

N94- 35067

Telecommunications Issues of Intelligent Database Management for Ground Processing Systems in the EOS Era

Joseph D. Touch

USC / Information Sciences Institute (touch@isi.edu)

ABSTRACT^{1,2}

Future NASA Earth science missions, including the Earth Observing System (EOS), will be generating vast amounts of data that must be processed and stored at various locations around the world. Here we present a stepwise-refinement of the Intelligent Database Management (IDM) of the Distributed Active Archive Center (DAAC - one of seven regionally-located EOSDIS archive sites) architecture, to showcase the telecommunications issues involved. We develop this architecture into a general overall design. We show that the current evolution of protocols is sufficient to support IDM at Gbps rates over large distances. We also show that network design can accommodate a flexible data ingestion storage pipeline and a user extraction and visualization engine, without interference between the two.

1: Introduction

In addition to its manned space program, NASA runs telemetry-gathering missions. Among the celestial bodies studied is the Earth. Current and future Earth science missions (including EOS) will generate enormous amounts of data. This data must be archived in an accessible manner to be useful for analysis. EOS in particular will generate a continuous stream of 11.5 Mbps, which isn't notable except that the stream is relentless over the life of the satellite (about 5-10 years), resulting in 5.2 Gigabytes of data per hour, or 45 Petabytes (10^{15} bytes) per year.

1. This paper is based on a consulting report the author prepared while at the University of Pennsylvania. A version of this report will appear as part of the "HPCC Data Management white paper," of the NASA GSFC Information Science and Technology Office (ISTO). - Code 930.1.

2. This research was partially sponsored by the Advanced Research Projects Agency through Ft. Huachuca Contract No. DABT63-91-C-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Army, the Advanced Research Projects Agency, or the U.S. Government.

This data is processed prior to storage to facilitate access, and retrieved and converted into a useful form; these functions comprise the EOS Data Management Software System (DMSS), which is an example of a more general concept called Intelligent Database Management (IDM). Here we present an overview of the telecommunications issues of IDM, which involves data ingestion, storage, fusion, and rendering. Data ingestion is the processing of data prior to storage; data fusion is the combining of various streams of stored data to form a composite information base suitable for direct rendering. The components of IDM for EOS are distributed globally over large distances (over 2000 miles) and bandwidth (1 Giga-bit/second). Thus telecommunications issues, including latency reduction, high bandwidth protocols, and distributed resource allocation are a fundamental component of IDM.

Here we present a stepwise-refinement of the Distributed Active Archive Center (DAAC - one of seven regionally-located EOSDIS archive sites) architecture. We also discuss how current protocols are sufficient to support the IDM DAAC. We describe a network design that accommodates both a flexible data ingestion storage pipeline and a user extraction and visualization engine.

The most abstract description of DAAC is a set of continuous satellite data input streams (between 16 Mbps and 26 Kbps, totalling 25 Mbps average, 164 Mbps peak), and a 200-500 Mbps sporadic user visualization stream, with low BW user commands. Internally, the input and output are related only by storage, i.e., the input stream archiving and output stream generation are independent. We partition the continuous input archive stream (ingestion) from the user command and visualization streams (extract), both of which operate on the data store. There also may be multiple ingestion and extraction streams per DAAC. The general design proposed uses separate subnetworks of heterogeneous processors - one for ingestion and the other for extraction. The processors and subnetworks form a dynamically-configurable dataflow engine, where subnetwork partitioning inhibits interference and provides recon-

figurability. We show that the telecommunications aspects of IDM can be managed by this physical resource partitioning.

More importantly, we show that existing protocols, or existing proposals to evolve these protocols, are sufficient to support IDM. There is a growing controversy in protocol research involving the use of existing protocols for high speed (Gbps) wide area (2,000+ mile) environments. There are several protocol issues involved, including soft real-time delivery (i.e., jitter control), guaranteed bandwidth (reservation), and accommodation of high bandwidth-delay product links. Issues under investigation (without evolutionary solutions) include hard real-time delivery (scheduled delivery constraints), and methods for latency reduction. IDM data processing (both ingestion and visualization) requires isochronous data transfer, i.e., controlled jitter in transmission and processing. Fortunately, the data collection is automated and uses loosely-coupled feedback from ground control, rather than from user visualization. Latency reduction affects only the user visualization control loop. The user-perceived latency is likely to be affected more by the extraction processing latency than by the propagation latency (typically 100 ms). Existing evolutionary modifications to existing data transport protocols accommodate soft real-time transfer (RTP), bandwidth reservation (ST-II, RSVP), and high bandwidth-delay product links given continuous data streams (TCP extensions for LFN's).

