

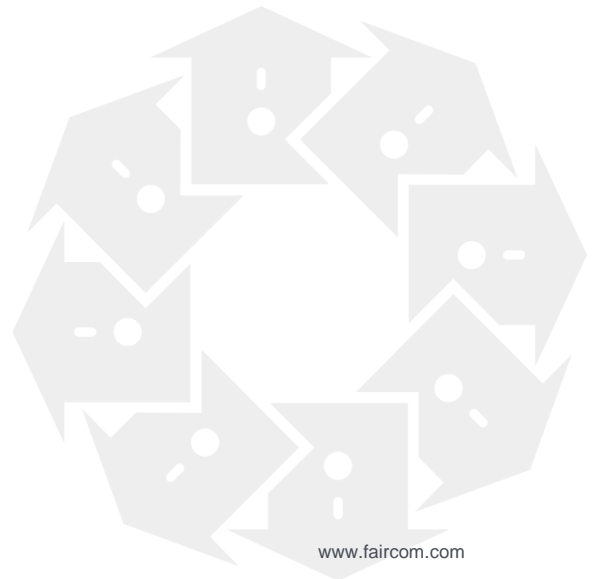


FairCom White Paper
Server Advantages



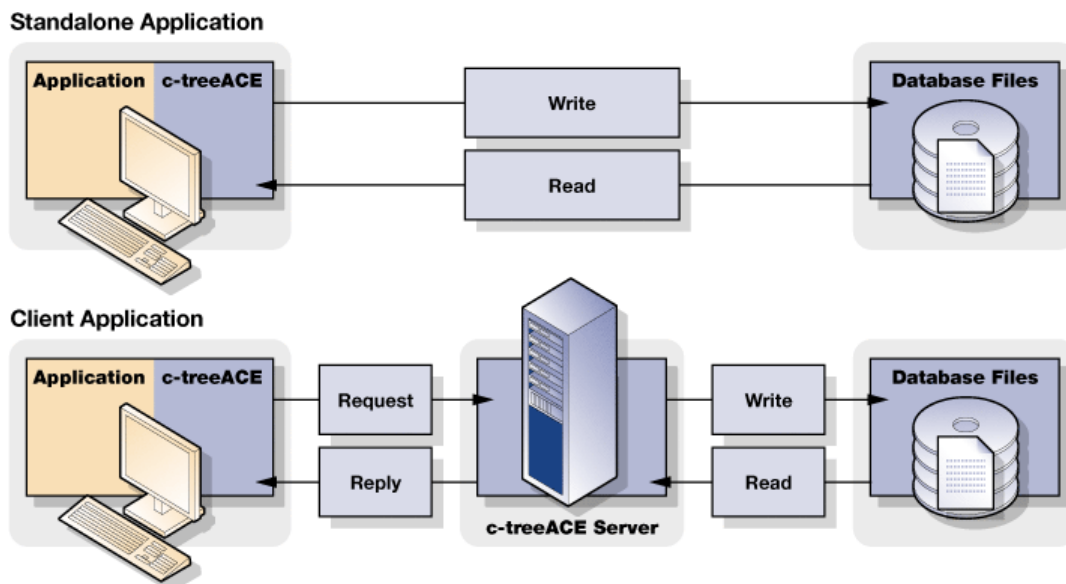
Contents

1.	Server Advantages	1
1.1	Client/Server Architecture	1
1.2	Advantages.....	2
1.3	Benefits.....	2
2.	Index	4



1. Server Advantages

Applications that rely on the c-tree “standalone” architecture use file and data management operations to directly manipulate groups of files and information in those files. The focus is on controlling files, and data, directly from an application. In a multi-user environment, as users and needs grow, the simple data access afforded by the multi-user standalone architecture (sometimes referred to as “FPUTFGET”) become cumbersome, unstable, or impossible to implement and maintain in a practical manner due to the fact that files and user applications reside on physically separate systems. The data integrity of the application is largely dependent on the file locking provided by the operating system -- complexities that are amplified when client machines are not on the exact same version and patch level.



