Introduction to the Storage Systems Program

John Wilkes, Richard Golding
Tim Sullivan and Carl Staelin

Computer Systems Laboratory
Hewlett-Packard Laboratories

August 10, 1995

HEWLETT PACKARD

Storage Systems Program
Mission

To make Hewlett-Packard storage systems:

✔ fast
✔ easy to manage
✔ profitable

HEWLETT PACKARD

Commentary
Mission

Our emphasis is on storage **systems** as a whole, rather than (say) on improving the performance of particular storage devices.

- **fast** is a necessary property in a world where competitive storage devices are commodity items: it's a "must have";
- **easy to manage** is our main thrust; you can also view this as a placeholder for things like highly available, robust, scalable, etc.;
- **profitable** implies moving further up the value chain, away from the cut-throat manufacturing competition in disk devices. HP AutoRAID is one example of this—our intent is to continue this trend, and help take HP further down this path.

Storage Systems Program
Underlying trends

Much more storage

❑ huge capacity increases

❑ increasing storage-device heterogeneity

Fast network connections to storage devices

Non-traditional data types as well as legacy data

❑ continuous media, images, hyper-documents

Ubiquitous data shared across many applications

❑ data accessible anywhere, any time

HEWLETT PACKARD

Commentary
Underlying trends

Storage capacity increases are driven by:

- existing improvements in device capacity (~60%/annum; current trends suggest that the numbers of drives bought will rise only slowly to meet legacy-data needs)
- new data types, including enormously more storage-intensive ones like continuous media (audio, video) and images—these will drive new capacity requirements and access properties
- integration of legacy data (e.g. text, dbms) with new types in hyper-documents (cf. the WWW)

Ubiquitous, universal data access will become the norm: people will expect to have access to the data they need wherever they are. Given that businesses will stand or fall on the availability of their data, the level of reliability expected will be that of the utility functions (power, water, phones).

Storage Systems Program
High-speed-network-attached storage devices

✔ Frees clients from I/O-scale limitations

- Pentium server can access same amount of storage as a T500

✔ Performance >= locally-attached

- remote devices are just as fast as local

✔ Network, not a bus

- devices shareable
- “scaling for free” from network switches
- data accessible from anywhere on the network

✔ Deployment: imminent!

- FibreChannel first (late ‘95); ATM next?
- can use a range of cost-effective technologies

HEWLETT PACKARD

Commentary
High-speed-network-attached storage devices

The change to network-attached storage will be revolutionary, not evolutionary. The biggest benefits come from lifting the connectivity and proximity limitations of SCSI:

- The need for a T500 simply to reach data will go away: a cheap, PCI-based 1Gb/s spigot will give even a lowly Pentium access to the entire storage system.
- Access to remote devices will be faster than to local ones, reversing a trend of the 1980s.
- FibreChannel networks will provide scalable networks capable of connecting Petabytes worth of devices to hundreds of clients.

We have a chance to capitalise on it if we act now—the early deployments (FibreChannel) will be happening in late 1995. Even when/if FibreChannel is replaced by ATM, the benefits will remain.

Storage Systems Program

Goal

Grand Unified Theosophical Storage:

- ✔ network-attached, smart storage devices
- ✔ department- to enterprise-scale systems
- ✔ client — service — data-utility model

Commentary

Goal

Our proposal is to capitalise on the trends we see, and develop a **new storage architecture** that is capable of meeting the demanding needs of late-1990s information systems and their clients:

- direct network-attach for storage to get connectivity and speed;
- smart storage devices to exploit low-level device-specific knowledge, ease management problems, and simplify building high-value services on top of commodity parts

The initial target will be department- to enterprise-scale systems (e.g., HP-UX, clustered NT) because they will be most in need of (and receptive to) our solutions. However, we plan to aggressively "trickle down" the technology to smaller-scale systems—both for economies of scale, and to exploit the large market they represent.

We envision the future as having a new three-tier data-access model to replace the traditional client-server one: the next slides describe this.