Some of the initial documents of the IDM project have described various aspects of the project, but none has considered the specific telecommunications aspects within this project, or the impact of those issues on the other design considerations. This document section is an effort to augment those discussions sufficiently to suggest a design of the telecommunications system, from which other design criteria implications will be readily evident.

2: DMSS Background

The DMSS project is described by a set of documents that address the overall structure of the EOS and EOSDIS projects, information management, query derivation from user directives, data modeling, internal processing, and database and computation scaling of the IDM DAAC system. None yet includes a discussion of the telecommunications issues involved or of the implications of those issues on other design criteria. This is partly because telecommunications research is relegated to other projects of the HPCC effort, and because the telecommunications issues may not require original research.

One document describes the scaling issues of the database and processing components of the project, but admits that the processing load cannot be accurately determined a

priori [12]. This indicates that a scalable processing solution is required, one in which dynamic load configuration is possible.

Others describe the static issues of database and visualization access to the EOSDIS [5], or the data distribution and archiving requirements [6]. There are dynamic corollaries to these static issues that describe the reconfiguration of the system in a flexible way.

Management considerations mandate a relatively centralized facility or small set of facilities (the DAACs) [7]. Relieving a centralized load requires a distributed facility, provided that the data distribution is not orthogonal to the geographic configuration. Because the DAAC facilities are geographically distributed, processing within the DAAC should occur at a MAN or LAN scale. The partitioning of data into functional and operational sets among the DAACs indicates that inter-DAAC access, i.e., processing requiring the participation of more than a single DAAC, would be unlikely at first.

Data modeling is described using spatial, spectral, temporal, etc. characteristics [1]. This includes a description of the effects of the variation in access method on the storage organization. These descriptions can be easily augmented to include communications and dynamic post-processing costs, so as to describe the telecommunications effects on data organization as well. The only difficulty is that the telecommunications costs are distributed, whereas the access method frequencies proposed are local to a particular DAAC component.

Finally, the high-performance processing of satellite data before initial archiving requires the use of specialized equipment and systems [2], [9]. The individual board design of these components and the functional decomposition into processing elements is a scalable solution [9]. The board integration currently relies on existing technologies for system design (VME/VSIB), rather than on true networking of components. The resulting pipeline permits chaining of processing elements within a single processing node (i.e., VME/VSIB backplane), but not among different backplanes. Further, components of a single system can be used to process at most a single data stream, thus prohibiting an ultimately flexible design.

3: Observations

There are other observations that affect the overall telecommunications recommendations as considered herein. These include the software issues regarding protocols and support, and topological issues.

3.1: Telecommunications software issues

Many of the software issues are already being considered at various levels of the EOSDIS effort. The software can be partitioned into 5 main areas: front-end user interface intelligence and expert systems, back-end satellite data processing before archiving, archival processing for data management, and extraction processing execution. Each of these areas has implications for the telecommunications organization, but the extraction processing is especially influential.

The front-end user system involves expert systems [10], connectionism [8] or neural networks [3], and support for user visualization tools. All of these are user-level front-end issues, and can require sharing of scheduling information at the user-access level. This indicates a network of user-support systems, with loose coupling of state information on the availability of back-end resources, and other competing front-end sessions.

The back-end satellite processing system involves the use of specialized hardware [2], [9]. Each of these systems can be independent, as it processes a specific data stream from a given satellite individually.

The data processing for archiving can also be independent because of the partitioning of the databases among the DAAC sites [7]. Data reorganization is presumed to occur within operational units of the DAACs [1].

The extraction processing, however, has not been thoroughly considered in relation to the system design [12]. Current distributed systems design indicates a back-end network of dynamically-allocated processing elements, which can be configured according to the extraction processing needs of the users. Such a system is exhibited by a networked version of the back-end satellite processing system, with a few modifications (see below). The goal is a back-end network for the servicing of the user-level requests, according to the decompositions suggested by the expert systems at the front-end.

There are other services that are of use, especially in the software portion of the system. Current protocol technology is sufficient to support the data rates and characteristics of the large linear streams of information indicated by satellite measurements. These include TCP/IP, for streaming data transfer. Other options include remote evaluation (late-binding RPC), and the conventional remote procedure call (RPC). Conventional RPC requires sending the data to a remote site and retrieving the results, a mechanism that describes the dynamic allocation of processing components but requires a central controller to scatter and gather the data, creating a communications bottleneck that existing networks cannot support. Late-binding RPC permits the processing components to transmit the results along to subsequent RPC's, rather than requiring collec-

tion of the results at the originator of the first RPC. This permits a dynamic pipeline to be created within an existing telecommunications paradigm.