Published 6/28/2019

1.1 Client/Server Architecture

The client/server architecture divides a logical application into a front-end client and a back-end database server. The client interacts with the application user to determine specific needs and only sends relevant requests to the server asking that the needed service be performed. The server accepts this request, performs the needed service, and relays the results back to the client. The client can then present the information to the user through whatever interface is appropriate. The server acts as a central traffic cop and can support many clients simultaneously. In addition, clients may even be different applications for a truly robust application suite.



1.2 Advantages

The key advantage becomes a division of labor between the client application and the centralized database server. Clients focus solely on the tasks they're intended to carry out for the user, while the server is solely dedicated to the most efficient data handling possible. This also eliminates the duplication of valuable resources such as memory and disk space that exists when these data management capabilities are incorporated into each application.

The most visible advantage, however, is pure performance. The complex issues of multiple users storing and retrieving data can be isolated within a single server process. As users increase in a strictly file-based environment, contention for those file resources quickly adds up. The server can consolidate those resources and take advantage of in-memory caching of data and indexes for scalability up to hundreds and even thousands of concurrent client connections. This is simply not possible with the standalone multi-user model.

1.3 Benefits

Client/server and multi-tier computing have become the models of choice in database systems for the most basic of reasons: increased speed, control, and efficiency in data management.

c-treeACE brings these benefits over traditional standalone models:

- **Performance:** In memory data and index caches mean data is immediately available to clients without expensive disk I/O. Standalone multi-user applications must flush every read and write operation to disk.
- **Scalability:** Scale from a few users to many users with a simple activation key. The multi-threaded server engine provides enhanced concurrency control and allows many more concurrent users to efficiently operate than is possible in a standalone multi-user environment.
- **Security:** Centralized data management provided by the server makes available many additional security features that are not practical in a standalone environment, such as user and file passwords; user, group, and file permissions; file and communication encryption; and logon control options. With only one process requiring access to data and index files, OS file permissions can be applied to restrict access to these files from casual users providing another layer of security. This is critically beneficial in health and financial industries in meeting HIPPA and other compliance measures.
- **Data Integrity:** Transaction processing makes available a multitude of advanced features not possible in multi-user standalone applications. Automatic-recovery with roll-forward/rollback control, dynamic "hot" backups, and a unique transaction history feature for elevated security tracking of changes are among these. Heavily used indices are automatically maintained for optimal speed and size, eliminating the need for rebuilding for performance.
- **Flexibility:** Heterogeneous support means Unix/Windows/Mac clients can share data across any combination of client/server platforms. Connect any client to any Server.
- **SQL:** c-treeACE SQL provides complete relational capabilities and is only available as an integral component of server-based models. SQL extends access to data through a variety of popular programming interfaces including ODBC, JDBC, ADO.NET, and PHP. These industry standard protocols allow data integration among multiple systems in the enterprise



Server Advantages

environment, vastly enhancing the availability of information through multiple channels including the web. Server-side stored procedures and triggers enforce business integrity rules on data and access for the ultimate in data integrity and security.

2. Index

A	
ADO.NET	2
Advantages	2
B	
back-end database server	1, 2
backup	2
Benefits	2
C	
centralized database server	1, 2
Client/Server Architecture	1
D	
Data Integrity	2
disk I/O	2
dynamic	2
F	
file-based environment	1, 2
Flexibility	2
FPUTFGET	1
front-end client	1
H	
Heterogeneous support	2
I	
in-memory caching	2
J	
JDBC	2
M	
multi-threaded	2
multi-tier computing	2
multi-user	2
O	
ODBC	2
P	
performance	2
PHP	2
R	
roll-forward/rollback	2
S	
Scalability	2
Security	2
Server Advantages	1
SQL	2
standalone	1, 2
stored procedures	2

T	
triggers	2