Storage Systems Program

Grand Unified Theosophical Storage

Theosophy, ***n.***

> Any of various philosophies professing to achieve a knowledge of God by spiritual ecstasy, direct intuition, or special relationships, especially a modern movement following Hindu and Buddhist teachings and seeking universal brotherhood.
>
> — **Concise Oxford Dictionary**

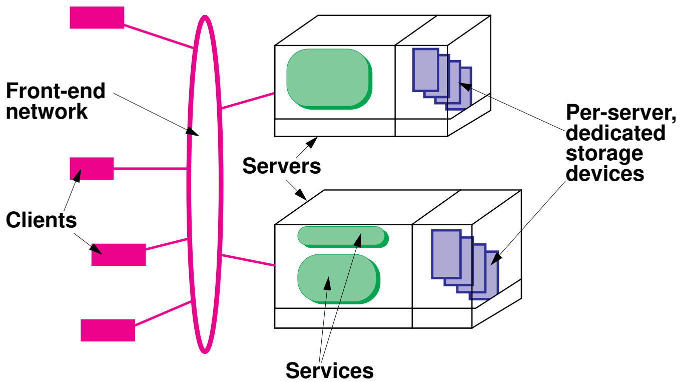
Commentary

Grand Unified Theosophical Storage

This is, of course, the guts of our proposal …

Storage Systems Program

The state of the world today



HEWLETT PACKARD

Commentary
The state of the world today

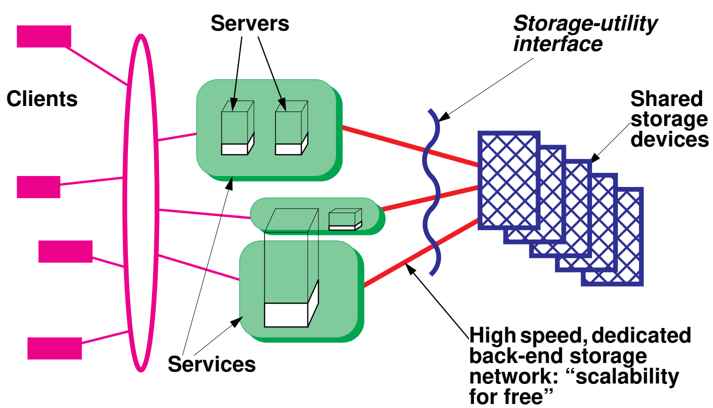
Clients talk Ethernet today, 100VG tomorrow; they do data-capture, user interface, and local data analysis (e.g., spreadsheets). They range from portable PDAs through PCs to UNIX workstations.

Clients talk to **servers** to get the information they operate on. These servers are typically large, dedicated systems that run applications to extract information from a formal or ad hoc database. Drawbacks include:

- few services are redundant;, so information availability is only good-to-moderate;
- adding a new application requires finding a server with spare capacity **and** a copy of the needed data; this may mean adding a whole new (large) server, and its own data copy;
- management is a nightmare: every system is different.

Storage Systems Program

Grand Unified Theosophical Storage (1)



HEWLETT PACKARD

Commentary
Grand Unified Theosophical Storage

In the first step towards the new world, decentralized information-access **services** take over from simple servers, to provide functions like data mining, indexing, and decision-support. The services will be fully redundant so that they can tolerate failures of individual servers.

This is made possible by putting the data accessed by the services into a **common, shared pool of network-attached storage devices**. (Remember: getting to these will be at least as fast as to local devices.)

These devices will operate as a scalable **storage utility**—with all the robustness and ubiquity this implies. Adding a new service simply means finding a server machine with spare CPU capacity.

To allow effective sharing and resource management, the storage-utility interface level needs to be higher than block READ and WRITE. This is where we have an opportunity to add value behind the storage-utility interface, and raise the profitability of storage components.

Storage Systems Program

System vision: beyond client-server computing

Clients: information presentation

- ❏ user interface, data capture, analysis

Services: information access and generation

- ❏ scalable systems; data mining; decision support
- ❏ "extracting information out of data"

Storage-utility: data access and storage

- ❏ common, shared, storage pool
- ❏ performance, availability, ease of management
- ❏ high-performance network-connected devices

Commentary

System vision: beyond client-server computing

A key distinction we are making is between **data** and **information**: the storage utility stores and provides raw data; the information services convert that into something useful. This separation avoids moving the entire information-access problem down into the storage subsystem.

