Two-Level Market Data Distribution Architectures: Are They Really Necessary?

15 July, 2001

This article discusses the following question: is the two-level architecture of many market data distribution architectures really necessary given the recent advances in networking infrastructure and protocols? We use the Reuters Market Data System as an example to provide a historical summary of "how did we get here?", and discuss the pros and cons of a single-level architecture.

Introduction

We begin, in Section 2. A Brief History of Two-Level Architectures, by providing a historical account of some of the most popular two-level market data distribution architectures: Reuters RMDS, Reuters Triarch and TIBCO's SASS platform. This is followed, in Section 3. Advances in Network Infrastructure and Protocols, by a brief summary of some important recent advances in networking technologies.

Readers who have a solid understanding of the two-level architecture of market data distribution platforms may wish to skip the historical descriptions of such systems, and jump directly to the hypothetical scenario of a one-level architecture, described in Section 4. A Single-Level Market Data

We finish with a summary in Section 5. Summary, and a few final thoughts in Section 6. What Is the Purpose of this Article?

A Brief History of Two-Level Architectures

The two-level architecture of the Reuters Market Data System and its precursors - the Reuters Triarch and TIBCO SASS platforms - have proven to be reliable and scalable platforms for distributing market data, especially in a high-volume and/or large subscriber base environment.

The primary component of the two-level architecture is the RTIC (Reuters TSA Information Cache), which takes the place of Triarch's Sink Distributor and TIBCO's TIC software. The main purpose of the RTIC is to provide current values and updates for subscribing client applications.

History of the TIC

TIBCO created the TIC software (and associated SASS protocol) over ten years ago, primarily to deal with high-volume broadcast feeds, such as the MarketFeed 2000 feed from Reuters. The MarketFeed 2000 feed was a digital broadcast feed providing up to 250,000 different instruments (this number eventually approached or exceeded one million instruments). Client applications would typically subscribe to a small percentage (less than 5%) of the entire universe of instruments, so it would have been extremely wasteful to broadcast all updates directly to the client network(s), especially on the 10Mb shared networks, common in those days. Furthermore, and most importantly, the data was published as UDP broadcast data.

The TIC provided a way to cache all of the instruments published by the MarketFeed 2000 feed handlers and to only forward updates to client applications for those instruments that had been subscribed to. Later, the SASS protocol was extended to handle interactive feeds, such as the Reuters Quote Feed. In-

teractive feeds generally only publish the instruments that have been requested by client applications.

The TIC software (and resulting two-level network architecture) was typically set up in the following network environment:

- Communication over TIB was via UDP broadcast.
- Client networks were shared LAN segments (often 10Mb) connected via bridges and/or routers.

The network environment that TICs currently run in is significantly different from the earlier network environments. These differences are key to our later discussions about whether the RTIC and a two-level network architecture are really necessary.

History of the Sink Distributor

Triarch's Sink Distributor software was introduced to serve a similar purpose to the TIBCO TIC software in that it caches data from feed handlers and distributes it on an as-needed basis to client applications. Unlike the UDP broadcast nature of TIBCO's TIB, the Sink Distributor originally distributed data to client applications using point-to-point TCP/IP connections. In large networks, it was necessary to have a hierarchy of cascading Sink Distributors to achieve the necessary scalability.

Reuters RTIC

The RTIC makes use of the best features of both the TIB and Triarch. On the client application side, the RTIC functions much like TIBCO's TIC, providing initial values and forwarding updates for only those instruments for which clients have registered an interest. Communication is over TIB/Rendezvous. On the feed handler side, the RTIC functions much like Triarch's SSL protocol, with facilities for load balancing and failover.

Advances in Network Infrastructure and Protocols

In recent years, significant technological advances in network hardware and protocols have made TIBCO and Triarch networks much more reliable than in the early days of these systems. Two key advances in this area are:

- IP multicast
- Widespread use of network switches, replacing hubs, bridges and routers

The combination of IP multicast and network switches has provided network architects with a great deal more flexibility in designing their TIB-based networks.

IP multicast provides an interest-driven approach to delivering what otherwise appears as broadcast data. IP multicast is extremely useful in TIB networks because it can significantly reduce the number of TIB/Rendezvous daemons that unnecessarily communicate with each other. It is beyond the scope of this article to describe the many ways the IP multicast can benefit the scalability and reliability of TIB networks, but its impact is significant - more so even than many TIB network managers realize.

Network switches provide the ability to only forward network traffic to those hosts for which the data is addressed. For example, on a basic shared LAN segment, point-to-point and multicast traffic usually hits all nodes connected to the hub, even though many/most hosts on the shared LAN segment are not logical destinations for the data. Network switches are smart enough to only forward "interesting" point-to-point and multicast data to each host. This is a rather simplistic view of how network switches operate, but it is

sufficient for the purposes of this article.

A Single-Level Market Data Distribution Architecture

Suppose for a moment that we put the cache-and-forward functionality of the RTIC directly into the feed handlers and attach feed handlers and clients to the same logical network. Assume that feed handlers only publish data for which client applications have registered an interest.

