



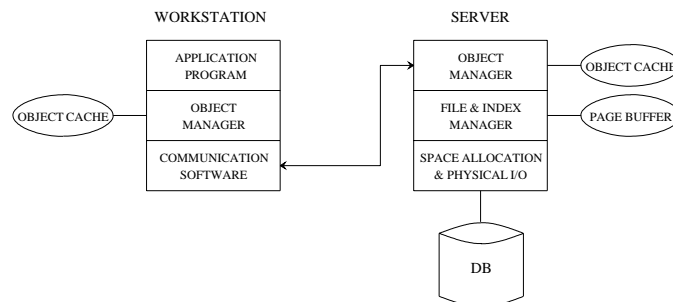
David DeWitt, et al

## A Study of Three Workstation-Server Architectures for Object Oriented Database Systems

Presenter **Konstantin Beznosov**

October 23, 1996

## Object Server: Architecture



- Server understands the concept of an object.
- Methods can be applied to objects on either the client or the server site.

# Object Server: Pros & Cons

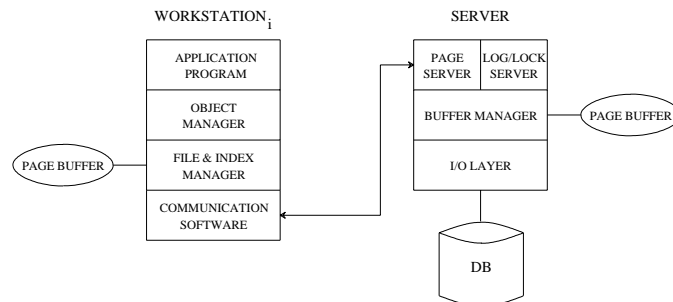
## Advantages

- Methods applicable everywhere
- Centralized concurrency control
- Cheap object-based constraints

## Disadvantages

- Complex objects require multiple expensive RPC calls
- Complicated server design to make it run arbitrary methods
- Expensive cache inconsistency check is required before every method
- References to multi-page object members are very expensive
- Each retrieved object “goes” through multiple buffers
- Overall design requires most of system CPU power to be on the server

# Page Server



- The unit of transfer is a disk page.
- Server does not understand semantics of objects.

## Page Server: Pros & Cons

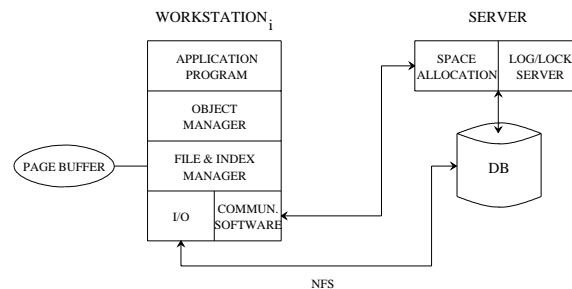
### Advantages

- Most of complexity on the client site
- Overhead of the server work is minimized
- More scalable than object-server approach
- Automatic pre-fetching of other objects if data exhibits space locality

### Disadvantages

- Methods can be evaluated only on client site
- Difficult to implement object-level locking
- Performance can significantly degrade if the clustering mechanism is not effective

## File Server: Architecture



- Features of page server.
- Concurrency control.
- Recovery.

## File Server: Pros & Cons

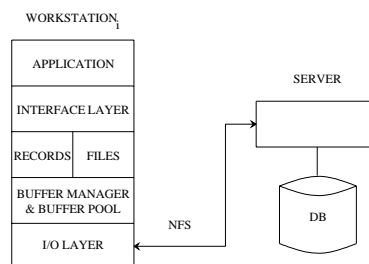
### Advantages

- Most of page-server advantages
- User-level context switches can be avoided during read or write operations
- Relies on remote file service technology which continues to evolve and be improved

### Disadvantages

- Same problems as page-server has
- NFS writes are very slow
- Page lock requests are expensive
- New disk page requests are expensive

