

Vast Data OST Driver

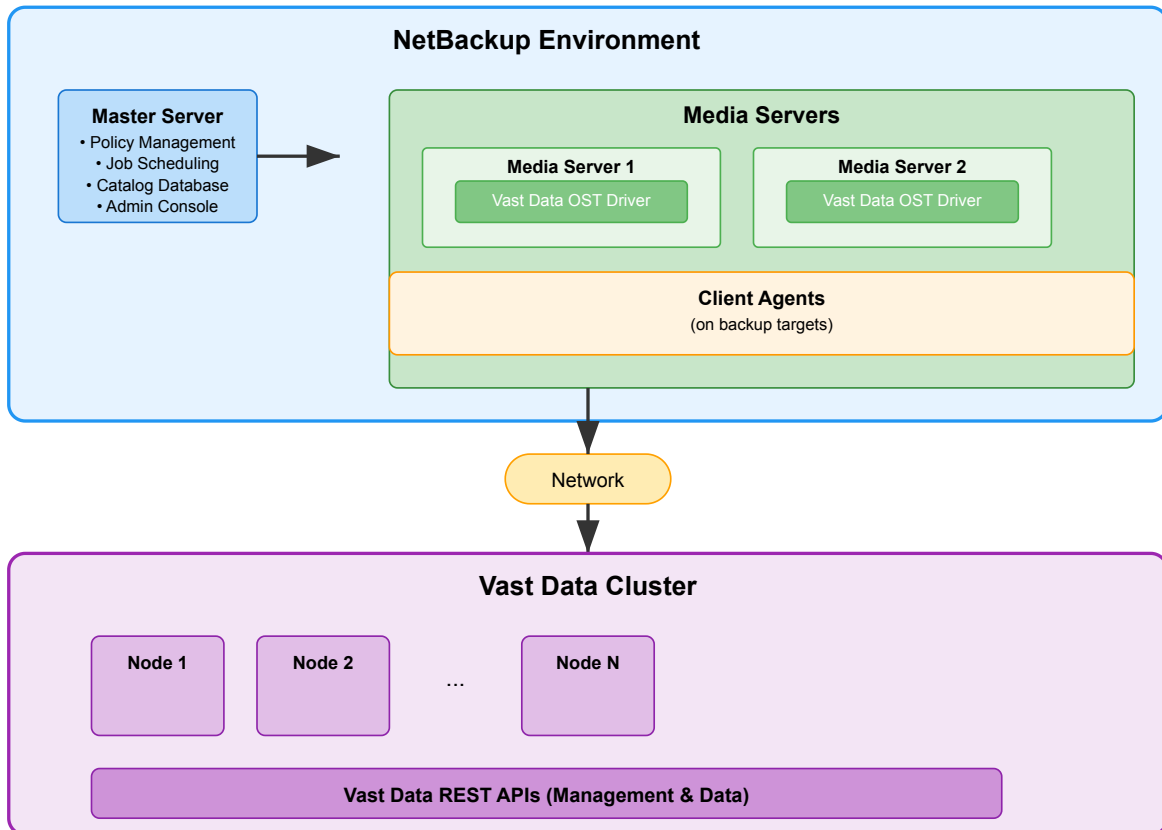
Architecture & Implementation Plan

NetBackup OpenStorage Technology Integration

Technical Architecture Document

Enabling enterprise-grade backup integration
between NetBackup and Vast Data storage systems