3.2: Network topology and protocol

Most of the telecommunications issues might be relieved by a LAN implementation. Exceeding LAN scale incurs a sharp decrease in data transmission rates. Existing routing, broadcast, and network management implementations also favor LAN scales. One solution is to distribute the access load among the components of a LAN, particularly existing high-speed LANs such as FDDI (100 Mbps) or FDDI-2 (200 Mbps). The FDDI protocol does not scale beyond a 100 meter diameter, but this is sufficient to support the locality of an individual DAAC.

The bandwidth requirements of this network (1 terabit/day) average out to a continuous 11.6 Megabits/second [12]. Conventional LAN technology (i.e., Ethernet) supports 10 Mbps, but only to a maximum of 80% load, i.e., 8 Mbps [11]. This theoretical maximum assumes a single source station on a network; competition among multiple sources decreases this to 60% (6 Mbps) [11]. Even a lightly loaded Ethernet is therefore unsuitable for even a single hop in the data path from satellite to disk storage.

New fiber optic LAN technology (FDDI) supports rates of 100-200 Mbps, large enough to support several simultaneous hops of the data stream if that stream is buffered and averaged over a 24-hour period (requiring 1 Terabit of tape delay). Burst data characteristics are not understood at this time, either within the EOSDIS system or in telecommunications in general, and so are not factored into any solutions.

3.3: Gigabit protocol issues

There are some relevant gigabit protocol issues that affect the design of the DMSS telecommunications system. These include protocol optimizations, rate control methods, and lightweight protocols.

Protocol optimizations are useful in the processing of stream data at gigabit rates; these include methods of header prediction in TCP and factoring frequent header cases out of the protocol stack. Other optimizations in the implementation of TCP have shown the operation of this protocol at rates near 400-700 Mbps, easily supporting both the satellite ingestion and user visualization components of this system.

Rate control methods provide processing adjustment to reduce queuing requirements, and reduce resulting jitter in the packet flow. 'Stop-and-go queuing', 'Leaky bucket', and 'Virtual Clock' are all similar methods for rate adjust-

ment. However, none are included in currently stable transport protocol implementations as yet.

Another protocol of interest, especially in the Data Ingestion operation, is the XTP protocol. XTP is a lightweight protocol that is designed to be implemented in VLSI hardware.

Other methods of achieving high performance protocols are not required here. These include other lightweight protocols, such as protocols that support fast RPC, or low latency transactions, or methods to reduce protocol complexity. The frequency of transactions in the DMSS system is not high enough to warrant these new protocols.

3.4: Suggestions for processing node scalability

One component of the DMSS system, the processing node architecture, is currently based on the VME/VSB bus interconnection [2]. A more flexible solution would use high-speed LAN interconnection methods, such as crossbar switches, for "backplane" communication among processing components.

One suggestion for possible research in this area would be the development of a VME/VSB virtual backplane, one which would permit the arbitrary interconnection of board-level components but exhibit a conventional VME/VSB interface. This crossbar-backplane would permit multiple FDDI interfaces per system aggregate, and thus multiple LAN loops among systems, permitting a more flexible implementation. This latter solution could be implemented incrementally, after the initial implementation phase of the design.

4: Stepwise-refinement

A first step toward the understanding of the communications structure is the stepwise refinement of the system design. These steps are based on the given NASA documentation, and general principles of system design.

The easiest implementation would be a LAN. The problem is that there are two load issues: direct query response load (computation and retrieval of actual data), and meta-data issues such as scheduling, load evaluation, and security access. Security issues are best solved by a physical partitioning of the network, with a coordinated set of controlled access points. The external access points comprise a set of nodes that interact through a separate meta-data LAN. Precomputed plans are sent through the controlled access point to the inner LAN for execution of the extraction.

Data ingestion occurs on the way into the inner data storage LAN, but does not use the outer security/scheduling LAN for access, since satellite-originated paths are presumed secured at the source. Further, the process of

ingestion need not alter the meta-data storage until archived inside the inner LAN.

The best way to understand these observations is to see their evolution and extraction from the existing characteristics of the DMSS system. Here we present a step-wise refinement of the telecommunications structure. We also present a description of the data flow and meta-data flows of the system, all to finally define the characteristics of the system sufficient to indicate a design.

4.1: Step 1 - the most general description

The most general description of the EOSDIS system is as an operational entity. Consider the DAAC as a 'black-box', with inputs and outputs [6] [7]. Inputs are comprised of the satellite data stream and the user queries. The output is the user visualization stream. The size of the arrows and lines is representative of the qualitative relative bandwidth requirements.

