M0128

Some Observations about Decentralization of File Systems

by

Jerome H. Saltzer

Summary

This short note takes the position that the inherent complexity of a decentralized and a centralized information storage system are by nature essentially the same. The complexity is the inevitable consequence of the objectives of the information storage system: information protection, sharing, privacy, accessibility, reliability, capacity, and so on. What is needed in both cases is a methodical engineering basis for designing an information storage system to specification. On the other hand a decentralized organization offers the potential for both more administrative flexibility, and also more administrative chaos.

This note is a summary of a talk presented at the IEEE International Computer Technical Conference held in Boston, Mass., September 22-24, 1971.

May 3, 1971

Some Observations about Decentralization of File Systems

by

Jerome H. Saltzer*

In recent years it has become more and more evident that while the purpose of computers may be to compute things, what most computers actually are used for is to remember things. We observe that designers propose ever more sophisticated storage hardware, supervisors, access methods, file systems, information retrieval packages, and networks, with the intent of improving abilities to develop and use information storage. Thus, attention no longer exclusively concentrates on making information processing units ever faster; concern now extends toward such issues as performance of multilevel memory systems, protection of information integrity and privacy, and continuous availability of access paths to the information base.

When one begins to inquire about a possible methodical basis for engineering of systems which remember things, he exposes a variety of structural issues. An issue commonly raised is to what extent a centralized file system has structural or other advantages or disadvantages when compared with a decentralized structure. Two images usually fill out this picture of debate: on the one hand is a monolithic operating system based on a single large computer with a repertoire of storage hardware devices;

* Massachusetts Institute of Technology, Department of Electrical Engineering and Project MAC.

on the other hand is a network of potentially independent and self-controlled computer systems linked together with communication lines which permit access to files stored in any network member's computer system. The thesis which we wish to pursue is that those two images, and the issue itself, represent an inappropriate line of division. We claim that, except for the effect on administrative problems, the image of a network actually contains no fundamental structural properties not also present in the image of a centralized operating system. There may be differences in relative importance of certain properties: these differences are of degree, not of kind. The consequence of this claim, if substantiated, is that the technical choice between a centralized or decentralized organization is one of engineering tradeoffs pertaining to maintainability, economics, equipment available, and the problem being solved, rather than one of functional properties or fundamental differences in complexity. We will pursue this claim by example.

The usual line of analysis of the properties of a network of linked computer systems proceeds from the assumption that one has a number of independent computer sites. He then adds links, and inquires about the new properties thereby acquired, and the new problems thereby encountered. This line roughly describes the experimental approach being taken in the construction of the ARPA network [1]. We may obtain some more effective insight, however, by inverting the line of analysis as follows: starting with a centralized system, what happens as one gradually decentralizes it? This line allows us to examine carefully and exactly when a new capability is added, a new difficulty encountered, or an old one conquered.

Suppose we have a centralized, monolithic^ hardware and software system which contains in one place enough storage and enough computing capacity to serve a community of users who are not centralized. Next, suppose that one or more of the information storage devices -- say a moving head disk -- is removed from the central installation, and placed at a distance. Although a large number of engineering details may be involved, the only significant change that the central system can notice is that the storage device has had its performance degraded. The bandwidth of the channel to the device is limited to that of available, economic telecommunication facilities, and the access time of the device is increased by some small multiple of the time required to propagate a signal to the remote location. If the distance involved is 1000 milea, the access time increase is perhaps 20 to 40 milliseconds, a change in degree but not in kind from the usual 20 to 50 millisecond access time to a disk.

If the remote storage device is under the physical control of one of the users, and has a "detachable" recording medium, the central system must be prepared for the additional complexity involved in keeping track of information on off-line disk packs: but that capability was already potentially needed in the central facility itself, to provide rapid reconfiguration and cheap long term but low usage storage. Here we have a good example in which decentralization may make a certain capability almost mandatory, while the centralized system may be able to finesse the situation in some cases. This observation is really the basis for the claim that a decentralized system does not introduce new issues so much as it focuses a spotlight on issues which were present, though perhaps being ignored, in the centralixed system.

An even better example of this phenomenon occurs when we decide to decentralize the computing capacity of the system by placing a central processor at a remote site. Suppose we first restrict ourselves by placing only processing capability and no storage -- not even a little bit of directly addressable primary memory -- at the remote site. In that case, the remote processor must depend on the central facility for all storage, and we have nothing more than a multiprocessor computer system, which was probably already a concern of the central facility, for reasons of reliability, and ease of changing installation size in response to demand. But of course a central processing unit is accustomed to making random accesses to a directly addressable storage device every microsecond or so, and a 20 millisecond transmission delay will slow the instruction processing rate down by several orders of magnitude. Thus, this particular form of decentralization is probably not practical, and at least some rapid access storage is going to have to be located at the remote site so that it can make local copies of frequently used information. And now, the overall system complexity takes a step upward as we begin to worry about the systematic management of multiple copies of potentially shared information.

Thus, physical separation of the processors of a multiprocessor computer system from their community storage facilities seems to force some new complexity into the situation. On the other hand, are we sure that only networks of computers exhibit physical separation of processors and storage? Modern processors operate at cycle times measured in tens of nanoseconds, which means that they become significantly slowed down when waiting for signals to propagate only a few feet. Thus it is becoming common to place in the processor itself a local, or buffer memory [2], If a centralized system operates more than one of these modern sophisticated processors, it also has the problem of systematic management of multiple copies of information.

Again we have an example in which a problem which must be faced in the decentralized system exists, at least latently, in the centralized system also. From an engineering point of view in both cases one needs a way of modeling patterns of access to shared information, so that he can then invent algorithms and evaluate strategies for having the information on the right storage device at the right time. The decision about centralized versus decentralized organization may thus turn out to be analogous to to that of a telephone network: to the extent that most calls are local, decentralization may be more effective. Unfortunately, the methodical basis for such engineering decisions is in its infancy.

Although we have argued that the intrinsic complexity of centralized and decentralized storage systems is similar, this is not to say that there are no differences of significance. As mentioned earlier, a decentralized system opens opportunities for additional flexibility in administrative control of individual network locations. Of course, this flexibility is a double-edged sword. On the one hand a site which is administered independently can set its own policies with respect to prices, reserve capacity, or scheduling algorithms. On the other hand, independent administrators can also develop a hodgepodge of differing operational hours, security measures, accounting controls, configuration differences, and operational methods, to the point that it is simply too much effort to try to use information at another site. Put another way, to the extent that a network of independently administered sites becomes more and more closely coupled to facilitate sharing of information, it must also observe a more and more centrally imposed administrative discipline.

In summary, then, we have taken the position that the inherent complexity of a decentralized and a centralized information storage system are by nature essentially the same. The complexity is the inevitable consequence of the objectives of the information storage system: information protection, sharing, privacy, accessibility, reliability, capacity, and so on. What is needed in both cases is a methodical engineering basis for designing an information storage system to specification. On the other hand a decentralized organization offers the potential for both more administrative flexibility, and also more administrative chaos,

References

[1] Roberts, L.G., and Wessler, B.D., “Computer Network Development to Achieve Resource Sharing," AFIPS Conf. Proc. 36 (1970 S.J.C.C.) pp. 543-549.

[2] Conti, C.J., Gibson, D.H., and Pitkowsky, S.H., "Structural Aspects of the System/360 Model 85," IBM Systems Journal 7, 1 (1968) pp. 2-21.

Work reported herein was supported by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract N00014-70-A-0362-0001. Reproduction is permitted for any purpose of the United States Government.