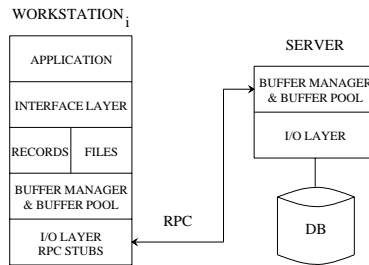
## File Server: Prototype



Remote File Server Design

- WiSS is run on one processor with its disk mounted via NFS

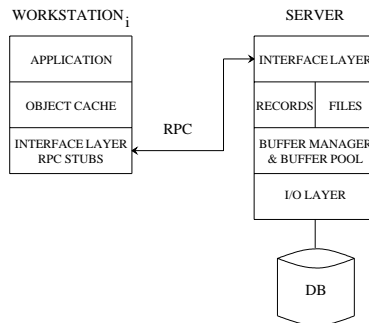
# Page Server: Prototype



Page-Server Design

Only I/O layer resides on the server, and the client has the rest

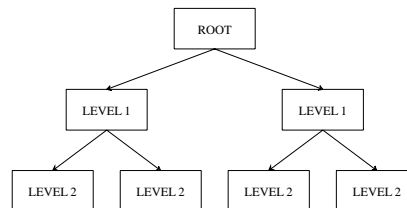
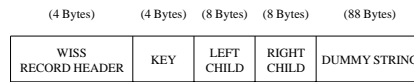
# Object Server: Prototype



Object\_Server Design

- Server contains all WiSS layers.
- Client has only object cache.

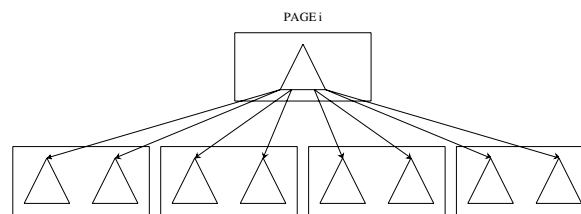
# Benchmark Database: Complex Objects



Complex Object Organization

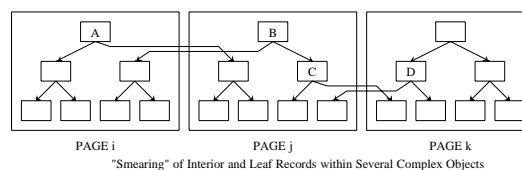
- Each object is composed of 7 records.
- Each record is 112 bytes long.
- Records are organized in the form of binary tree of depth 2.

# Benchmark Database: Parameters



A Complex Object with Its 8 Component Objects  
(Clustered in a 5-Page Region)

- Clustering Region size
- Clustering factor



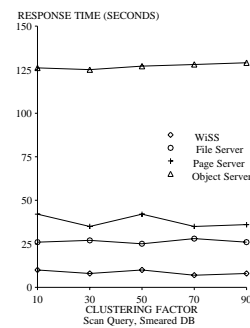
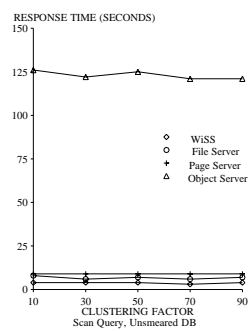
"Smearing" of Interior and Leaf Records within Several Complex Objects

- Smearing factor

## Benchmark Phases

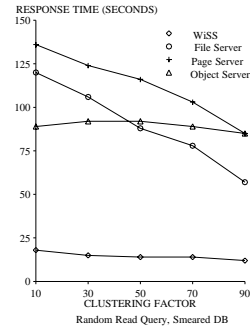
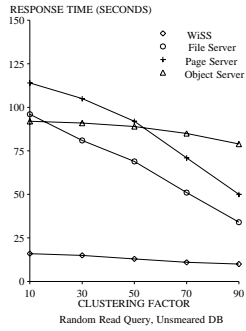
- I. Build five identical sets A, B, ... E
- II. Sequential Scan query on sets A through E
- III. Random Read query
- IV. Random Update query
- V. Sequential Scan, Random Read, and Random Update queries on the same set