The input satellite data stream of 1 Terabit/day averages out to 11.5 Mbps (Figure 1). The user commands require negligible bandwidth, both because of their small textual content and their sporadic nature. The user visualization estimate is based on a 1000x1000 pixel display, changing at a rate of 24 frames/second (movie-quality video), at a depth of between 8 bits/pixel and 24 bits/pixel. This results in a session bandwidth of between approximately 200 and 600 Mbps, or 8-24 Mbits per frame.

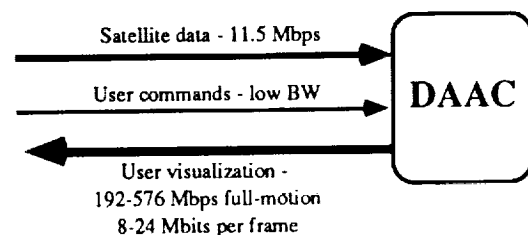


FIGURE 1. Step 1: System input/output

The input satellite stream thus requires a T-3 signal line (45 Mbps), assuming the 1 Terabit/day rate can be smoothed to per-second equivalent. The user commands can be accepted over conventional modem/dialup lines. The raw visualization stream requires SONET STS-12 rates, which are unlikely to be available for user deployment in the time-frame of this project. A lossless intraframe compression at these rates may be available, and would result in a 20-40 Mbps stream, which could be supported by FDDI LAN technology (100 Mbps). Lossy compression, such as JPEG, can further reduce this requirement to the Ethernet LAN realm at approximately 4-8 Mbps.

4.2: Step 2 - partition processing / data

The next step in this refinement involves the partitioning of the system component into data and processing components (Figure 2). The DAAC design is readily partitioned in this manner. This partition is modeled after the so-called 'Von-Neumann' computer architecture design.

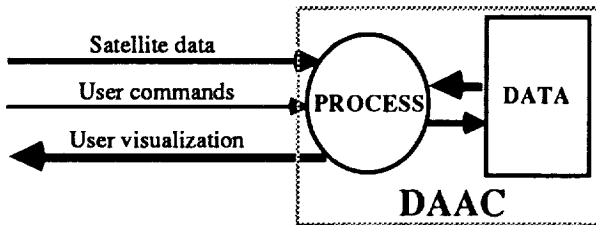


FIGURE 2. Step 2: Von Neumann decomposition

The processing requirements of this diagram are described in [12]. At this point, the processing and communications requirements are not sufficiently specified to determine the design, as was noted in the NASA analysis [12]. At this point, it is evident that this partitioning is not optimal, because the satellite data input stream and user visualization streams are largely independent, yet are processed in a single entity.

4.3: Step 3 - separate input / output streams

The next step in the refinement includes the description of the physical and algorithmic components. The processing is partitioned into ingestion, command processing, and extraction components (Figure 3). The ingestion portion occurs in specialized hardware [2]. The command processing translates user input into algorithms for extraction, which are executed in the extraction component [3]. By this diagram, the I/O intensive components are the ingestion and extraction [4], but there is substantial computation involved in the translations done by the command processing as well [8], [10].

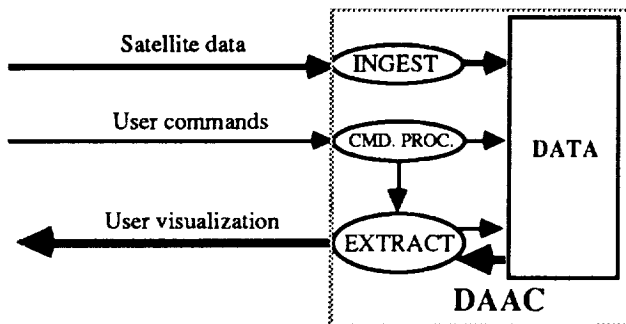


FIGURE 3. Step 3: Internal input/output streams

User commands therefore interact with the extraction process, but not the ingestion, which can be relegated to a separate component. Further, because the database is responding largely to command information (vs. data) from the command and extraction interfaces, the database might benefit from a partitioned internal structure, so that data input into an unorganized archive component can be isolated from the extraction access bandwidth requirements.

4.4: Step 4 - Separate into physical components

The final step in the refinement is the addition of partition information from the known implementation of existing components. The ingestion engine is known to be composed of a number of pipeline stages with separate control and pipeline communication paths [2], [9] (Figure 4). The database is also known to be composed of meta-data indicating the semantic modeling of the data structure, and an auxiliary processing element to monitor this modeling, in addition to the data itself [1].

