

1. Tuning Parameters

What are the key tuning parameters? What do they affect, and how are they changed (via configuration file, database, or GUI)?

- **How large should the Application Log Size be set?**

Ingest - Have not tested to see what is a good application log size. This is set in the config files – the AppLogSize parameter. The default value currently set in .cfgparms for all config files is 50,000.

DPS - AppLogSize = 200000

Science Data Server - AppLogSize = 5000000. The default in the config file is 1000000. This value can be changed in the SDSRV and HDF EOS configuration files via the Mkcfg process.

Data Distribution - Determined by parameter AppLogSize in .CFG file. Current set to 1M.

1.1 Ingest

- **How many database connections should be set in the config files?**

For single-threaded programs, only need one database connection (EclnPolling, EclnGUI, EclnAuto).

For EclnGran, it depends on the maximum number of granules that can be processed at a time (see TotalGranuleThreshold from the InGranuleServerInfo database table). For a maximum of 5 granules, 2 database connections are probably enough. For EclnReqMgr, there should be at least 2 database connections. During the ETE testing at EDC, only 3 database connections were actually needed by EclnReqMgr. The number of database connections is set in the config files – the INGEST_CONNECTION_POOL_SIZE parameter. If the parameter is not set, then the default value is 10.

- **How many granule servers should a DAAC be using?**

Each granule server can process multiple Earth Science Data Types (ESDTs), but each ESDT can only be assigned to one Granule Server. At EDC, there are two Granule Servers configured – one to process Landsat-7 data and the other to process ASTER data. Also, currently all Granule

Servers run on the same machine. Distributing them to different machines would possibly do more to enhance performance.

In order to configure a Granule Server, there needs to be a row in the InGranuleServerInfo and InValGranuleServerUR tables. In order for a particular ESDT to be processed by a particular Granule Server, the GranuleServerURKey must be set to the appropriate Granule Server in the InDataTypeTemplate table for each data type. The InGranuleServerInfo, InValGranuleServerUR, and InDataTypeTemplate tables must be set manually via SQL commands into the Ingest database.

- **How many listening threads should be established?**

The number of listen threads is set for EclnAuto and EclnReqMgr in their config files using the ListenThreads parameter.

The current value set in .cfgparms for ECS Assist for EclnAuto is 7 and for EclnReqMgr is 30.

The config file ListenThreads parameter for EclnGran is not used.

The number of listen threads for a Granule Server is set in the software using the TotalGranuleThreshold column in the InGranuleServerInfo table multiplied by 3. The default value for the TotalGranuleThreshold is currently 40. It is not recommended to make it larger because then the Granule Server will grow too large. (If you are running with three modes active, especially with more than one Granule Server per mode (e.g., EDC has two Granule Servers), then you may run out of swap space on the icg machine very quickly.) The TotalGranuleThreshold determines the maximum number of granules that a Granule Server will process at any one time.

- **How do you set the number of queued requests?**

Request Manager receives requests and breaks them into granules and then queues all the granules. If the appropriate Granule Server is not processing its maximum granules (TotalGranuleThreshold from the InGranuleServerInfo table), then one or more granules will be removed from the queue and sent to the Granule Server. So there is not really a way to set the number of queued requests.

There is a maximum number of requests and volume which can be processed by Ingest at one time. These are the MaximumTotalRequests and MaximumTotalVolume columns in the InSystemParameters table. These parameters can be set via the Ingest GUI. When a request comes in from one of the clients (GUI, Polling or Auto) to Request Manager and it causes

one of these maximums to be exceeded, then the request is failed and will not be sent to Request Manager.

- **What is the maximum number of requests from data provider?**

For each data provider, there is a maximum number of requests and a maximum data volume. These are the MaximumRequests and VolumeThreshold columns in the InExternalDataProviderInfo table. These can be set via the Ingest GUI. When a request comes in from one of the clients (GUI, Polling or Auto) to Request Manager, and it causes one of these maximums to be exceeded, then the request is failed and will not be sent to Request Manager.

- **What are other key parameters for Ingest?**

In the InSystemParameters table there is a parameter called MonitorTimeForCompletedRequest. (Probably can be set via Ingest GUI, but Check this one!) This is the number of minutes that a request, once completed, will remain in the database tables that get displayed on the Ingest Monitor and Control GUI window. After the time has elapsed, then the request will be moved to the database summary tables and can then be viewed using the Ingest GUI History Log window.

