Performance Management

Fundamentals of IdentityIQ Implementation

IdentityIQ 7.0

SailPoint

Overview

Performance Management

- Performance Management Approach
- Tools & Resources
- Common Pitfalls to Avoid
- Things You Can Do to Improve Performance

Performance Management Approach

SailPoint

SailPoint Services Approach

- Make performance an integral part of your project
- Don't wait until the end of the project to worry about performance.

Pre-Implementation

- Architecture Evaluation
- Sizing Workshops
- Concurrency Planning
- Example Scenarios

Implementation

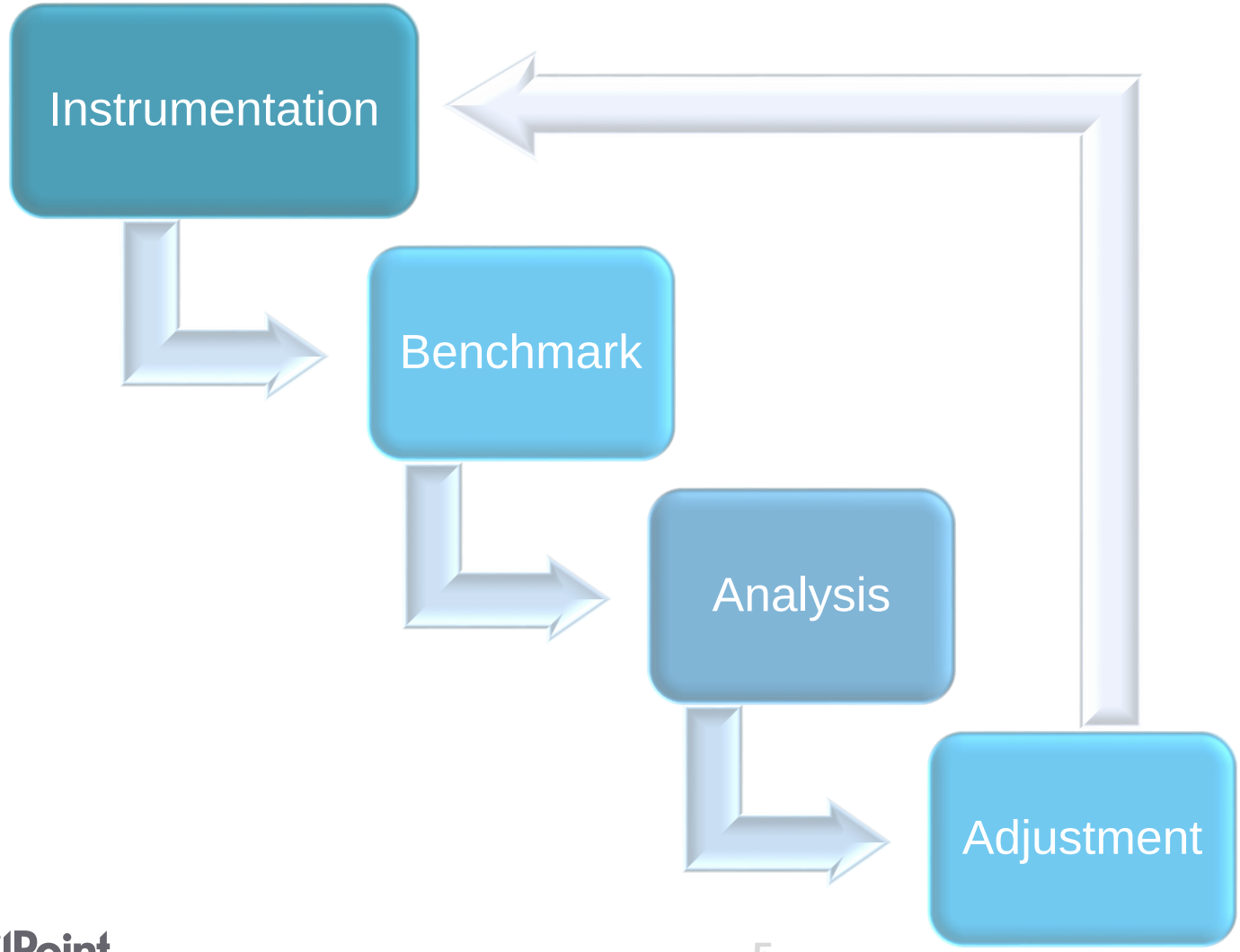
- Performance Deployment Checklist
- Instrumentation
- Benchmarking

Post-Implementation

- Upgrades
- Monitoring
- Future Scale Planning

Performance Management Methodology

Approach



Resources and Tools

SailPoint

Resources and Tools

Compass – Performance Resource Center

- Performance Optimization Checklist
  - Single sheet review list for staff supporting IdentityIQ
- Performance Management Guide
  - Why and how to change things for the better
- IdentityIQ Hardware Sizing Guide
  - How much hardware is needed to support IdentityIQ?
- Partitioning Best Practices 6.2+
- …and more!