## Clustering and Smearing: Scan



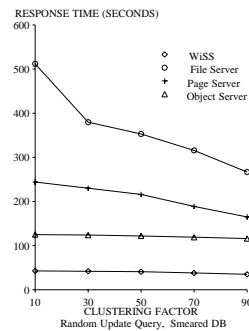
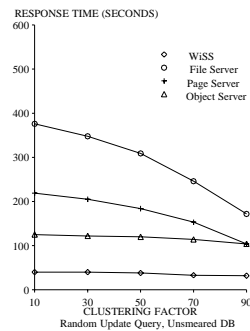
- No design is affected by clustering
- Extremely poor performance of **object server** for unsmearred database
- Substantial degradation of **page** and **file servers** with smeared data

# Clustering and Smearing: Random Read



- **Page and file servers** are very sensitive to the database clustering
- **Object server** performance is independent of smearing and clustering

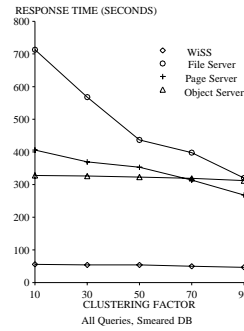
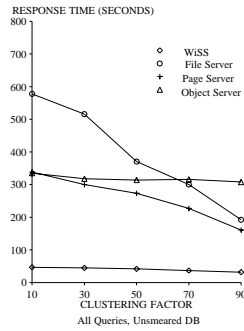
# Clustering and Smearing: Random Update



- **File server** performance is very bad because of stateless Sun NFS protocol
- **Object server** performance is very good and is independent of smearing and clustering

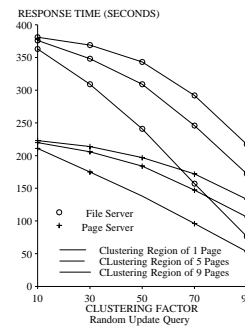
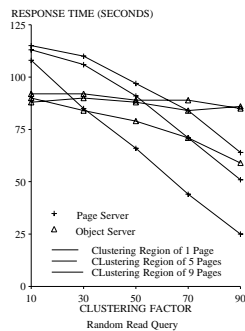


# Clustering and Smearing: All Queries



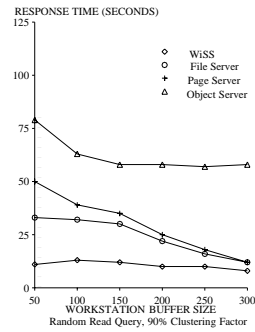
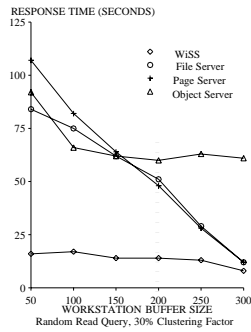
- **Object server** is not affected by clustering or smearing
- **Page and file server** designs can outperform **object server** only when objects are highly clustered

# Sensitivity To The Clustering Region Size



- All architectures are sensitive to the size of the clustering region
- **Page server** will do the best if the clustering region is relatively small or the clustering factor is high

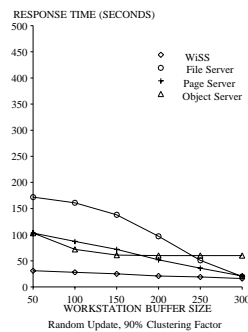
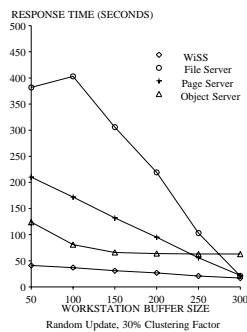
# Client Cache Size: Random Read



- **File and page servers** improve with the size of the buffer pool up to the size of the database
- RPC costs become decisive factor in **object server** architecture when cache becomes half as large as the database

*Do "Large buffer sizes do a better job at compensating for the lack of effective clustering than the opposite" ?*

# Client Cache Size: Random Update



- **Page and file servers** are very sensitive to the size of the client cache.
- For **page and file servers**, more memory is not a reasonable substitute for effective clustering.

## Conclusions

- No “gold bullet”
- **Page server** is better in case of high clustering and large enough buffers
- **Object server** is no good for “scanning” applications but it is the choice for limited memory clients or big database and for unclustered or smeared data
- A hybrid architecture may be necessary to maximize overall performance

Copy of this presentation slides  
<http://www.cs.fiu.edu/~beznosov/COP6545/presentation-workstation-server.ps.gz>