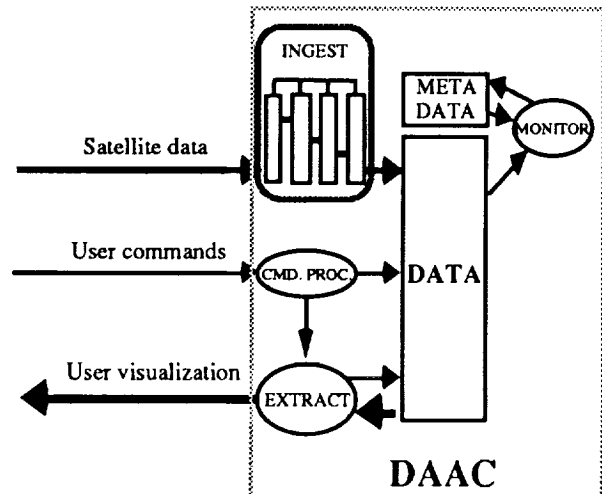


FIGURE 4. Step 4: Internal physical components (as already specified)

4.5: Step 5 - Replicate interior components

This final step in the refinement indicates that the user interface issues can be considered independent of the satellite data ingestion procedures. The decomposition does not yet indicate the systems issues involved, because only single users and satellite streams are indicated. We can augment the structure further by adding multiple copies of each entity, to denote how the replicated components interact.

We do not replicate the database component of the DAAC because we consider user requests within a single DAAC only. This is reasonable because the data of the DAACs is partitioned based on expected use and semantic content. Merging of data streams may occur, but is expected to be managed by the merging of independently delivered visualization streams from a number of independent DAAC sources.

Figure 5 denotes the interaction between the command processing elements in a scheduling capacity. The extraction processes are shown as independent, because determining overlapping computation is intractable and of little benefit given independent user control. The satellite processing streams are independent, but the number of pipeline stages is flexible and may be allocated from an aggregate rather than within dedicated system sets.

The decomposition shown here indicates the components of interaction and the bandwidth characteristics between them. It is also useful to view the data streams by semantic partitioning, in data flow and meta-data flow diagrams within the same structure.

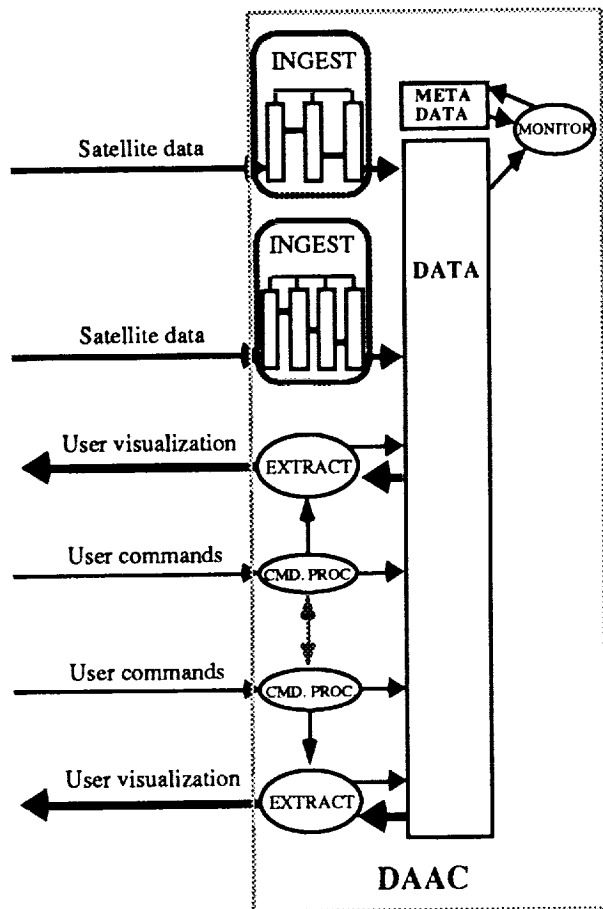


FIGURE 5. Step 4: Internal physical components (as already specified)

4.6: Data flow diagram

The data flow of the system can be described by two diagrams: one indicating the satellite data, the other indicating the archived data. The control operations specified by the user input are considered meta-data flows, shown later.

The satellite data flow consists of 1 Tbps streams pipelined through archival processing systems (Figure 6). If these systems are statically specified, the existing design of fixed-pipeline configuration will suffice [2]. If the satellite processing components are dynamically allocated, a network must be established among the elements. We assume here that these processing stages are largely static because of the data that would be lost during any reconfiguration. Thus, the satellite data flows represent fixed interconnections beyond the underlying dynamic network design. Further, scalability is provided by the addition of separate processing systems for additional satellite data streams in an independent fashion with linear cost. The streams can be compressed from the satellite to the pipeline processor, but the inter-process bandwidth requirements do not necessitate intermediate compression prior to storage.

