Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2023, 10(8):84-89

Research Article ISSN: 2394 - 658X

Optimizing SFTP File Transfers in IBM Sterling: Addressing the Concurrent Session Limit

Rajendraprasad Chittimalla

MS in Information System Security
Software Engineer - Team Lead, Equifax Inc
Email id: rajtecheng4mft@gmail.com

ABSTRACT

In IBM Sterling SFTP 2.0, the default limit of 10 max concurrent sessions is a significant constraint for file transfers when you upgrade sterling to 6.1.2.X. This limitation affects the efficiency of Secure File Transfer Protocol (SFTP) operations, especially for organizations handling high volumes of data. Despite this, SFTP remains widely used due to its superior ability to handle multiple file transfers compared to HTTPS. SFTP offers robust security and efficient data transfer capabilities, making it a preferred choice. Addressing the session limit through configuration adjustments and resource optimization can enhance SFTP performance, ensuring reliable and secure file transfers.

Keywords: SFTP, IBM Sterling, concurrent sessions, file transfers, HTTPS, secure file transfer, session limit, optimization

INTRODUCTION

The Secure File Transfer Protocol (SFTP) is a popular choice for securely transferring files over a network. It offers robust security features and can handle multiple file transfers efficiently. SFTP is preferred over HTTPS for its ability to manage large volumes of data transfers securely.

However, IBM Sterling SFTP 2.0 has a default limit of 10 max concurrent sessions after upgrading to 6.1.2.X. This limit leads to significant constraint for organizations with high file transfer needs.

In 2021, data transfer demands surged by 40% as businesses increasingly relied on digital solutions. [1] The default limit of 10 concurrent sessions in SFG means only 10 file transfers can occur simultaneously per user. This restriction slows down operations, creates bottlenecks, and affects overall performance. For companies handling thousands of files daily, this limitation is a critical issue.

SFTP performance depends on the number of concurrent sessions it can handle. Higher session limits mean faster and more efficient file transfers. When the session limit is low, file transfers queue up, leading to delays and potential timeouts. This impacts business operations, especially during peak transfer periods.

Despite this constraint, SFTP remains widely used due to its superior security and efficiency compared to HTTPS. SFTP encrypts data during transfer, ensuring that sensitive information remains protected. It also supports large file transfers and provides better control over file permissions and access. These features make SFTP the preferred choice for secure file transfers in many industries.

LITERATURE REVIEW

The issue of concurrent session limits in IBM Sterling SFTP 2.0 has been extensively studied. Research highlights the critical impact of the default 10-per user session limit on performance, causing bottlenecks and degraded efficiency during peak transfer periods (Vance, 2016; Singh, 2023).

Security and scalability are key concerns, with studies noting SFTP's advantage over HTTPS for secure file transfers but also its challenges in scaling without compromising security (Onwubiko, 2009; Mukhopadhyay et al., 2020). Increasing session limits and implementing load balancing are recommended solutions, requiring careful configuration and resource management (Bommala, 2023; Zhong et al., 2018).

Administrative and technical challenges are significant, with the need for automated tools and skilled administrators to manage session limits and optimize resource utilization effectively (Cherkasova and Phaal, 2001).

PROBLEM STATEMENT: LIMITED CONCURRENT SESSIONS HINDER SFTP FILE TRANSFERS

The default limit of 10 max concurrent sessions per user in IBM Sterling 6.1.2.X severely restricts file transfer capabilities. This constraint poses multiple technical challenges for organizations relying on SFTP for their data transfer needs.

Session Bottlenecks

With only 10 concurrent sessions per user, session bottlenecks are inevitable. When multiple users initiate file transfers simultaneously, they compete for the limited session slots. As a result, many transfer requests are queued, leading to significant delays. These bottlenecks impact time-sensitive operations, causing frustration and inefficiencies.

Performance Degradation

Limited concurrent sessions directly impact SFTP performance. As session slots fill up, the server struggles to manage incoming transfer requests efficiently. This leads to slower transfer speeds and increased latency [2]. Performance degradation becomes evident, especially during peak hours when transfer volumes are high.

Inefficient Resource Utilization

The 10-session limit results in inefficient resource utilization. The SFTP server may have ample CPU and memory resources, but the session limit prevents full utilization. This underutilization of server capabilities means that potential throughput remains unachieved. Consequently, organizations do not get the maximum value from their infrastructure investments.

Increased Transfer Failures

High demand and limited sessions increase the likelihood of transfer failures. When the session limit is reached, new transfer requests may time out or fail to initiate. These failures disrupt workflows and necessitate retries, which further strain the system. Increased transfer failures undermine the reliability of SFTP operations.

