Logo will appear here

FSG Pricing Optimization

Problem Landscape | Identified Problems and Proposed Solutions | Project Artifacts

Problem:

The problem in focus is the one in querying live data via Microsoft Access. As it is currently understood, there are two queries that take far too long to complete. There are two tables being dynamically created in Access through these queries, the All Missing PricesandAll Missing Rates tables. On the surface, this was not an issue. The tablescreatedendupbeing under 1 MB while the limit for Microsoft Access is a whopping 2GB. However, as funds have been added and due to the natureofsubaccounting, the query itself pulls much more than the 2GB limit. The problem lies with the fact that the tables containing all the necessary information are both dynamic and temporary. Therefore, we recommend implementing the following solution to address this issue.

Sponsors:

Jinkins, Crystal M <CMJinkins@dstsystems.com >; Cady, Thomas J <TCADY@dstsystems.com >; Colwell, Arthur <AColwell@dstsystems.com >

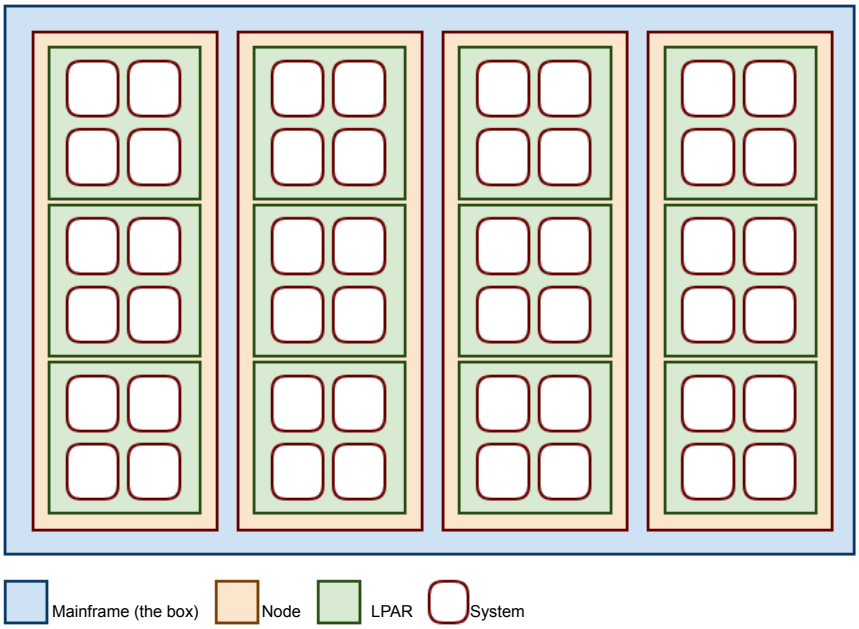
Proposed Solution:

The ultimate goal is to speed up the request for information related to all missing prices and funds. To that end, we believe we need information stored semi-permanently in the form of a modern DBMS implementation. The new database will be client facing and have three tables in it: All Known Funds, Price History, and Rate History. These tables will be queried directly rather than going platform at a time and jumping across nodes in the mainframe. The idea is that we'Ã‚Â™ll be able to similarly pull information as always, but stepping outside of Microsoft Access and its implicit 2GB limit will allow for a much more scalable system.

Currently, there is a messaging system in place known as MQ. It currently sends pricing information to the mainframe to allow for the automatic transmission of fund prices. We suggest a second messaging queue be implemented in the exact same manner as the first, but these messages will instead populate the tables on the modern DBMS. The pricing associates will then have a dashboard connected to those tables so they can visualize all the relevant information they need in real time. This will involve some work from the iSeries team, but it only requires that they implement another queue similar to the current one, which is a highly favorable alternative to an update to the mainframe infrastructure. This will also involve some middleware implementation and, of course, visualization of the dashboard itself. These two tasks can be handled for the most part by the Toronto R&D team.

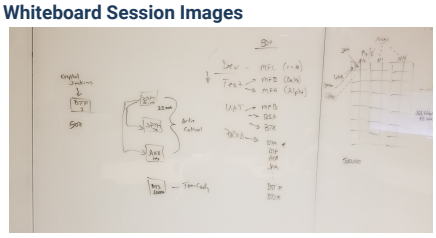
Pricing is done by at least three different areas: Full Service, Sub-Accounting and BFDS. Each area needs to see the same sort of information, but it needs to be scoped to just the Systems and Fund Sponsors of interest to each area. As such, we need to consider using profiles that capture the scoping needed.