Resources and Tools

Expert Assistance

- Performance Assessments
  - Architecture analysis
  - Sizing workshops
  - Performance reviews
    - Benchmark and instrument your installation
    - Perform diagnostics on connectors, DB, application servers
    - Provide formal report and recommendations
- Performance Discussion Group
  - Compass → IdentityIQ Forums → Performance

Performance Monitoring Tools

Measuring

- Log4J timings

  2015-07-01 09:58:17,314 INFO QuartzScheduler_Worker-4 sailpoint.api.Aggregator: 1481 - Aggregation start

  2015-07-01 09:59:24,513 INFO QuartzScheduler_Worker-4 sailpoint.api.Aggregator: 1481 - Aggregation complete

- System out println messages

Performance Monitoring Tools

Measuring Rule Performance

- Meter method
  - Track number of calls
  - Calculate min/max/avg execution time
  - Debug Page → “wrench” → Call Timings

- Example code

```
import sailpoint.api.Meter;
//Begin your code
Meter.enterByName("MyMeter");
//Do work
Meter.exitByName("MyMeter");
//End of code
```

Call Timings

Name	Hits	Errors	Min (ms)	Max (ms)	Total (ms)	Average (ms)
AccountPolicyExecutor:evaluate	960	0	1	125	1948	2
EntitlementCorrelator.analyzeContributingEntitlements	960	0	0	1	31	0
Identitizer.diff detections	960	0	0	1	8	0
MyMeter	18	0	107	194	2697	149

Performance Monitoring Tools

Task Throughput Report

- Analyzes completed tasks
- Provides benchmark to compare IdentityIQ environments
- Helps diagnose and resolve performance bottlenecks
- Download from Compass

Report Result

Report Details

Name	Task Througput Details	Started By	spadmin
Type	Live Report	Started	5/13/15 12:27:24 PM
Description	Displays record throughput statistics and performance based on TaskResults.	Completed	5/13/15 12:27:36 PM
Status	Success	Progress	Download PDF Download CSV

Return To Reports | Edit Report

Report Data

Task Name	Completed Date	Records Processed	Wall Clock Duration	Aggregate R/S	Working Threads	Partitions Time (ms)	R/S/T	Delete Time	D
Aggregate Accounts - HR	4/8/15 9:11 AM	250025	86.949	2875.536	102	4811607	51.962	0	0
Aggregate Accounts - LDAP	4/8/15 10:26 AM	250016	4451.715	56.161	102	442262665	0.565	0	0

