

High-Performance Computing (HPC) Management Software

Module: HPC Manager

1. Overview

The *HPC Manager* project provides a unified web interface for High-Performance Computing cluster administration.

It integrates **React (Frontend)** and **Node.js (Backend)** to interact with **SLURM**, **LDAP**, and **Database Services** (MariaDB, MongoDB) to handle cluster operations, job scheduling, monitoring, and data management.

The project follows a **microservice-style modular architecture** to ensure scalability, security, and maintainability.

2. System Architecture

2.1 Architecture Summary

Layer	Technology	Description
Frontend	React 19.1.0	SPA dashboard for cluster, job, and file management
Backend	Node.js v20.15 / Express.js	RESTful API and service orchestration
Databases	MongoDB Atlas(Now Removed) / MariaDB / MySQL (SLURM Accounting DB)	Persistent storage for admin/user data, SLURM metrics
Directory Services	LDAP	Centralized authentication (LDAPS secured)

Containerization	Docker 20.10+	Deployment packaging for frontend, backend, and databases
Monitoring	Prometheus + Grafana	Node and cluster performance visualization
Web Proxy	Nginx	Reverse proxy and load balancing

3. System Modules

3.1 Frontend (React)

Framework: React 19.1.0

Package Manager: NPM v10.8

Key Modules:

- **Dashboard Module:** Displays real-time cluster status via WebSockets.
- **Job Management Interface:** Allows submission, tracking, and cancellation of jobs.
- **File Manager:** Supports file CRUD operations with drag-and-drop upload.
- **User Management Panel:** Handles admin-level user CRUD operations.

Enhancements & Contributions:

- Refactored UI components for better responsiveness and dynamic state management.
- Implemented real-time updates for **Munge**, **NFS**, **Sync Node**, and **Compute Node Agent Controller** buttons.
- Debugged file management module (directory navigation, upload/download issues).
- Optimized API integration between frontend and backend (async error handling).

Frontend Commands:

```
cd frontend
```

```
npm install
```

```
npm run dev
```

3.2 Backend (Node.js / Express.js)

Runtime: Node.js v20.15

Core Dependencies:

- express
- mongoose
- mariadb
- jsonwebtoken
- bcryptjs
- ldapjs
- axios
- socket.io

Backend Modules:

- **Authentication Service:**
 - LDAP integration with role-based access control.
 - JWT-based session management with MFA support.
- **Job Scheduler Interface:**
 - Connects to SLURM for job submission, queue management, and cancellation.

- Uses REST endpoints to communicate with the SLURM accounting database.
- **Cluster Management Service:**
 - Handles real-time cluster and node status monitoring.
 - Includes auto-refresh via WebSockets for performance metrics.
- **System Services:**
 - API integrations for NFS, Munge, and Node Syncing operations.
 - All services modularized under `/services` directory.

Enhancements & Contributions:

- Developed and tested backend logic for service controllers (Munge, NFS, Sync Node).
- Implemented error-handling middleware for stable API responses.
- Enhanced system logging for debugging through Winston and Morgan.
- Optimized SLURM data fetch intervals and reduced API latency.

Backend Commands:

```
cd backend
```

```
npm install
```

```
npm run start
```

4. Database Architecture

4.1 SLURM Accounting Database (MySQL)

HPC Manager connects to the existing `slurm_acct_db` for retrieving and displaying compute job data.

Important Tables:

- `cluster_job_table` – Job metadata and execution state
- `cluster_assoc_table` – User-account associations
- `cluster_usage_hour_table` – Hourly utilization stats
- `res_table` – Resource definitions
- `user_table` – User data from SLURM accounting

4.2 Admin & User Databases

Database	Engine	Purpose
MongoDB Atlas	NoSQL	Stores admin and configuration metadata
MariaDB	Relational	Stores user data, credentials, and job history

Configuration File:

```
.env
```

```
MONGO_URI=mongodb+srv://...
```

```
MYSQL_HOST=localhost
```

```
MYSQL_USER=root
```

```
MYSQL_PASS=*****
```

```
JWT_SECRET=*****
```

Note: Replace these values during deployment to ensure security.

5. Installation and Environment Setup

5.1 Prerequisites

Component	Version
OS	Rocky Linux 8.10
Node.js	20.15+
NPM	10.8+
Docker	20.10+
MongoDB	6.0+
MariaDB	11.6.2+
Nginx	Latest Stable

5.2 Setup Steps

```
git clone https://github.com/[repo]/hpc-manager.git
```

```
cd hpc-manager
```

```
npm install
```

```
npm run dev
```

5.3 Test Login

- **Username:** aetpl
- **Password:** 123456

Modify `.env` before final deployment with updated credentials and secrets.

6. API Endpoints (RESTful Design)

Method	Endpoint	Description
POST	<code>/api/auth/login</code>	Authenticate user
GET	<code>/api/users</code>	Fetch all users (Admin only)
POST	<code>/api/users/add-user</code>	Add new user
PUT	<code>/api/users/:id</code>	Update user
DELETE	<code>/api/users/:id</code>	Remove user
GET	<code>/api/jobs</code>	List user jobs
POST	<code>/api/jobs/submit</code>	Submit new job
GET	<code>/api/jobs/:id</code>	Fetch job details
DELETE	<code>/api/jobs/:id</code>	Cancel job
GET	<code>/api/cluster/status</code>	Cluster health summary
GET	<code>/api/cluster/nodes</code>	Fetch node details
GET	<code>/api/cluster/usage</code>	Resource utilization overview

7. Logging, Monitoring & Testing

- **Logging:** Configured via Winston (`/config/logger.js`) for API and service-level logging.
- **Monitoring:** Prometheus scrapes Node Exporter metrics; Grafana visualizes performance dashboards.
- **Testing:** Postman Collection included (`/tests/hpc-manager.postman_collection.json`).
- **LDAP Testing Tool:** Apache Directory Studio used for verifying DN-based authentication.

8. Known Issues & Future Improvements

Area	Description	Recommendation
File Manager	Upload latency for large files	Implement chunk-based upload
Service Control	Timeout on heavy cluster	Introduce retry mechanism
Job Scheduler	Limited dependency management	Add DAG-based scheduling
UI	Missing real-time graph refresh	Integrate socket.io events on metric updates

9. Contributor Notes

Tasks Completed by [Your Name]:

- UI enhancements for cluster service controls (Munge, NFS, Sync Node, Compute Node Agent Controller).

- Frontend bug fixes in File Management and Job History modules.
 - Backend API integration for cluster service modules.
 - Tested and validated LDAP integration via Apache Directory Studio.
 - Updated `.env` and `config.js` for MongoDB and MariaDB connections.
 - Improved frontend responsiveness and reduced API call latency.
-

10. References & Access

Resource	Description
<code>/frontend/src/</code>	React UI source code
<code>/backend/src/</code>	Node.js services and APIs
<code>/db/</code>	Database schemas and setup scripts
<code>/docs/</code>	Technical documentation and architecture diagrams
<code>/tests/</code>	Postman API testing files