In the InSystemParameters table there is a ScreenUpdateInterval parameter. It is the number of seconds at which the GUI will refresh. This can be set via the Ingest GUI.

In the EclnAuto.CFG config file, the RETRY_TIME_INTERVAL is the number of seconds before Auto will retry to send a request to Request Manager. If the parameter is not in the config file, then the default used is 600 seconds.

In The EclnGUI.CFG, EclnReqMgr.CFG and EclnGran.CFG config files there is a SAVEONEXIT parameter. If it is not there, then the default used is false. It only needs to be set to true for debug purposes. When it is true, staging disks will not get cleaned up. For Granule Server, when the SAVEONEXIT parameter is true, the local preprocessing disk also will not get cleaned up.

In the EclnGran.CFG config file, there are two parameters for retrying rpcs to SDSRV, SDSRV_RETRY_INTERVAL and SDSRV_RETRY_ATTEMPTS. The SDSRV_RETRY_INTERVAL is in seconds.

In the InSystemParameters table, there are two parameters for retrying to send a DDN to the Landsat7Gateway, CommunicationRetryCount and CommunicationRetryInterval. (Check if on GUI) The CommunicationRetryInterval is in seconds. The default values installed with

the database are 5 for both parameters. When a DDN is sent, the default rebinding policy of 5 times every 15 seconds is used. The CommunicationRetryCount and CommunicationRetryInterval are used for retrying the rebinding. For example, if the default values are used, then the DDN will be tried a total of 25 times.

In the EclnReqMgr.CFG config file, there are two parameters for retrying rpcs to Advertising, ADVERTISE_RETRY_WAIT_TIME and ADVERTISE_RETRIES. The time is in seconds. There is also an ADVERTISING_SYBASE_LIMIT parameter. This is the number of ESDTs that will be sent to Advertising in one request. If it is set to 0, then all ESDTs from the Ingest database will be sent at once. This parameter only needs to be changed if Advertising comes up against a limitation by Sybase as to how many ESDTs it can handle in one request.

To change the Polling interval, the PollingTimerInterval parameter in the section for the appropriate data provider in the InPolling.cfg file can be changed. It is in seconds.

1.2 PDPS

- **How many retries should be set for SDSRV ?**

DpPr_MAX_RETRIES = 30; the amount of retries for acquires/inserts before giving up; and the time (in seconds) between retries is DpPr_WAIT_PERIOD = 120.

- **What are the key tuning parameters for Deletion Server?**

ListenThreads = 7

- **What are the key tuning parameters for Job Management?**

ListenThreads = 7

DpPrAutoSysMaxJobs = 3200; the maximum number of jobs in AutoSys at one time. Divide this by 8 to get the number of PGEs that can be in AutoSys at one time.

- **What are the key tuning parameters for Resource Management?**

DpPrRM_MAX_RETRIES = 100; the amount of retries when creating a Data Manager object.

DpPrRM_RETRY_PERIOD = 120; the amount of time (in seconds) between retries.

- **How are the database connections set for PLS?**

This is also handled in the .CFG files, but this is not a concern for PLS since our only server maintains only one connection to the database.

- **What else might be considered a key tuning parameter?**

SubMgr only:

The parameter SleepDelayForFailures determines how long (in seconds) the SubMgr waits after a failure to retry. The parameter SleepDelayForTimers, determine how long the SubMgr waits between checking for DPR timers.

PRE/SubMgr:

Other additional parameters are MaxSlippagePerc and AcceptableCertainty. MaxSlippagePerc determines what percentage a granule can slip by and still be considered a match. AcceptableCertainty determines how much overlap a granule must have at minimum.

1.3 Science Data Server

- **How many database connections should be set in the configfiles?**

There are several parameters in the SDSRV config file that control the number of database connections. DSS_NUM_GEN_CATALOGS controls how many DsMdCatalog objects get created within SDSRV on start up. This value has direct impact on the number of SYBASE connections held by SDSRV. The default for this parameter is 15. This parameter can be changed in the SDSRV configuration file via the Mkcfcg process.