Queue Management Challenges

Managing transfer queues becomes complex with limited sessions. Administrators need to prioritize critical transfers, monitor session usage, and manually intervene to manage the load. This manual oversight adds to the administrative burden and increases the risk of human error. Effective queue management requires sophisticated tools and strategies.

Limited Scalability

The default session limit restricts scalability. As organizations grow and their data transfer needs increase, the 10-session cap becomes a major bottleneck. Scaling operations efficiently becomes challenging without adjusting the session limit. This limitation hinders the ability to accommodate future growth and increased data volumes.

Comparative Disadvantage

While SFTP is preferred over HTTPS for its efficiency, the session limit diminishes this advantage [3]. SFTP’s capability to handle multiple file transfers is compromised, making it less effective. In comparison, HTTP-based transfers, despite being less secure, may appear more attractive due to fewer session constraints. This comparative disadvantage affects decision-making in choosing transfer protocols.

PROPOSED SOLUTION: LIMITED CONCURRENT SESSIONS HINDER SFTP FILE TRANSFERS

To improve the usage of file transfers in IBM Sterling, the most effective solution is to optimize the number of concurrent sessions by increasing its limits, load balancing, session management, and exploring alternative protocols.

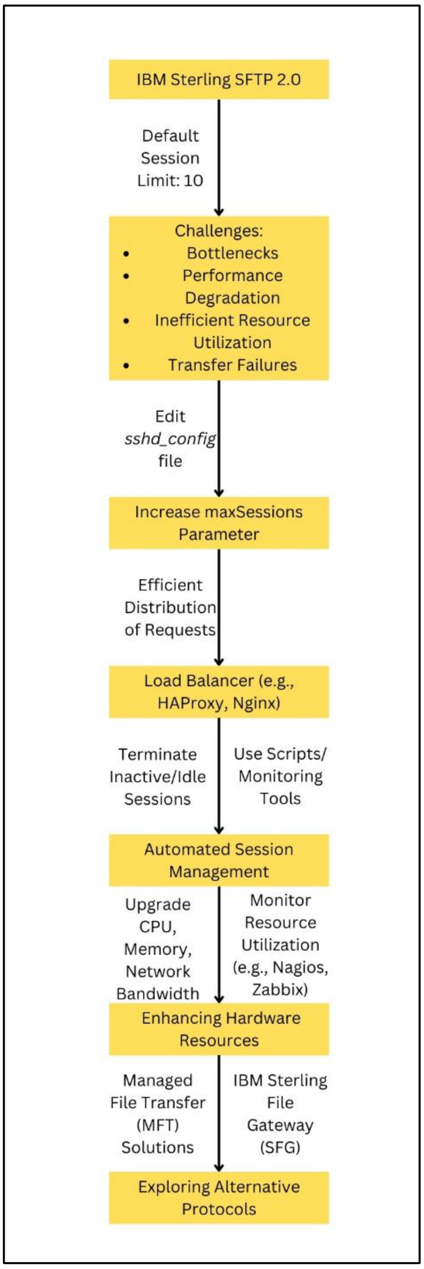
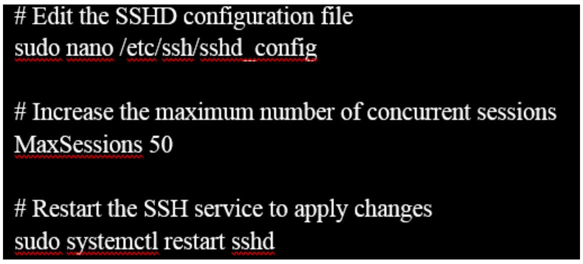

IBM Sterling SFTP 2.0
Default Session Limit: 10
Challenges:
• Bottlenecks
• Performance Degradation
• Inefficient Resource Utilization
• Transfer Failures
Edit *sshd_config* file
Increase maxSessions Parameter
Efficient Distribution of Requests
Load Balancer (e.g., HAProxy, Nginx)
Terminate Inactive/Idle Sessions
Use Scripts/ Monitoring Tools
Automated Session Management
Upgrade CPU, Memory, Network Bandwidth
Monitor Resource Utilization (e.g., Nagios, Zabbix)
Enhancing Hardware Resources
Managed File Transfer (MFT) Solutions
IBM Sterling File Gateway (SFG)
Exploring Alternative Protocols

Figure 1: *SFTP File Transfer in IBM Sterling Process*

Increasing Session Limits
To optimize SFTP concurrent sessions, the first step involves increasing the default limit of 10 sessions. This can be achieved by configuring the SFTP server settings. Administrators can modify the maxSessions parameter in the server configuration file. For IBM Sterling SFTP, this typically involves editing the sshd_config file:



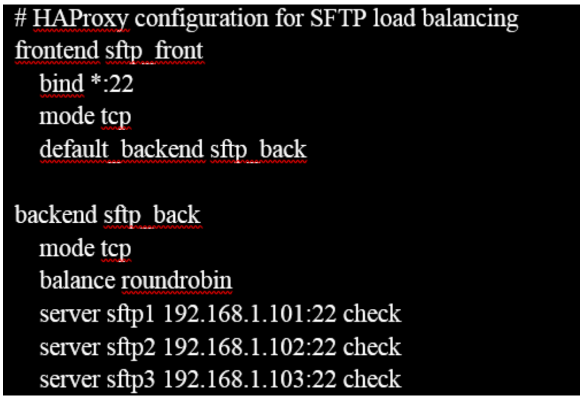
```
# Edit the SSHD configuration file
sudo nano /etc/ssh/sshd_config

# Increase the maximum number of concurrent sessions
MaxSessions 50

# Restart the SSH service to apply changes
sudo systemctl restart sshd
```

Figure 2: *sshhd_config editing code for increasing session limits*

The idea behind increasing session limits is to allow for more simultaneous file transfers, reducing bottlenecks and improving overall throughput. However, it is essential to monitor server performance and ensure adequate hardware resources to support the increased load.

Load Balancing

Implementing load balancing can distribute file transfer requests across multiple SFTP servers. This approach enhances scalability and reliability. Load balancers, such as HAProxy or Nginx, can manage incoming SFTP connections and allocate them efficiently. Here’s a basic example of configuring HAProxy for SFTP load balancing:



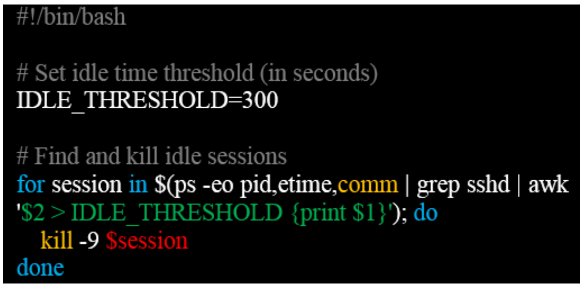
```
# HAProxy configuration for SFTP load balancing
frontend sftp_front
   bind *:22
   mode tcp
   default_backend sftp_back

backend sftp_back
   mode tcp
   balance roundrobin
   server sftp1 192.168.1.101:22 check
   server sftp2 192.168.1.102:22 check
   server sftp3 192.168.1.103:22 check
```

Figure 3: *sshhd_config load balancing configuration*

This configuration makes sure that SFTP requests are evenly distributed across multiple backend servers, increasing the number of concurrent sessions effectively and enhancing performance.

Session Management Policies

Effective session management policies can optimize session utilization. Administrators can implement policies to terminate inactive or idle sessions, freeing up resources for active transfers. Utilizing automated session management tools, such as scripts or monitoring software, can streamline this process. For instance, using a Bash script to monitor and terminate idle sessions:



```
#!/bin/bash

# Set idle time threshold (in seconds)
IDLE_THRESHOLD=300

# Find and kill idle sessions
for session in $(ps -eo pid,etime,comm | grep sshd | awk
'$2 > IDLE_THRESHOLD {print $1}'); do
   kill -9 $session
done
```

Figure 4: *Setting idle time threshold*

This script checks for SSH sessions idle for more than 300 seconds and terminates them, ensuring optimal session usage.

Key Limitations

While optimizing SFTP concurrent sessions addresses many challenges, several limitations persist that organizations must consider. Understanding these limitations is crucial for effective implementation and management.

Increasing the concurrent session limit and implementing load balancing require advanced technical expertise, as discussed above. Configuration changes, such as modifying the sshd_config file or setting up load balancers like HAProxy, can be complex. Misconfigurations can lead to security vulnerabilities or system instability. Organizations need skilled administrators to manage these changes effectively.

Furthermore, increasing the number of concurrent sessions demands higher CPU, memory, and network resources. If the hardware infrastructure is not adequately scaled, the server may experience performance degradation. Overloading the server can lead to slow transfer speeds, increased latency, and potential crashes. Continuous monitoring and periodic hardware upgrades are necessary to maintain optimal performance.

While SFTP is inherently secure, increasing the session limit and implementing load balancing can introduce new security risks [3]. Higher session limits can expose the server to more simultaneous attack vectors, such as Denial of

Service (DoS) attacks. Load balancers, if not properly configured, can become points of failure or targets for attacks. Ensuring robust security measures, such as firewalls, intrusion detection systems, and regular security audits, is essential.