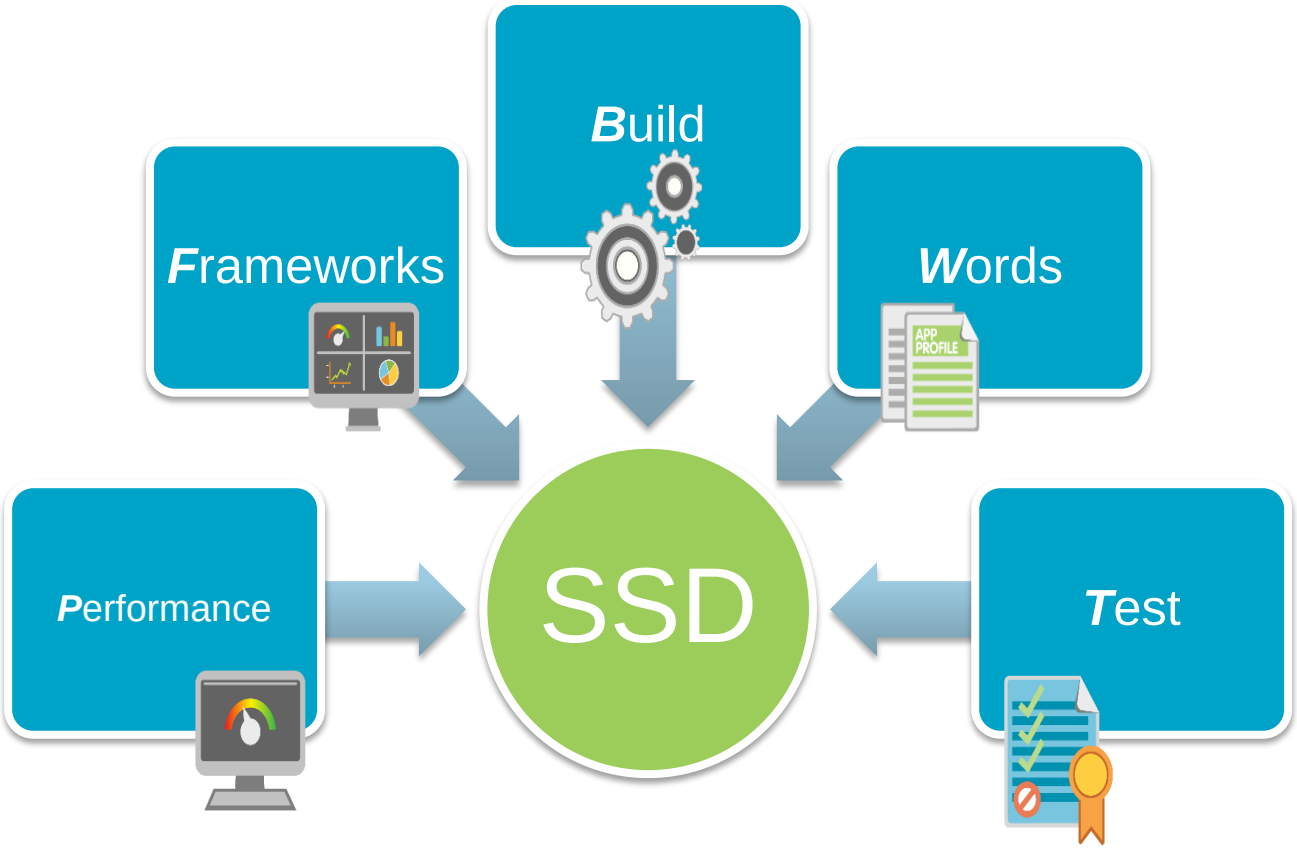
Additional Performance Resources

The Extended Team

- IdentityIQ performance is usually a team effort
- Tuning for optimal performance requires wide expertise
  - Operating systems, databases, networking, Java servlets, J2SE environments, etc.
- Example
  - Database
    - DBA can analyze and tune for performance
  - Application server
    - Admins can monitor threads, connections, etc.
  - Target applications (i.e. LDAP)
    - Admin can analyze and tune for performance

Implementation Tools

Services Standard Deployment (SSD)



Services Standard Deployment

Components

- Services Standard Build (SSB)
  - Build site-specific war files, integrate with change control systems
- Services Standard Framework (SSF)
  - Collection of workflow templates for extending IdentityIQ to address common provisioning use cases
- Services Standard Performance (SSP)
  - Collection of guides and tools used to help measure and modify performance
- Services Standard Words (SSW)
  - Document templates for each phase of the project lifecycle
    - Requirements
    - Functional Design
    - Technical Design
    - Unit Testing guide
    - Go Live
- Available on Compass

