Performance Requirements

> **Source File:** generated-documents\technical-design\performance-requirements.md
> **Generated:** 16/07/2025 at 14:00:40
> **Generated by:** Requirements Gathering Agent - PDF Converter

PerformanceRequirements

Generated by adpa-enterprise-framework-automation v3.2.0
Category: technical-design
Generated: 2025-07-14T21:05:44.091Z
Description:

Certainly! Based on the provided project context and best practices for enterprise software performance requirements, here is a comprehensive set of **Performance Requirements** for the ADPA (Advanced Document Processing & Automation Framework):

Performance Requirements: ADPA Enterprise Framework

1. Performance Goals

- **High Responsiveness:** Ensure all document generation, API, and integration operations meet enterprise-grade responsiveness suitable for business-critical workflows.

- **Consistent Throughput:** Maintain steady throughput for concurrent document generation, AI requests, and integration tasks, even under peak usage.
- **Enterprise Scalability:** Seamlessly scale horizontally to support large teams, multiple projects, and high-volume automation scenarios.
- **Reliability Under Load:** Guarantee system stability and predictable degradation under high or unexpected load.

---

2. Response Time Requirements

2.1 API Endpoints (Measured at 95th percentile)

- **Document Generation (POST /api/v1/generate):**
  - ≤ 3 seconds for standard templates (≤ 10 pages)
  - ≤ 8 seconds for complex/AI-powered templates (≤ 30 pages)
- **Templates Listing (GET /api/v1/templates):**
  - ≤ 500ms
- **Confluence/SharePoint Publishing:**
  - ≤ 5 seconds for documents ≤ 5 MB
- **Health/Readiness Endpoints:**
  - ≤ 200ms
- **Admin Web Interface Page Loads:**
  - ≤ 1.0 second for dashboard and core pages

2.2 CLI Operations

- **CLI Command Feedback:**
  - Initial response ≤ 500ms
  - Completion of standard document generation ≤ 5 seconds

---

3. Throughput Expectations

- **Concurrent API Requests:**
  - Support at least 100 concurrent active API sessions without degradation.
- **Document Generation:**
  - Minimum 20 documents/minute (with average document size ≤ 10 pages) per node.
- **AI Provider Calls:**
  - At least 50 AI provider calls/minute per node, with automatic failover.
- **Integration Operations:**
  - Simultaneous publishing to at least 3 integrations (e.g., Confluence, SharePoint, Adobe) without blocking.

---

4. Scalability Requirements

- **Horizontal Scaling:**
  - The system must support horizontal scaling via stateless microservices.
  - Adding new API server/worker nodes should linearly increase throughput.
- **Elastic Resource Management:**
  - Support dynamic scaling in cloud/containerized environments (e.g., Docker/Kubernetes).
- **Multi-tenant Support:**
  - System must maintain isolation and fair resource allocation across multiple enterprise clients.

---

5. Resource Utilization

- **CPU Utilization:**
  - Average CPU usage per node ≤ 70% under typical load; never exceeds 90% for >5 minutes.
- **Memory Utilization:**

    - Average memory usage ≤ 75% of available memory per service.
- **Disk I/O:**
    - Document generation and storage processes should not exceed 80% of disk I/O bandwidth.
- **Network:**
    - API and integration endpoints must not saturate available network bandwidth; utilize connection pooling and keep-alive.

---

6. Load Handling

- **Graceful Degradation:**
    - Under overload, system must return 429 (Too Many Requests) or similar error, never crash.
- **Backpressure:**
    - Implement request queueing and rate limiting (e.g., express-rate-limit) to avoid overload.
- **Retry Strategy:**
    - Failed AI provider or integration calls must be retried with exponential backoff, up to 3 times.

---

7. Caching Strategy

- **Template Caching:**
    - Frequently used document templates and assets must be cached in memory (e.g., Redis) for ≤ 100ms retrieval.
- **AI Provider Response Caching:**
    - Cache non-personalized AI responses to reduce duplicate provider usage.
- **API Response Caching:**
    - Cache non-sensitive API responses (e.g., template lists) for at least 60 seconds.
- **Invalidation:**
    - Invalidate cache on template or configuration updates.

8. Performance Metrics

- **API Latency:**
  - 95th/99th percentile response times for all endpoints.
- **Throughput:**
  - Requests per second (RPS) per node/service.
- **Error Rate:**
  - ≤ 0.1% failed requests under normal load.
- **Resource Usage:**
  - CPU, memory, disk I/O, network per node.
- **External API Latency:**
  - Average and max response time for each AI provider/integration.
- **Queue Depth:**
  - Number of queued/rate-limited requests.
- **Cache Hit Ratio:**
  - ≥ 90% for template cache.

---

9. Monitoring Requirements

- **Real-Time Dashboards:**
  - Integrate with monitoring tools (e.g., Prometheus, Grafana, Azure Monitor) for live visibility into all metrics above.
- **Health Checks:**
  - Expose `/api/v1/health` and `/api/v1/health/ready` endpoints, monitored every 30 seconds.
- **Alerting:**
  - Automated alerts for:
    - API latency > target thresholds (95th percentile)
    - Error rate > 0.5%
    - CPU or memory > 85% for > 10 minutes
    - Integration/AI provider failures
- **Log Aggregation:**

    - Centralized logging (e.g., via Winston/Morgan/ELK) for all request/response cycles and error events.
- **Audit Trails:**
    - Track document generation, publishing, and user actions for post-mortem analysis.

---

10. Performance Testing Plan

- **Automated Baseline Tests:**
    - Run `npm run test:performance` after every release; validate all response time and throughput targets.
- **Load Testing:**
    - Simulate 2x expected peak concurrent users (e.g., 200+ API sessions) for 1 hour; system must remain stable, with <1% error rate.
- **Stress Testing:**
    - Incrementally increase load until resource limits are hit; system must degrade gracefully, not fail catastrophically.
- **End-to-End Scenarios:**
    - Test complex workflows (multi-step document generation, integration publishing) under load.
- **Integration Tests:**
    - Validate AI failover, caching effectiveness, and queueing under external provider/API slowness.
- **Reporting:**
    - Generate detailed test reports with metrics, bottleneck identification, and recommendations.

Baseline Requirements Recap

- **API 95th percentile response time ≤ 3s for standard ops**
- **Minimum 100 concurrent API sessions per node**
- **Average CPU ≤ 70%, Memory ≤ 75%**
- **Graceful degradation with error messages under load**

- **Cache hit ratio ≥ 90% for templates**
- **Automated, continuous performance testing and alerting**

Note: All requirements must be reviewed and updated quarterly or upon significant architecture/infrastructure changes.

End of Performance Requirements