVS's on the queue that it would have taken their small system 36 hours to process them all, so being practical people they simply threw the first 32 hours worth away and just processed the remainder. There is nothing real-time nor reliable about that.

By the end of the 1980's AVS was a major bottleneck, but we had a saviour: the GDS's. Part of the basic business model of both Amadeus and Galileo was to impose seamless availability and secure sell on all the partners. Sabre and Worldspan soon followed and by the mid 1990's availability caching was restricted to bi-lateral AVN agreements between GDS's and their partner airlines. The introduction of seamless availability coincided with another development arising from the ever increasing competitive pressure in the airline business: revenue management. With the implementation of sophisticated revenue management logic, particularly O&D logic, into the availability determination, availability no longer directly related to whether

there were seats actually available on a flight or not. Availability now means: "am I willing to sell a seat on this flight, in the requested fare class, at this time, given who and where the passenger is in combination with his entire itinerary, and irrespective of whether the flight is actually empty, full or overbooked". Each major airline reservations system has unique logic to determine its answer. Hence, availability might tell me that I can have a seat, but my neighbour is told there are no seats, even though he requested exactly the same as I did at the same time.

Then came Internet, e-commerce and an explosion of travel portals. Those who do not learn from history are doomed to repeat it, so the gurus of the e-commerce world of travel portals think that this wonderful new concept of availability caching in the portals themselves is a solution to a problem. My contention is that the problem is not understood nor has it been analysed and defined. The proposed caching solution will only work in simple cases (basically systems which have no or simple revenue management logic; e.g. low cost carriers), but even then, it will not work always. With caching, customers will be told they can book an itinerary, but when they say go ahead, the reservations system refuses to do it. Even worse, is that customers will be told they cannot book something, when in fact the reservations system would be perfectly prepared to allow them to do so. Since the availability as the reservations system defines it is so complex, it can change in theory every few hundred milliseconds from yes to no and back again.

Hence, let us look at why the travel portal gurus think they need to do caching. Let us analyse the problem. We are told that caching solves two problems: response time and price. Response time, given the obsolete technology employed by most of the components involved in the process, including the portals themselves, is indeed a major problem. Price, however, is a red herring, since almost without exception, availability requests to GDS's and/or individual airline systems are free. Hence, why spend money in terms of hardware, software, operations and monitoring to imperfectly replicate something which you currently get for free?

The problem thus is purely response time. Given the current set up, what the response time to an availability request actually is depends on what you are asking and to whom you are addressing the question. An extreme example would be: if you are logged into the Transavia site and ask for availability on flights from Amsterdam to anywhere departing September, returning October, the system gives you 1 month outbound and 1 month inbound availability in 2-4 seconds without any caching whatsoever. Assuming you could ask the same question of a travel portal connected to a GDS, which in turn is connected for seamless availability to KLM, where Transavia is hosted, then the answer would probably take 1 to 2 minutes. To understand why this is so, you have to understand the underlying systems in the chain and what they are capable of. All the systems have the prerequisite architectural capabilities (TCP/IP and XML) to do what Transavia does, but most of them have not applied those capabilities to the problem. To convince them all to do so, to agree the standards and to ensure a coordinated implementation of what is necessary in all the vast multitude of inter-system links between the 4 major GDS's and the approximately 50 individual reservations systems remaining would be a gargantuan task.

Hence, on availability caching for portals we are caught between a rock and a hard place…

Bruce Taylor & co.
Amsterdam, September 2003