Granule Reconciliation with CMR

If you have any useful posted documentation on using CMR to do Granule Reconciliation, please point me to it. If anyone (CMR or Providers) have opinions about indexing LastUpdate or ideas on how to reconcile granules vs CMR, please chime in.

I'd like to reconcile my provider granule records against CMR using a CMR (REST) interface. In the past I reconciled LAADS provider holdings vs ECHO using DataManagementService API GetDatasetInformation (WSDL+FTP) for a specified collection and LastUpdate time range covering a few days to a few weeks at a time. In my database I have a table of products sent to ECHO that is indexed both by GranuleUR and collection+LastUpdate. At the moment LastUpdate is not searchable in CMR. I'm told it could be indexed and made available for search, but whether that is worth doing might depend on whether it is useful to others or not. With FTP ingest it was necessary to reconcile recently sent metadata once, making the LastUpdate time condition critically useful; but with REST I can skip that as I assume if CMR gives me an accept message the granule metadata won't be dropped after that.

The most obvious alternative might be to search by revision_date. Under normal circumstances with REST that is presumably slightly greater than the LastUpdate I supply which is my time when I gather metadata to send for about 10 products at a time. (That time is useful for me to record because for example if I get an associated browse product after that time, I queue the data product to be resent in 15 minutes with whatever browse URLs are then available.) However the REST site might be unavailable for 15 minutes or take minutes to respond as I've seen ECHO REST do, there are always small clock skews, and until last week all LAADS granules were ingested via ECHO FTP which sometimes took days. All of that makes comparison of LastUpdate to revision_date difficult. While I might be able to add revision_date to my table, I don't know if it even gets returned in a granule accept message or would require a separate query. What I like about LastUpdate (vs revision_date) is that I supply it and you record it, so we both agree on the value.

The second obvious alternative might be to search by GranuleUR range. However in my system I store the numeric part of my GranuleUR and CMR search would consider it a character field. I could possibly create a functional covert to varchar index on my (Postgresql) table to make the search useful - that may be the easiest solution under my control assuming I only care about full reconciliation and not reconciliation of metadata sent in a specified (e.g. recent) time range. (Our reprocessing usually reuses old GranuleURs, so our GranuleUR has no relation to any range of LastUpdate.)

Finally as an aside I'm interested in some idea about how indexing works in your system. Specifically if I query on specified collection/entry_id (short_name + version) and a range of say revision_date, does your system use composite indexing (entry_id + revision_date), would it only use the most useful of those indexes or do some sort of bit map join, or are all search conditions in some way useful as indexes rather than just filters? Your CMR Client Partner User Guide#BestPracticesforQueries seems to imply that specifying collection + range is helpful - does that mean your searchable attributes are indexed across all collections and within each collection, or just within each collection for some attributes? Is a separate collection + GranuleUR range query for each of my collections better than a single query of provider + GranuleUR range?