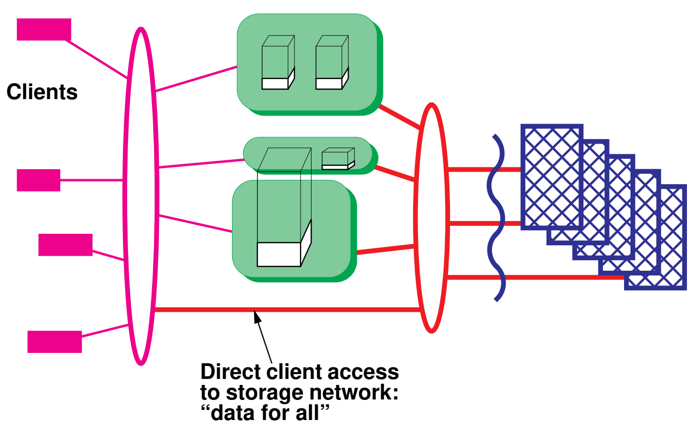
The capabilities of the storage utility subsystem must include:

- scalable capacity and performance, including incremental on-line upgrades;
- bulletproof data availability;
- greatly improved manageability and ease of use.

Getting the best performance and robustness from a scalable system of this size and complexity will have to be automated—it cannot be imposed on people.

Storage Systems Program

Grand Unified Theosophical Storage (2)



Commentary

Grand Unified Theosophical Storage

In the next step towards the new world, clients will be given direct access to the storage utility functions, bypassing the information services if they wish, and allowing them to do their own information mining.

The main additional implication here is a need for additional security. Before, the (trusted?) information services could be used to restrict improper access; now, the storage utility has to do it itself.

Storage Systems Program

GUTS goals

✔ Fast

✔ Easy to manage

✔ Profitable

Commentary

GUTS goals

Our goals here are just the same as they were for the overall Storage Systems Program (and as they have been for the Storage Systems Project for the last 2 years).

The next few slides detail the implications of these goals in the context of our proposal.

Storage Systems Program

GUTS goals

Fast:

✔ high throughput — including continuous media guarantees

✔ scalable across wide capacity range (GB to TB) and performance needs (100:1)

✔ application-aware adaptations for performance

Commentary

GUTS goals

The performance goals require:

- high aggregate throughput (with low latency, of course!)
- guaranteed delivery for time-sensitive data like continuous media (this is not optional—"best effort" solutions will simply not be adequate)
- these capacity and performance requirements mean that the system will eventually have to scale to hundreds of storage devices
- physics still limits the rate at which storage-device performance increases compared to processors—fundamentally, this can only be addressed by exploiting application-specific information, so that the storage system can anticipate and plan for future accesses.

Storage Systems Program
GUTS goals

Easy to manage:

✔ self-healing: fault tolerant, highly available services

✔ self-managing: express policies, not mechanisms

✔ self-configuring: adjusts to workload; predicts capacity/performance bottlenecks

HEWLETT PACKARD

Commentary
GUTS goals

The manageability goals require:

- **self-healing** systems that do not need people in the critical path for of intervening to correct faults and fix failures.
- **self-managing** systems that take as input declarative specifications of what the system is trying to achieve, not how it should be done (e.g., "the payroll data needs to be utterly available; it is updated infrequently, although read often", rather than "make two copies of these files, spread across those three disk arrays, and configure them with RAID 5 data layouts and a stripe size of 64KB")
- **self-configuring** systems that adapt to the actual workload generated by the system's clients, that can isolate current performance bottlenecks, and that can predict future ones before they become a problem—e.g., to enable sale-or-return solutions.

Storage Systems Program
GUTS goals

Profitable:

✔ superior value (ease of use, flexibility, scalability)

⋗ premium price

✔ commodity cost structures (exploiting standard parts)

⋗ increased profitability

HEWLETT PACKARD

Commentary
GUTS goals

Our profitability model is that driving ISG's recent moves into higher-value components like the HP AutoRAID array:

- higher customer value means premium price
- commodity cost structures means higher profit

We'd like to emphasize the latter: we believe completely in using off-the-shelf, high-volume parts in every place we can. This means that we are only interested in innovating (and driving a new standard) where we can make a significant value-added contribution to HP's storage systems business. (Note that this may still mean high-risk investments, and pre-standardization work on new interfaces in industry groups like NSIC, the National Storage Industry Consortium.)

Storage Systems Program

Key research objectives

Remove technical obstacles to:

- ❏ attribute-managed storage systems
- ❏ mixed continuous-media and traditional data
- ❏ scalable, distributed storage system

Commentary

Key research objectives

We have prioritized what's required to make all this real, and extracted three focus areas for us to work on at HP Labs:

1. Attribute-managed storage is how we propose to solve the manageability problem (more on it below).

2. We believe that dedicated, specialized solutions like video-servers will fulfill short-term and/or niche market needs, and that the general-purpose solution of allowing arbitrary mixing and matching of storage demands to resources will win the day. This objective is to provide cost-effective continuous media solutions in a general-purpose framework.