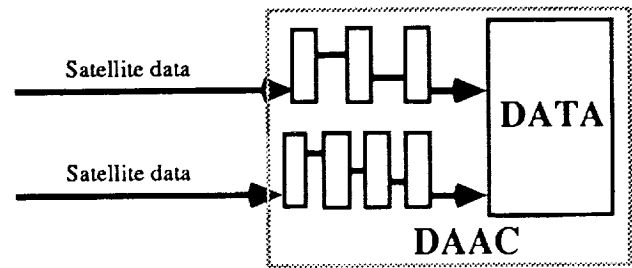


FIGURE 6. Satellite data flow (11.5 Mbps) - fixed interconnection

As a side-bar, we note that compression of archived data may have unanticipated effects on the communication load of the user visualization processing, as well as affecting retrieval. Extraction of particular information is complicated by stream encoding because it may require the decoding of a large section of data to locate a particular item, especially if the encoding destroys the key information. Effort should be made to avoid this if random access is required.

Further, a variable bit-rate encoding may cause fluctuating loads on the storage and extraction processes. While the storage process may be able to accommodate this fluctuation, the output data fluctuation will generate variable bit-rate streams to the extraction processors, which will require jitter control to permit stream merging. Recent research has also indicated that variable bit-rate streams

can cause interference effects in networks, even when the streams do not directly compete for resources.

The user visualization may require an arbitrary amount of pipelining, merging, and interleaving of extracted archive information (Figure 7). Whereas the individual streams are independent to permit independence in user control, the allocation of resources to the extraction processes is necessarily highly dynamic. The resources of extraction processes should be part of a dynamically reconfigurable network so additional resources can be added for additional functionality or scale of service.

Processing the streams usually occurs in the uncompressed domain, so any compression should occur at the final stage before user output. As a result, compression cannot be effectively used to reduce internal network load. The resulting interaction requires a very high bandwidth, very high connectivity network, such as BISDN (i.e., ATM).

If the user visualizations are restricted to conventional resolutions (500x500 at 1-8 bits, rather than 1000x1000 8-24 bits deep), the data streams are reduced from 200-600 Mbps to 6-50 Mbps, at full-motion 24 frames/second. While these streams cannot be accommodated in even a single Ethernet hop, a modified FDDI ring can be used.

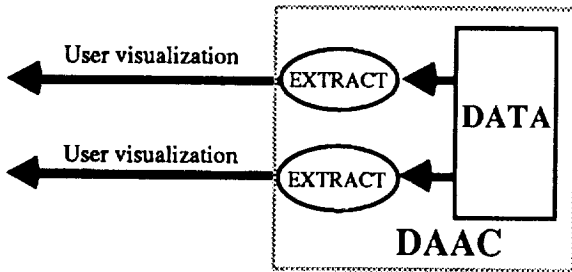


FIGURE 7. User visualization flow
(200-600 Mbps raw /
5-12 Mbps compressed)

Consider the dual-ring FDDI. Each level of the ring can accommodate 100 Mbps. There are some recent protocol systems which permit the utilization of multiple segments of the ring simultaneously; this would permit sequences of processors on the ring to be configured as a pipeline, and the output would be collected on the other ring. The result would permit redistribution and configuration of extraction processing resources within the ring.

If the visualization stream is not full-motion or full-color, the bandwidth required would be reduced even further. Also, it is not clear at this time whether the full bit-rate is required during extraction or is the result of data stream merging costs, the latter of which could be transmitted to the user in a repeating loop.

4.7: Meta-data flow diagram

The other flows denote the control streams. Some of these streams are user-specified control, and others are control between components of the system (Figure 8). These are not high-bandwidth paths that dominate the network design, but are communication paths which must be provided, at least transitively, by the interconnection topologies.

These streams include: interaction among the pipeline elements, monitoring of archive access for dynamic reorganization, user commands, extraction commands, database retrieval commands, and communication between the command processors for distributed resource allocation.