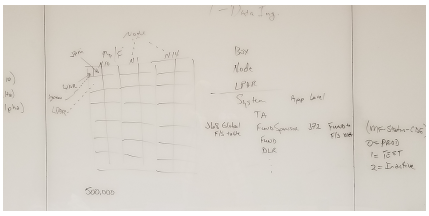
Mainframe 101
The physical hardware (aka the box) is composed of nodes which are further divided into LPARs (logical partitions), this is true of any mainframe. The use of the LPARs are application dependent. For our purposes and understanding, the LPARs are divided into systems. Systems may be dedicated to a single client or shared between multiple clients. Each system has its own database schema, meaning the pricing (and rates and distributions) data we are interested in is spread across multiple systems.

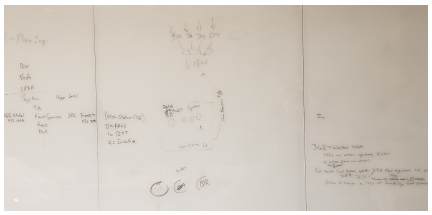


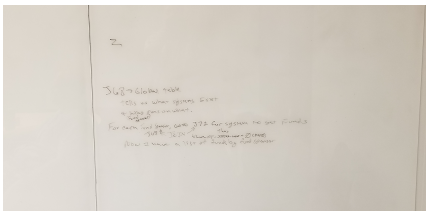
Funds are priced daily and as the prices come into the `systems` the `GT4` (Pricing Table) tables are updated. These price updates trigger an existing sub-routine to send messages to an MQ where consumers can respond as needed. In Feb. 2018 changes were added to the sub-routine by a Client Development team (James Rao, David Churn, David Jones, et al.) to tap into the messaging to process for the purpose of sending messages out of the mainframe to the EDP. The solution was to create a "remote" Q (simply a pass-through Q) on each `LPAR` to forward its messages to the appropriate Open Systems MQ. The result is a many-to-one solution were many LPAR MQs forward to a single Open System MQ. This applies to Test, UAT and Prod platforms as shown below.

Whiteboard Session Images



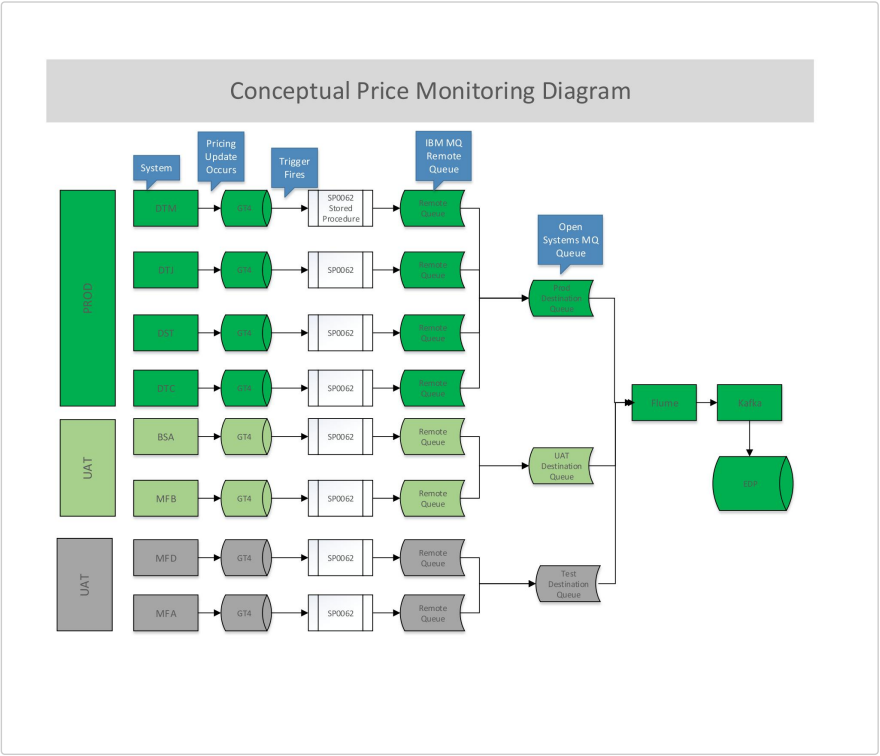






High Level MQ/Open Systems solution - Note, this is prices only. Rates to be included down the road, but should follow a similar model.

Updated: 10 May 2018



Initial concept diagram.