3. Getting to a scalable, fault-tolerant system with hundreds of nodes remains a daunting technical challenge, even if we subset the problem to the domain of storage systems.

The next slides describe each of these in more detail.

Storage Systems Program

Key research objectives

Attribute-managed storage systems

- ❏ Characterizations (especially performance and availability)
  - storage devices
  - storage utilities (devices + service layer)
  - storage customers: applications, services
- ❏ Mapping function
  - capacity planning
  - performance, availability, cost, …
  - inter- and intra-device data placement

Commentary

Key research objectives

Attribute-managed storage is our solution to the manageability problem. The basic idea is to have clients to specify *what* they need, not *how* it is to be provided.

The technical problems fall into the following main parts:

- how to *specify what customers need* (services, applications);
- how to *specify what devices can deliver* (including when front-ended by different pieces of software, such as disk drivers);
- the *mapping problem:* how to map customer needs onto device capabilities; this includes where and how to place data, taking into account performance, availability, cost, etc.;
- using trend analysis and device capability information to identify and predict system bottlenecks before they occur.

Storage Systems Program

Key research objectives

Mixed continuous-media and traditional data

❑ resource reservations and guarantees

❑ mixed continuous media and traditional data on the same devices

❑ data layouts for improved device utilization/efficiency [note: we plan to lift these, rather than develop them]

Commentary

Key research objectives

Providing guaranteed performance for continuous media is itself a hard problem. Fortunately, we can first solve it in a narrow domain: the storage utility.

Fundamentally, we need to use results from the real-time community to provide predictable behavior. Challenges include finding ways to:

- *specify* resource reservations without requiring too much prior knowledge from system- and application-builders.
- *grant* resources to data-movement, rather than processes
- *mix* guaranteed accesses and regular ones on the same devices
- while *optimizing* data placement and scheduling for efficiency.

It is our hope that we can leverage much of the low-level portions of this work, rather than inventing all new stuff, but the composition problem is still a complex one.

Storage Systems Program

Key research objectives

Scalable, distributed storage system

❑ metadata management

- locking, naming, data-finding

❑ distributed-resource optimization

- RAM, disk spindles, other devices
- load balancing

❑ caching, replication, redundancy

Commentary

Key research objectives

Controlling a hundred-node computer system is a daunting task. Fortunately, there's a huge amount of prior work in this area, and the main task is to choose between the options, rather develop new solutions. We've chosen to split the problem into:

- metadata management: metadata is internal information about where data is stored and its state (naming, location, and locking)—separating it from the distributed data management proper will be a major simplification, we believe;
- exploiting the pool of storage devices as a single resource, across which transparent load-balancing can be done;
- exploiting caching and redundancy (including replication) to meet performance and availability goals.

Storage Systems Program

Research partners and idea sources

❑ Universities

- CMU/Garth Gibson and the Parallel Data Lab (also NSIC?): direct network-connected smart storage devices
- UC Berkeley/the NOW project and Univ. of Washington/Hank Levy: distributed metadata, and RAM-cache management
- University of Arizona: Scout OS

❑ IEEE Mass Storage Systems reference model

- Virtual Storage Layer: an example of attribute-managed storage
- ISG SSSI storage architecture: a refinement of the IEEE MSS model

❑ HP Labs

- DataMesh; VOD project-Alfred hardware; Future Systems Dept. performance modelling; Hamlyn interconnect protocols

Commentary

Research partners and idea sources

The intent here is to list some key partners for our research agenda's success. The idea you should get is that we plan to leverage ideas from elsewhere wherever we can, and only innovate where we have to! Here's a couple of highlights:

- Garth Gibson at CMU has (independently) proposed to do almost exactly what we are advocating; IBM, DEC, and Seagate have already expressed strong interest through NSIC. Our hope is that this will enable us to collaborate on pre-competitive, pre-standardization research.
- The IEEE MSS Virtual Storage layer is close to what we are proposing as the first-cut interface to the storage utility pool; the SSSI storage architecture is a refinement that we developed on behalf of ISG.
- Our prior HPL research work in these areas means that we are as well positioned to succeed as any research project in the world, if we can only fix our resource crunch.