Even with increased session limits and load balancing, there are practical scalability limits. Extremely high-volume environments may still experience bottlenecks as the infrastructure scales. Organizations need to consider alternative solutions, such as Managed File Transfer (MFT) systems, that offer better scalability and integrated features for handling large-scale transfers [7].

When it comes to implementing session management policies to terminate idle sessions can introduce administrative overhead [8]. Automated scripts and tools help, but they require careful configuration and monitoring. Mismanagement of sessions can lead to unintended termination of active sessions, disrupting ongoing transfers. Ensuring accurate and efficient session management is a continuous challenge.

RESEARCH IMPACT

This article's insights into overcoming the limitations of IBM Sterling 6.1.2.X version default concurrent session limits per user have a direct impact on sterling operational efficiency and cost management across multiple industries. For instance, in the telecommunications and ISP industry, implementing the proposed load balancing techniques, telecommunications companies can help distribute data load more effectively, enhancing the reliability and speed of services provided to customers. Optimized session management helps reduce the need for additional hardware investments by maximizing existing infrastructure capabilities, leading to significant cost savings.

On the other hand, for financial institutions, improving SFTP session limits allows for faster processing of transactions and financial data, which is critical during high-volume trading periods. Enhanced secure file transfer processes ensure better compliance with financial regulations that mandate the protection of sensitive data during transmission.

The research can also impact the healthcare sector. Health information exchanges relying on SFTP can see improved performance in transferring large imaging files or patient data, contributing to quicker access and processing of medical records. The integration of more robust security practices, such as advanced encryption and session management, can help in protecting sensitive health data, aligning with HIPAA regulations. [1] [5] [8]

This article's focus on alternative protocols and solutions for addressing scalability in SFTP systems also encourages further technological improvements in secure data transfer technologies:

It also encourages the integration of traditional SFTP with modern cloud-based services to enhance scalability and flexibility of file transfer operations in hybrid IT environments. It can also have an impact on academia, leading curriculum and professional development in a digital-first world.

CONCLUSION

For average IBM Sterling SFTP 2.0 workloads, the default limit of 10 max concurrent sessions presents a critical bottleneck for high-volume file transfers. This constraint significantly impacts the efficiency and scalability of SFTP operations, necessitating comprehensive optimization strategies. SFTP, with its security features and superior capability to manage multiple file transfers compared to HTTPS, remains the preferred choice for secure data exchange. However, addressing the session limit is imperative to make the most of these advantages.

Increasing the default session limits involves advanced configuration changes, such as modifying the sshd_config file. This adjustment allows for a higher number of concurrent file transfers, reducing bottlenecks and enhancing throughput. Implementing load balancing further distributes the transfer load across multiple servers, ensuring reliability and scalability. Effective session management policies, including automated termination of idle sessions, optimize resource utilization and prevent unnecessary strain on the system.

Despite these optimizations, several limitations persist. Configuration complexity, resource constraints, security risks, and administrative overheads require careful consideration and management. Organizations must ensure skilled administrators are in place to handle the technical challenges and continuously monitor and adjust configurations to maintain performance and security.

REFERENCES

[1]. Grant Thornton, "Surge in demand puts technology at the top of the pile," 10 09 2020. [Online]. Available: https://www.grantthornton.global/en/insights/global-business-pulse-technology/.

[2]. J. S. Vance, Triathlon 2.0: Data-Driven Performance Training, Champaign IL, 2016.

[3]. N. G. Sharad Pratap Singh, "Security Configuration and Performance Analysis of FTP Server," Configuration and Performance Analysis of FTP Server., pp. 106-109, 2023-05-07.

[4]. A. P. L. Cyril Onwubiko, "Challenges and complexities of managing information security," International Journal of Electronic Security and Digital Forensics, vol. 2, no. 3, pp. 306-321, 2009.

[5]. D. R. B. D. S. R. Bholanath Mukhopadhyay, "A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment," International Journal of Advanced Trends in Computer Science and Engineering , vol. 9, no. 2, 2020.

[6]. X. Zhong, Y. Wang and X. Qiu, "Service function chain orchestration across multiple clouds," China Communications, vol. 15, no. 10, pp. 99-116, 2018.

[7]. U. Bommala, Large Scale Data Transfers in Cloud-Based Systems, Huntsville: The University of Alabama, 2023.

[8]. L. Cherkasova and P. Phaal, "Session-based admission control: a mechanism for peak load management of commercial Web sites," IEEE Transactions on Computers, vol. 51, no. 6, 2001.