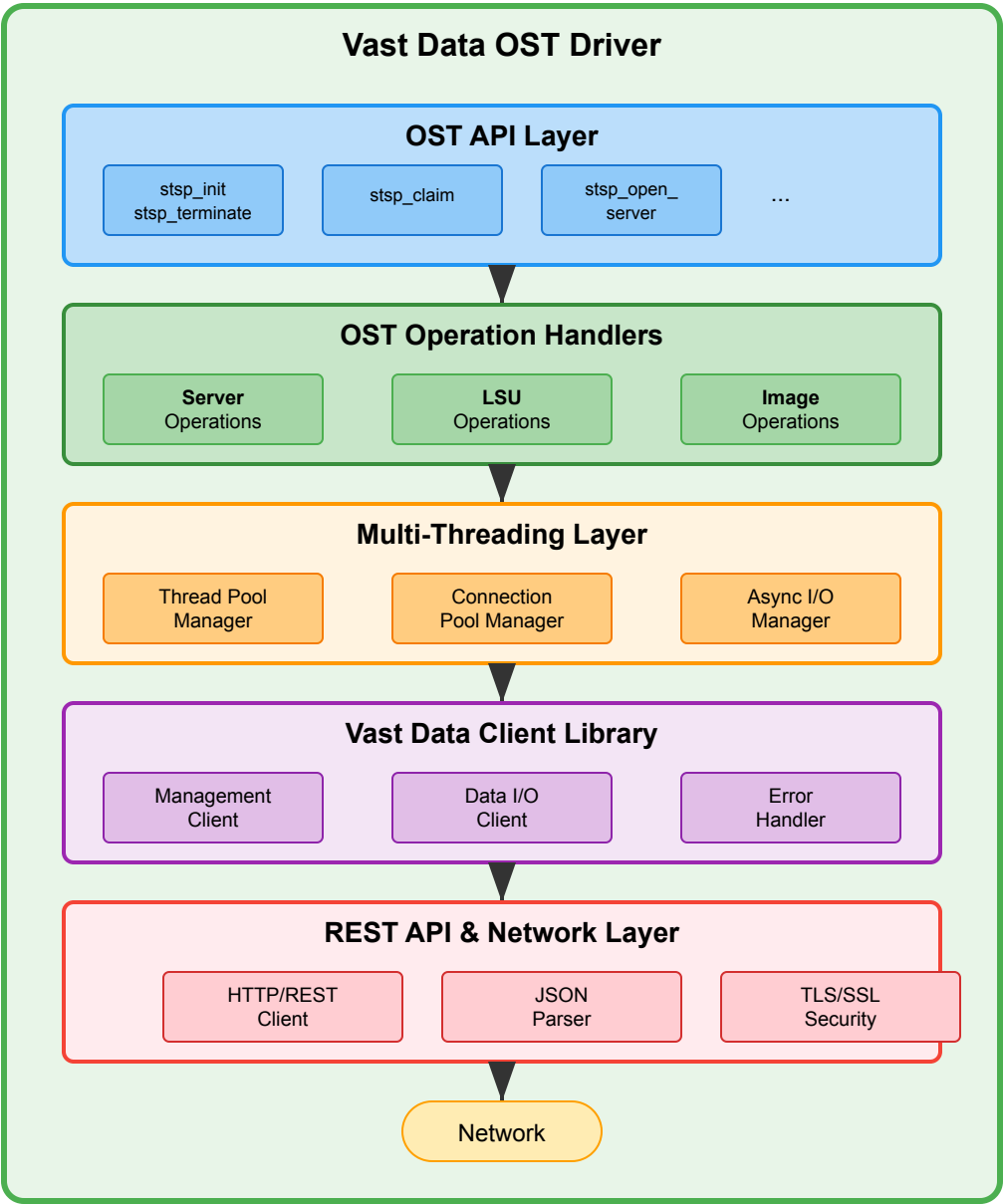
1. High-Level OST Driver Architecture



Key Architecture Components:

- **Master Server:** Single management point with NetBackup console
- **Media Servers:** Multiple workers running OST drivers
- **OST Driver:** Translation layer between NetBackup and Vast Data
- **Vast Data Cluster:** Scale-out storage with REST API access

2. Detailed OST Driver Internal Architecture



OST Threading Model:

OST Handle Requirements:

- **Single-Threaded Handles:** Each OST handle (server, image, LSU) can only be accessed by the thread that created it
- **No Handle Sharing:** Handles cannot be shared between threads - each thread creates its own handles
- **Plugin Versions:** Must build both single-threaded (.so) and multithreaded (MT.dll/.so) versions
- **Internal Thread Safety:** Plugin internal implementation must be thread-safe using synchronization primitives
- **Concurrency Model:** NetBackup creates multiple threads, each with dedicated handle sets

Architecture Understanding:

- **NetBackup Core Library:** Manages multiple threads, each with its own set of handles
- **OST Driver Interface:** Provides single-threaded handles as per OST specification
- **Internal Implementation:** Can use thread-safe shared resources (connection pools, etc.)
- **Concurrency Model:** Multiple threads each with dedicated handles, not shared handles

3. Key Integration Points with Vast Data

Required Vast Data REST APIs

Management REST APIs

- Volume/Namespace creation and management
- User authentication and authorization
- Quota and capacity management
- Performance monitoring and statistics
- Cluster health and status reporting

Data REST APIs

- High-performance file I/O via REST
- Directory and metadata operations
- Atomic file operations (create, write, close)
- File deletion and cleanup
- Snapshot and versioning APIs

Performance Requirements

Throughput Targets

- **Single stream:**
Maximize per-stream throughput
- **Multiple streams:**
Scale linearly with stream count
- **Concurrent jobs:**
Support enterprise-scale concurrent backups

Latency Targets

- **File operations:**
Minimize API call latency
- **First byte:** Optimize connection establishment
- **File completion:**
Efficient commit operations

Scalability

- **Media Servers:**
Support multiple concurrent servers
- **Connections:**
Efficient connection pooling
- **Storage capacity:**
Scale with Vast Data cluster

4. Success Metrics & Deliverables

Technical Deliverables

- ✓ Complete OST driver (.so/.dll files)
- ✓ Source code with comprehensive documentation
- ✓ Unit and integration test suites
- ✓ Performance benchmarking tools
- ✓ NetBackup certification results

Documentation

- ✓ Installation and configuration guide
- ✓ Administrator's reference manual
- ✓ Troubleshooting and support guide
- ✓ Performance tuning recommendations
- ✓ Integration architecture documentation

Success Criteria

- | | |
|---|---|
| ✓ Pass NetBackup OST certification | ✓ Achieve target performance benchmarks |
| ✓ Support enterprise-scale deployments | ✓ 99.9%+ backup job success rate |
| ✓ Successful customer pilot deployments | ✓ Seamless NetBackup admin experience |

Key Benefits

- **Native Integration:** Deep NetBackup integration vs generic protocols
- **High Performance:** Multi-stream support for enterprise throughput
- **Scalability:** Support for multiple Media Servers and concurrent operations
- **Enterprise Features:** Advanced backup operations like image duplication
- **Simplified Management:** Single NetBackup interface for all operations