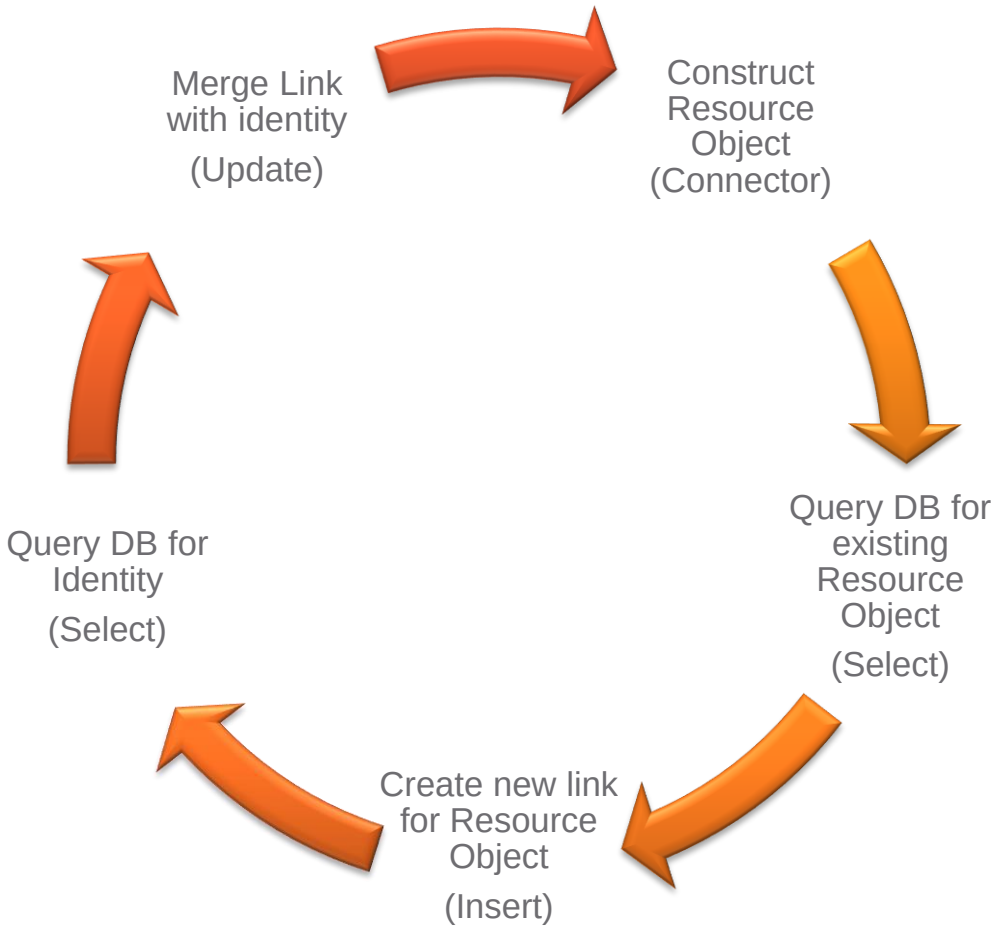
Common Pitfalls

SailPoint

Network, Network, Network

- Traffic between the application and DB server is very sensitive to latency
  - Low Latency connections are best
  - Higher bandwidth connections improve performance
  - Aggregation and certification generations on most network segments should have a .3 ms latency or lower
- Jumbo frames can have a dramatic impact on performance improvement; 1500 MTU is too small for efficient communication in IdentityIQ

Database Example – Aggregation



Virtualization

- Follow VM best practices for applications with large Java heaps
  - Memory reservations
  - CPU reservations
- I/O contention particularly for network resources can represent a severe performance issue
  - Dedicated NICs

Rules/BeanShell

- Lines in rules have a direct correlation to execution time
- Iterative rules have the most impact on performance
  - Move processing out of iterative rules and into non-iterative rules (from Build Map to Pre-Iterate)
  - Use Custom Global or state objects to maintain state during iterations
  - Rules that perform external lookups, joining or other merging of information, will be slow
- Rules that perform correlation or query option searches need to use searchable/indexed attributes
- Rules are more performant than scripts
- No amount of rule tuning can have the impact that responsible rule development can have
- Rule performance profiling is key to a good deployment

Application Server

- Memory, Memory, Memory
- Newer versions of IdentityIQ heavily leverage caching and need memory
- Check hardware against specjbb2005 performance specs, if you aren't getting at least 200K JBO/ps per JVM, performance will be poor
- Tomcat runs with the least amount of overhead, but implement what you know

Database Server

- Watch for I/O contention in shared DB environments
- Ensure that the DB connection pool is adequately sized
  - 50 connections (default)
    - Adequate for sandbox or demos
    - Not typically adequate for production
- Most installations with poor DB performance have typically missed or removed indexes or need additional indexes
- ***Make sure you have the proper and latest JDBC drivers installed (very important!)***
  - See Compass: *JDBC Drivers and IdentityIQ*

