

When to Use a Remote Call Instead of a File Server Access

Overview

A Fat Client application accessing indexed files through the isCOBOL File Server is an efficient architecture for many business applications. In this deployment model, the application logic executes on the client computer, while indexed file operations are performed through the File Server. Each indexed file operation requires communication with the File Server, generating network traffic between the client and the server.

However, some business operations may require reading or updating a very large number of records while returning only a small amount of information to the user. In these situations, the network traffic generated by the file operations may become the dominant factor affecting performance.

An alternative is to execute that specific business operation on an Application Server through a Remote Call. In this case, the client invokes a program that runs on the Application Server, where all file processing is performed locally. Only the input parameters and the results of the operation are exchanged between the client and the server, significantly reducing the amount of network traffic generated by the business operation.

This article demonstrates this approach using a simple sample application.

Sample Scenario

The downloadable sample contains CALLSEARCHCUST.CBL, SEARCHCUST.CBL and a JISAM indexed file containing approximately 200,000 customer records.

SEARCHCUST.CBL sequentially scans the entire file, applies selection criteria and returns only the three customers with the most recent purchase date. The sample intentionally scans all records to illustrate the impact of network traffic.

Compile the programs:

```
iscc *.cbl
```

Sample Configuration to run using the File Server

runfs.properties:

```
iscobol.file.index=remote
iscobol.file.remote.host=127.0.0.1
iscobol.file.remote.port=10998
```

Run Command:

```
Iscserver -fs
```

```
iscrun -c runfs.properties CALLSEARCHCUST
```

Both CALLSEARCHCUST.CBL and SEARCHCUST.CBL execute on the Fat Client. File access is performed remotely through the File Server.

Sample Configuration to run with Remote Call against an Application Server

```
runas.properties:
iscobol.remote.code_prefix=isc://127.0.0.1:10995
```

Run Command:

```
iscserver
```

```
iscrun -c runas.properties CALLSEARCHCUST
```

CALLSEARCHCUST is executed on the fat client, but the SEARCHCUST program should be located on the application server with the data files so that it can run locally to the files.

Relevant File Processing Logic in SEARCHCUST

Here is the important logic in SEARCHCUST. It intentionally scans approximately 200,000 records while returning only three records to the caller.

```
MOVE LOW-VALUES TO cust-key
START cust-file KEY IS NOT < cust-key
  INVALID KEY EXIT PARAGRAPH
END-START
```

```
PERFORM UNTIL EXIT
  READ cust-file NEXT RECORD
  AT END EXIT PERFORM
END-READ
```

```
IF ... matching criteria ...
  PERFORM update-top-3
END-IF
END-PERFORM
```

When a Remote Call Can Help

If a business operation performs a large number of file reads and/or updates, executes significant business logic and returns only a relatively small amount of information to the client, executing the operation on the Application Server through a Remote Call can significantly reduce network traffic. This principle is not limited to sequential scans.

Conclusion

The File Server remains an appropriate solution for many applications. However, if performance analysis identifies operations dominated by remote file access, moving those specific operations to the Application Server through a Remote Call is often worth evaluating.

This deployment flexibility gives developers the power to optimize individual business operations without redesigning the overall application architecture. Only the operations that are network-intensive need to be moved to the Application Server, while the remainder of the application can continue using the File Server model.

Online URL: <https://support.veryant.com/phpkb/article.php?id=359>