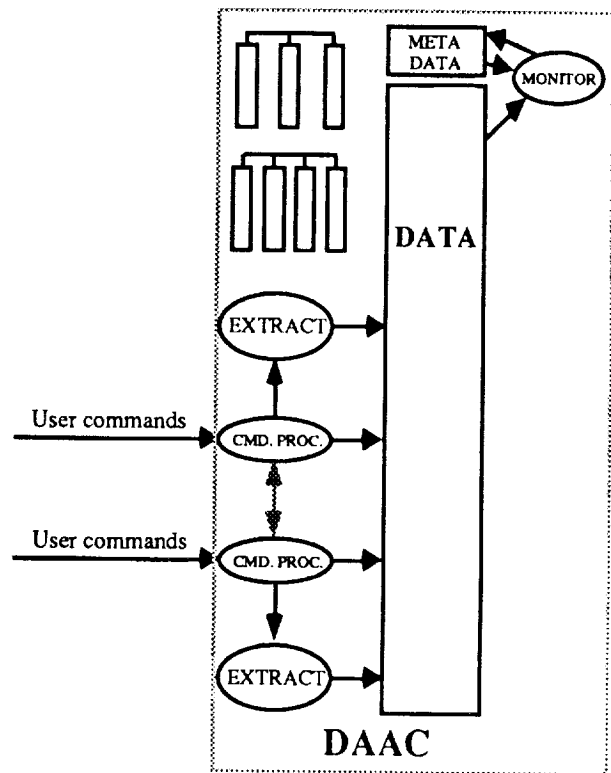


FIGURE 8. Meta-data (control, management) flow
(low bandwidth)

4.8: Indicated design

The following are the recommendations for the design of a network for a DAAC system that is flexible, scalable, and secure.

There is a multiple ring structure. The rings comprise the data input, query transformation, and data processing and output components of operation. By separating the structure thus, the satellite processing is partitioned from the user-level operations and the query processing is parti-

tioned from the internal extraction operations. The former provides a robust isolation between data input and output and the latter provides a similar isolation of user and data processing. The result is a robust and secure system.

The distribution of satellite processing resources in a high-speed ring (FDDI II) or BISDN network (ATM) provides enhanced pipelining capability, scalability, and dynamic reorganization of resources not afforded by the current, fixed interconnection within individual backplanes [2]. This requires the use of emerging FDDI II protocols supporting the simultaneous use of multiple ring segments, sometimes called 'multiple tokens'. This feature of the network design was emphasized by the meta-data description of the system.

The distribution of user query processing components among a low speed ring or bus (Ethernet, for small distances, or token ring for larger distances) provides links among the command processors to support distributed resource allocation at scheduling time [10], [8]. This interconnection was indicated by the step-wise refinement method.

The dynamic allocation of computation elements for extraction processing is similar to the satellite processing ring, i.e., both indicate an FDDI II modified multi-token ring or an ATM switching system (supporting full interconnections via a SONET-rate crossbar or multistage interconnection network). In the case of the extraction processing, the status of processors must be monitored by the query processing network for resource allocation. The idea is that the resources of the high-speed extraction network are allocated 'out-of-band', at scheduling-time, in the query processing network.

The monitoring of the database usage and structure can also occur within the query processing network, because it is a resource re-allocation function.

The result is a system that is composed of three networks: one isolated multi-token FDDI II network or ATM switching system for satellite data processing and a slow query processing network linked to another fast extraction processing network. Security is enforced in the slow query processing network by nature of its physical partitioning from the other two networks.

The general structure is visualized and instantiated with canonical networks in Figure 9. The basic description is as follows. A control console/host computer to each network is assumed.

The satellites are connected to SatNet with T-3 (45 Mbps) lines. The Pipe Processors are as described in the Ingestion portion, modified to provide a network interface, rather than a VME/VSB interface [8]. SatNet is either an FDDI II multi-token ring, or (optimally) an ATM BISDN network, providing full crosspoint interconnection with rates of STS-3 (155 Mbps) to STS-12 (620 Mbps). Until such technology is commercially available, a conventional

analog crossbar can be used, because the connections within this network are not frequently modified.

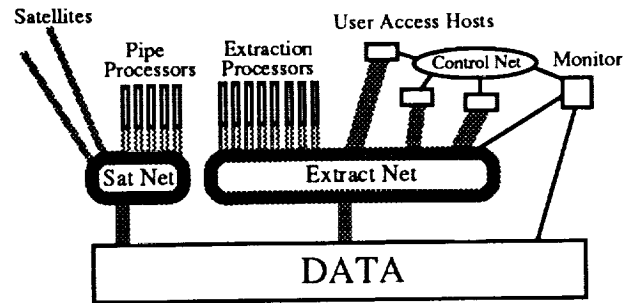


FIGURE 9. Generalized structure of telecommunications of EOSDIS DAAC

The same Pipe Processors can be used as extraction engines, with downloadable programs, or by workstations, as available. The design of ExtractNet supports heterogeneous systems, including supercomputers, workstations, and special-purpose Pipe Processors (as in SatNet).

The extraction processors can be connected to an ATM BISDN switch to implement the ExtractNet component. The ExtractNet should not be implemented with FDDI II or a crossbar, due to the highly dynamic reconfiguration that needs to occur to support varying user-specified extraction processes. The ExtractNet has a high bandwidth link to the database and another to each of the user-host processors. This latter link supports individual visualization streams.