What have we lost with a single-level architecture?

Actually: very little, if anything. Most importantly, there would be no additional network or CPU load (e.g. TIB/RV daemon) on client machines because feed handlers, even feed handlers for broadcast feeds, would only publish data subscribed to by client applications.

Logically, and from a network perspective, it makes no difference to client applications whether the data is sent (via IP multicast over TIB/RV) from a feed handler or from a RTIC. Feed handlers can easily provide initial values, just as the RTIC currently does. Indeed, all interactive feed handlers already (sort of) do this. Of course, feed handlers also provide updates, which are normally received by TICs and forwarded to client networks.

In order to meet our assumption that feed handlers only publish data for which client applications have registered an interest, broadcast-based feed handlers would have to become interest-driven, much like an interactive feed. This is a relatively simple feature to add to broadcast feeds.

The only additional load on feed handlers is the requirement to provide initial values to new subscribing clients.

What have we gained?

For a start, we have cut the total amount of network traffic by at least one half, although the actual reduction in network traffic with respect to the client network is zero, since client applications must ultimately receive the same data in both architectures. Note, however, that at many companies the same network switches are responsible for managing both networks in the two-level architecture, so the overall load on network switches is reduced. Fewer overall network resources are required: fewer switch ports and (possibly) fewer switches.

There is also less software to manage (and purchase). No RTIC is required. Feed handlers may become (slightly) more complicated than they are today (e.g. in order to handle client subscription counts), but no more complicated than the RTIC.

Finally, we have gained some minor speed improvement (probably less than 100 milliseconds), since the single-level architecture removes the latency of two networks and intermediate software.

What's missing?

Some other features of the current two-level architecture also need to be considered:

- Load balancing and fault tolerance
- Value-added "mid-level" services

Load balancing and fault tolerance

There is no reason that the load balancing and fault tolerance features of the RMDS architecture (like the

TIBCO SASS architecture's GSM) could not be reproduced in the absence of an RTIC.

For broadcast feeds, load balancing is not a necessity, since one member of a fault tolerant pair of broadcast feed handlers can handle all client subscriptions. Hence, no separate server process is needed to broker client subscription requests for broadcast feed handlers. Note that it still may be useful, though not at all necessary, to provide load balancing for broadcast feeds. For example, under normal operating conditions, both instances of a fault tolerant pair could each publish half of the subscribed instruments. However, if one feed handler fails, the entire load is still passed onto the other feed handler. Fault tolerance for broadcast feed handlers can be easily achieved using a simple heartbeat mechanism.

For interactive feeds, load balancing is still required. A separate process would probably still be needed to broker the incoming requests from the client applications. "N+1" load balancing and failure could, in fact, become a built-in part of interactive feed handler software.

Note that any network traffic generated by a load balancing component (which is isolated from client networks in the two-level architecture) would have no impact to client nodes in a single-level architecture as long as separate multicast addresses are used for this type of communication.

Value-added "mid-level" services

A typical two-level architecture contains applications that act as both subscribers and publishers. These applications are generally connected to both the client and feed networks, hence the term "mid-level" services. They often act as feed handlers publishing into RTICs, which then forward updates to client applications.

There is no reason that these same mid-level applications, in a single-level architecture, would add any more load to the client network than in a two-level architecture.

Summary

In a switched network environment with IP multicast, a single-level network architecture can provide exactly the same functionality with respect to client (subscriber) nodes without introducing any additional load and risk. It does so with fewer components and by reducing total network traffic in half (or more for broadcast feed handlers).

This type of architecture was not feasible a few years ago, when standard market data distribution platforms were comprised of shared LANs (i.e., hubs, bridges and routers), and network protocols used for data distribution were point-to-point TCP/IP and UDP broadcast. Switched networks with IP multicast provide a much more flexible and scalable framework for market data distribution.

A single-network architecture, with fewer components and a lighter overall network load, would presumably cost less to install, administer and maintain than a dual-network architecture.

What Is the Purpose of this Article?

Firstly, what is it not? The leading market data distribution platforms, such as Reuters' RMDS and Triarch, and TIBCO's SASS are proven architectures for market data delivery. This article is not intended to suggest that companies should throw out such systems, if they already have them. Indeed, there is no simple way to convert a dual-level network into a single-level network, primarily because the feed handlers currently communicate with the record cache (e.g. RTIC) using a different message format and/or API than the record cache communicates with client applications.

As a provider of market data feed handlers and infrastructure components, Wombat Consulting will ensure that its products will work just as well in a single-network architecture as in a two-network architecture. In addition to continuing to support the two-level architecture, Wombat intends to provide the ability for its feed handlers to publish directly into a single-level architecture - by itself, or alongside a two-level architecture.

A single-level architecture could be based on a simple TIB/RV reliable publish/subscribe backbone, or it could some other platform, including existing in-house architectures. Wombat's market data products will operate in virtually any of these scenarios.