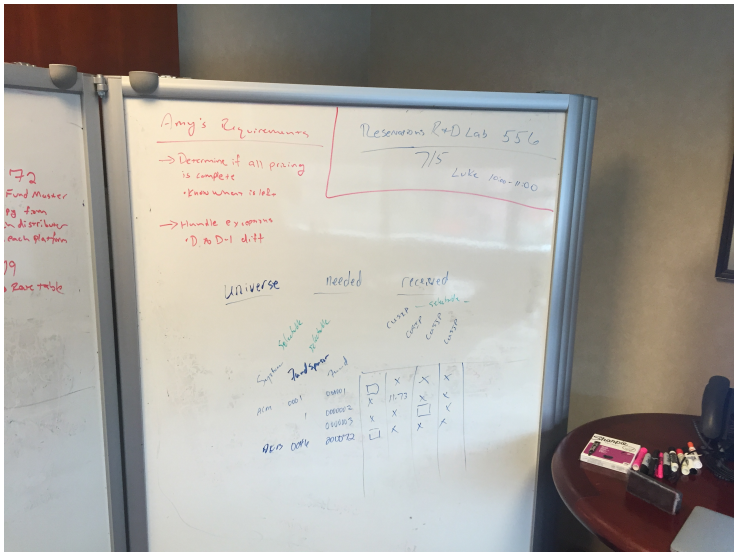
Pricing Message

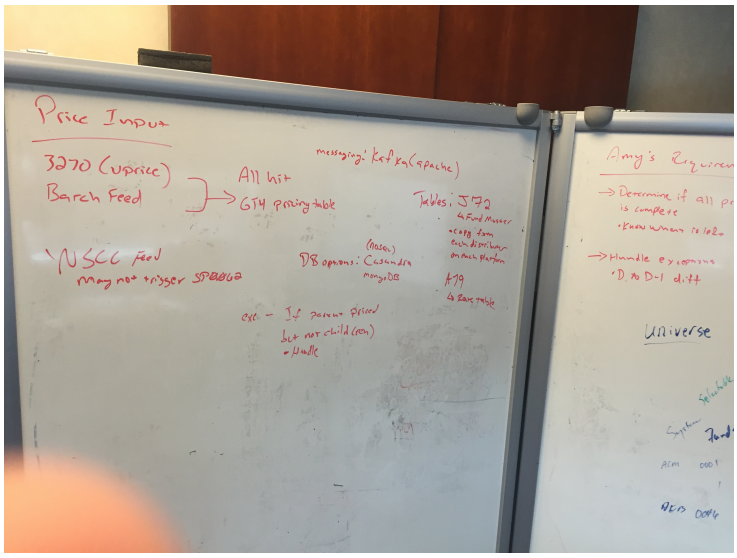
As of **10 May 2018**, just over 500,000 funds exist on PROD. The expectation is at least one message per fund a day Monday through Friday during the pricing window of roughly 5:00 PM to 8:00 PM. Message size is 329 bytes.

SP0062 MQ JSON Message Format

```
{
    "SYSTEM_ID": "MFL",         <<< CHAR(3)system the message is from
    "TABLE": "GT4",             <<< CHAR(3)pricing table name
    "ACTION": "I",              <<< CHAR(1)database action causing this message(I)nsert,    (U)pdate, or(D)elete
    "SEND_DATE_TIME": "ccyy-mm-dd-hh.mn.ss.micros",
    "DATA": {
        "FUND_CODE": 1234567,
        "MF_PRICE_DT": "ccyy-mm-dd",
        "MF_PR_SCH_TYP_CD": "1",
        "MF_PRICE_HH_TM": hh,
        "MF_PRICE_MM_TM": mn,
        "LAST_MNT_DATE_TIME": "ccyy-mm-dd-hh.mn.ss.micros",
        "MF_NAV_PR": 12345.1234,
        "MF_PR_EQZN_FCT_RT": 1234567890.12345
    }
}
```

Initial White-boarding session with Roger Stanley and Dan Greene...
Here are the formats of the pricing message. We will receive one for inserts and another for updates.