The user access hosts are connected via a relatively conventional token ring, such as FDDI, or even Ethernet. Commands and resource allocation are processed on this network; these are low-bandwidth activities. It is assumed that one user-host will support each user connection because of the bandwidth required per user connection for high quality full-motion video. If still video is used, multiple users can be supported per station.

The ControlNet is used for distributed resource allocation among the user hosts and out-of-band resource allocation of the components of ExtractNet. SatNet allocation can occur off-line because the network is reconfigured only periodically.

A separate monitor host performs low bandwidth computations, such as database restructuring for performance [1]. The design of the database to support dual high-bandwidth ports, or possibly multiple high-bandwidth retrieval ports to ExtractNet, is beyond the scope of this section.

User access is restricted to the ControlNet, where queries are processed within the hosts, or possibly off-loaded into the Extraction Processors of ExtractNet, or a separate high-performance engine connected to ControlNet (like

the monitor). The access control is both physical and logical, so that the user commands are prohibited from utilizing the ExtractNet or SatNet. The interaction is similar to that of RPC, where user commands are decomposed into fixed, preexisting procedures that are pipelined together. User access is as secure as in RPC.

5: Conclusions

Here we have presented an architecture for the DMSS of the IDM DAACs developed by stepwise refinement. We have discussed how existing protocols are sufficient for use in this architecture to support both data ingestion and data fusion and visualization.

The DMSS architecture presented is scalable, partitions the DMSS via gateway access servers, and includes internally replicated processing components. We have also shown a design in which control is distinct from data streams, both logically and topologically.

The architecture we show permits various implementations:

- gateway as authenticator only,
 remainder as centralized server.
- gateway as delegator
 using vector pipelined REV processors.
- gateway as authenticator
 using REV on workstations.

It is this latter approach we feel is most general, scalable, and useful for the architecture of the DMSS IDM DAACs.

6: Acknowledgments

We would like to thank the members of the NASA GSFC Information Science and Technology Office (ISTO). - Code 930.1, notably Nick Short, as well as Bob Crompt, and Bill Campbell, for their input into this work. We would also like to thank Jim Chesney of NASA for providing detailed information on the DMSS ingestion hardware. We also thank the members of USC/ISI HPC Division for their constructive comments on this document.

7: References

- [1] Campbell, W., and Short, Jr., N. "Using Semantic Data Modeling Techniques to Organize an Object-Oriented Database for Extending the Mass Storage Model." International Astronautical Federation 42nd Congress, Montreal, October 1991.
- [2] Chesney, J., Speciale, N., Horner, W., and Sabia, S. "High Performance VLSI Telemetry Data Systems." Amer. Inst. Aeronautics and Astronautics, Second Int'l. Symp. on Space Information Systems, Pasadena CA, September 1990.
- [3] Crompt, R. "Automated Extraction of Metadata from Remotely Sensed Satellite Imagery." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991.
- [4] Crompt, R., Campbell, W., and Short, Jr., N. "An Intelligent Information Fusion System for Handling the Archiving and Querying of Terabyte-Sized Spatial Databases." International Space Year Conference on Earth and Space Science Information Systems, Pasadena CA, February 1992.
- [5] McDonald, K. "Information Management Challenges of the EOS Data and Information System." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991, pp. 258-267.
- [6] Quirk, B., and Thompson, R. "Early-EOS Activities at the Land Processes Distributed Active Archive Center (DAAC)." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991, pp. 339-351.
- [7] Ramapriyan, H., and McConaughy, G. "Version 0 EOSDIS - An Overview." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991, pp. 352-362.
- [8] Short, Jr., N., and Shastri, L. "The Application of Connectionism to Query Planning/Scheduling in Intelligent User Interfaces." *Telematics and Informatics*, Vol. 7, Nos. 3/4, 1990, pp. 209-220.
- [9] Shi, J., and Grebowsky, G. "A Performance Model for Real-time Packet Processing." Mission Operations and Data Systems Directorate paper, NASA GSFC, 1990.
- [10] Short, Jr., N. "A Real-Time Expert System and Neural Network for the Classification of Remotely Sensed Data." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991, pp. 406-418.
- [11] Tanenbaum, Andrew S. *Computer Networks*. Second Ed., New Jersey: Prentice-Hall, NJ, 1988.
- [12] Wharton, S., Chang, H., and Krupp, B. "Sizing the Science Data Processing Requirements for EOS." *Technical Papers ACSM-ASPRS Annual Convention, Vol. 3 - Remote Sensing*, 1991, pp. 478-487.

1
2
3
4

1
2