SDSRV_CATALOG_CONNECT_INSTRUCTIONS controls how the DsMdCatalog connects to the database and has a direct impact on the total number of connections made. The default value is SQSOnly. The value "SybaseAndSQS" causes each DsMdCatalog to create two database connections, one to SQS and one directly to Sybase SQL. This option doubles the number of connections specified in the DSS_NUM_GEN_CATALOGS, but it offers significant performance gains by using direct SYBASE SQL Server connections when possible. The value "SQSOnly" causes each DsMdCatalog object to create only one database connection. This connection is to the SQS Server. This options requires less total connections, but lacks the performance improvements gained by using the direct SQL Server connections. The value "SybaseOnly" should not be used. This is for future use and does not support spatial data processing.

This parameter can be changed in the SDSRV configuration file via the Mkcfcg process.

- **What is the right number of HDF-EOS Servers?**

Note that only sites that are doing Landsat-7 distribution need to have HDF-EOS Servers. There should be a minimum of 3 HDF EOS Servers configured. Since the HDF EOS Servers now run on the SGI, the number of HDF Servers should be set when running make config on the Sun and the SGI. This number should be the same on both machines. The parameter NumOfHDFServer can be changed in the ACFG file via the Mkcfcg process.

- **When and how do ESDTs define backup copies of data (that is, when is the data archived in two separate places)?**

ESDTs are archived in two separate places when a value (*what value?*) is entered for backup id when the ESDT is installed via the SDSRV Gui. The value for the primary and backup archive are determined by the STMGT Cl. The values are stored in the DsGeESDTConfiguredType database table.

- **What are other key configuration parameters for the SDSRV?**

The SDSRV_AUTO_INSPECT_SWITCH is the SDSRV configuration parameter that controls whether SDSRV returns commonly inspected metadata attributes as part of the search request. The default value is on. When this feature is on, a RPC is eliminated for frequently inspected attributes. The SDSRV_AUTO_INSPECT_LIST configuration parameter controls which metadata attributes are returned with each granule as part of the Search service. The domain can include any metadata attribute name that occurs within the “top level” of the metadata GIPParameterList within SDSRV (any INSPECTable attribute).

There are two parameters associated with the HDF EOS Server restart, HDFEOS_RESTART_ABSOLUTE_MEMORY and HDFEOS_RESTART_RELATIVE_MEMORY. HDFEOS_RESTART_MEMORY sets the maximum memory that each HDF EOS Server can consume for all requests during the lifetime of the server. For 4PX, the number should be set to 1500000000. HDFEOS_RESTART_RELATIVE_MEMORY sets the maximum memory that each HDF EOS Server can consume for one single request. For 4PX, the number should be set to 2000000000. These numbers are set high so that the server will never restart. The values will change for 4PY and 5A. This functionality will only work correctly with a 5A SubAgent.

SDSRV_NUM_INT_SESSION controls the maximum number of concurrent sessions the SDSRV will start to execute asynchronous acquire requests that

are persistently stored in the queue. The default number is 128. If this number is set too low and there are a lot of Landsat requests, the other requests may never be serviced. There is an enhancement to this design for Drop 5B.

1.4 IDM

There are no significant tuning parameters for IDM servers.

1.5 Storage Management

- **What are the critical tuning parameters for STMGT?**

- Staging area, Read-only cache size

These parameters are tuned in tandem, from the STMGT GUI. They determine how much disk space is available for staging of files (both for Ingest and acquires), and how large the read-only cache is. When either area is exhausted, requests will hang until space becomes available.

- Checksumming

Checksumming can be enabled or disabled both for acquires and for inserts. The CHECKSUMSTATUS entry in the EcDsStArchiveServer.CFG file controls whether or not a checksum is calculated for each file inserted into the archive. The corresponding entry in the EcDsStStagingMonitorServer.CFG file controls whether or not a checksum is calculated when a file is acquired from the archive. Note that checksums are calculated on retrieval only when the file is first moved from the archive to the read-only cache. So long as the file remains resident in the read-only cache, the checksum will not be recalculated. Since checksum calculation is a highly time-consuming process, and is CPU-intensive, both of these servers can be significantly impacted by enabling checksumming.