What we need to know in EDP side:

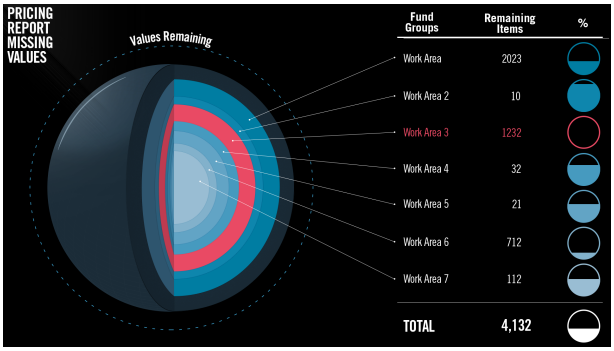
We will need to query the EDP for all funds that need to be priced, as well as which funds need rates. The EDP will replicate the necessary tables in order to provide this information. **It must be determined at what interval the data needs to be updated (hourly, daily, weekly)**. The following tables are replicated in the EDP and contain the data needed to build a dashboard:

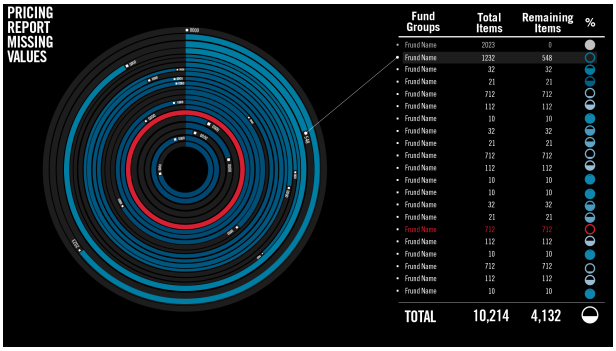
J68	Global Fund Sponsor Table Can get the list of existing systems Can get the list of Fund Sponsors that exist per system
J72	Mutual Fund Table Get the list of Funds per Fund Sponsor that need to be priced MF-STATUS-CDE determines "active" funds; 0=Prod (aka active), 1=Test, 2=Inactive **J72 Copybook** **Example J72 extracts:** **J72 BSA.txt** **J72 MFB.txt**

GT4	Mutual Fund Price Table Get count or list of Funds that have not yet been priced (today)

Dashboards

Workingthroughsomedashboard examples. We need 2. One that is up at all times for everyone to see, and another where the price validators can do research on what is not priced.





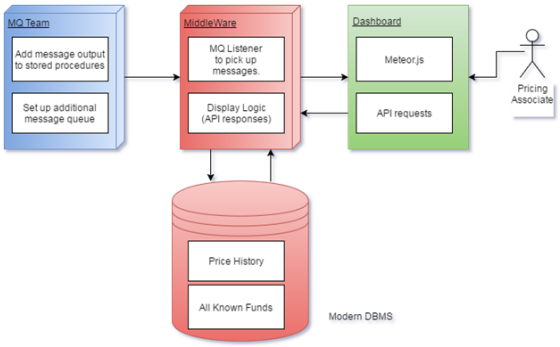
How will this work

Each time a fund gets priced (from a data feed from a customer, NSCC feed, manual entry into UPrice, other), an edit is made to the GT4 pricing table for the associated fund family/sub accountingnode. Currently, it the SP0062 stored procedure is triggered on any edits to GT4. The SP0062 determines if the fund record is a parent of child records (the samecusipthat exists in othersub accountingnodes), or if it is a child. If a parent, a message is sent via MQ to the node(s) where the children exist. If it is a child, nothing else occurs.

We are exploring making changes to the SP0062 to essentially intercept any updates to GT4, and send another message to a queue that is being monitored on the Open Systems side. We will have our own queue listeners in open systems that will monitor the new queue, andkeeprecordofall pricing that has been done.

How we can utilize Message Queues to our Advantage

The idea here is that for every transaction that takes place, that is, every time a price or rate is pushed through into the database, one of two processes takes place. One process is for depositing a rate into the system and the other for storing a price. When these pieces of information are being pushed, a message is generated in a message queue. We can pull each message (but not delete it) and then generate a list of Funds that have been priced.Nowunfortunately, that is not exactly whatthepricersneed. But that is fine;insteadwewill just need to get a table that has every fund that needs to be priced, but only once. Once we have that Required to be priced list, we can just begin pulling funds from it as messages begin coming in.



Summary

From our perspective, it seems that Microsoft Access can be an extreme hindrance to the Pricing Team getting their job done on time. In fact, the relianceofAccess is troubling as it is causing major slowdowns and with more funds being added in September things will only get worse. Our proposal is to move away from Microsoft Access in favor of an interactive dashboard implementation. This dashboard would ideally include the ability to see what funds need to be priced on both a master level and by platform. From this point,wewi™llrequest feedback about what exactly the pricing team needs to see in that dashboard and move forward from there.