- **How should the of listening threads be configured for Storage Management Servers?**

Server	# of listening threads	Comments
Pull Monitor	1	Pull Monitor is single threaded
Staging Disk	125	Maximum based on 1024 file descriptors

		divided by 8 file descriptors per request
Staging Monitor	125	same as Staging Disk
Archive	125	same as Staging Disk
FtpDisServer	<8	Based on DCE's practice of setting the inbound rpc queue to 8 times the number of listen threads. If there are more than 8 listen threads to FtpDis, then some of the requests being sent to Pull Monitor could be lost since the Pull Monitor has only one
IngestFtpServer	Derived – 125 Max	Based on the Number of InGranSrvs time the number of Listen Threads per InGranSrv. The The Maximum number of threads available to the INGESTFTP Server is 125.
D3Server	7	Only one request can be processed at a time.
8MMServer	30	Threads for each request from DDIST for media distribution plus extra for pinging the server.
PrintServer	Derived	Number of physical media devices being used for distribution of data up to a maximum of 125. Realistically, 1 would suffice in most situations since there are no long running rpcs being done by the print server.

- **How should the Caches (e.g., read-only cache) be sized?**

Pull Cache PullArea read-only cache is defaulted at 2,000,000,000 (blocks). There is no need for much user space because the user files are symbolically linked back to the files in the ROC.

Staging Read only cache depends on the server and the data being handled. Staging Read Only cache comes out of the same disk space as staging disk. The more need there is for staging disk, the smaller the size of the read only cache can be. If the staging disk is being used primarily to link files from the read only cache, then the read only cache can take most of the entire staging area (90% or so). If there is data being staged in support of Ingest or subsetting of data for distribution, then the read only cache will be smaller (say 40 or 50% of the total staging area).

- **How do you set ftp pull expiration time?**

It depends! The expiration time is used as a cleanup mechanism for files that have not been pulled. The factors determining the expiration time include:

- the frequency of file being left behind in the pull area (the more files the lower the expiration time)
- the size of the files (the bigger the files the lower the expiration time)
- the capacity of disk used (the higher the capacity used the lower the expiration time)
- the pull expiration time is defaulted to 24 hours

1.6 Data Distribution

- **How many database connections should be set in the configfiles?**

DBMaxConnections is currently set to 15 in DDIST “.CFG” file
DBMaxConnections is currently set to 4 in DDIST GUI “.CFG” file

- **How are priorities allocated to threads?**

Priority thread allocation is set in the DDIST DB table DsDdPriorityThread. The priorities are set via a perl script, EcDsPTEdit.pl, or, if the user is sql savvy, through command line sql. These seem to change a lot during testing when Ray or Steve Fox is around. The current defaults are:

LOW 28
NORMAL 128
HIGH 64
VHIGH 5
XPRESS 2

1.7 IDG

- **How many database connections should be set in the configfiles?**

Pre-set by Sybase SQL server. There is no such one in application-level.
(What is the number of db connections for the Subscription Server?)

- **What else might be considered a key tuning parameter?**

Set DebugLevel = 0 in the config file for Subscription Server.

1.8 MSS

- What might be considered a key tuning parameter?

SubAgent:

EcMsAgSubAgent.ACFG

```
{ "MsAg_LogMaxSize"      = "190000" },
{ "MsAg_LogLevel"       = "1" },
{ "MsAg_MssEventLevel"  = "1" },
{ "MsAg_LocalPollInterval_sec" = "15" },
{ "MsAg_MetricRefreshInterval_sec" = "2" },
{ "MsAg_DiscoverInterval_sec" = "1200" },
{ "MsAg_RetrySleep_sec"   = "30" },
{ "MsAg_NumRetries"      = "10" },
```

EcMsAgSubAgent.PCFG:

```
{ "MsAg_UpdateInterval_sec" = "99" },
{ "MsAg_AppShutdown_sec"    = 20 },
{ "MsAg_ProgShutdown_sec"   = 15 },
{ "MsAg_ProcShutdown_sec"   = 10 },
```

EcMsAgSubAgent.CFG:

```
DebugLevel = 3
```

DeputyAgent:

EcMsDeputyAgentApp.ACFG:

```
{ "MsAg_LogMaxSize"      = "10000" },
{ "MsAg_LogLevel"       = "0" },
{ "MsAg_RetrySleep_sec"  = "172800"},
```

EcMsAgDeputy.PCFG:

```
{ "MsAg_UpdateInterval_sec" = "99" },
{ "MsAg_AppShutdown_sec"    = 20 },
{ "MsAg_ProgShutdown_sec"   = 15 },
{ "MsAg_ProcShutdown_sec"   = 10 },
```

EcMsAgDeputy.CFG:

DebugLevel = 3

Ecsd:

None

OVMMap:

None

HP OpenView (COTS):

See COTS documentation