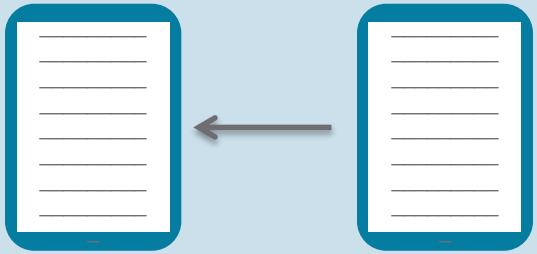
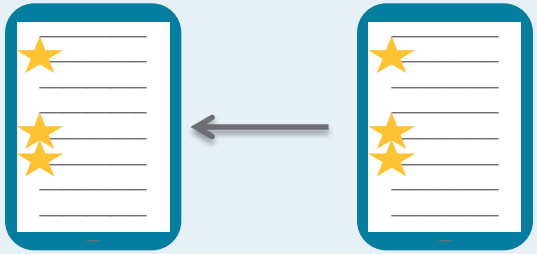
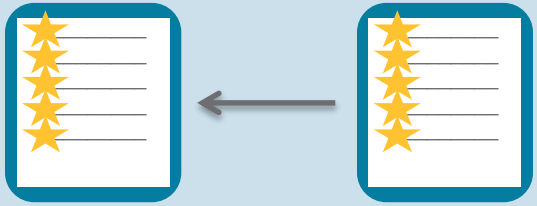
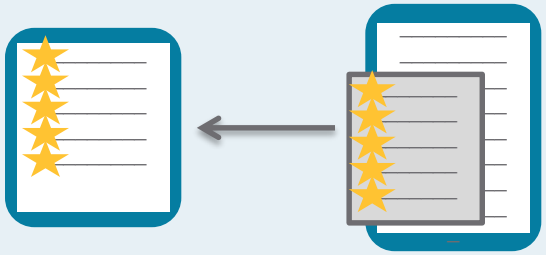
Application Sources

- Delimited and JDBC datasources should be sorted for optimum data load performance
- Network latency can play a major role in aggregation of JDBC data
  - Consider table replication as an alternative
- LDAP servers may need supplemental indexes to deal with aggregation filters
- API sources (SAP, IDM etc.) are best served via a differential (delta aggregation) or alternative (flatfile/JDBC) approach

Things You Can Do to Improve Performance

SailPoint

Aggregation Strategies

	IdentityIQ	Application
Process All • Every account read and processed • Task option ***Disable optimization of unchanged accounts*** = ***true***		
IdentityIQ-based Optimization (default) • Every account read • Only those with changes are processed • Task option ***Disable optimization of unchanged accounts*** = ***false***		
Custom Delta Processing • Manage own change (i.e. write changed accounts to a flat file and process flat file) • Task option ***Detect deleted accounts*** = ***false***		
Connector-based Delta Aggregation* • Read and process only accounts with changes that have taken place after benchmark • lastModData, usnChanged, etc. • Task option ***Enable Delta Aggregation*** = ***true***		

*Not supported by all connectors

Identity Refresh Strategies

- Identity Refresh task can take time due to processing involved
  - Attribute Refresh
  - Role/Entitlement Processing
  - Policy Violation Checking
  - Risk Calculations
  - Provisioning (attribute sync, role provisioning, process events)
- Refresh only Identities that have changed
  - Delta refresh – aggregation marks cubes with changes for refresh processing
  - Controllable with Task options
    - Filters, time stamp, only update modified identities
- Split Refresh into different tasks
  - Attribute Refresh/Role/Entitlement Processing/Provisioning multiple times a day
  - Policy Checking/Risk Scoring at less frequent intervals
- Divide and Conquer
  - Multi-threading
    - Run on *task* servers
  - Partitioning (mutually exclusive to multi-threading)
    - Run on *request* servers
  - Limit number of threads/partitions – context switching will limit performance

Partitioning

Overview

- Partitioning supports splitting up tasks across multiple servers and threads
- Three items can be partitioned
  - Aggregation
    - All at application level (6.3+)
    - Certain Connectors at connector level (6.2+)
      - LDAP, AD, JDBC, Delimited File, PeopleSoft, SAP
  - Identity Refresh Tasks
  - Manager Certification Generation

Server Object

- View Host Status (read only)
  - “Gear” → Global Settings → Host Configuration

Host Configuration

Hosts

Host Name	Request Threads	Task Threads	CPU Load	Last Sync	Status
perftest-task1	0	0		12/01/13 10:21:12 pm	Online
perftest-task2	0	0		12/01/13 10:21:12 pm	Online

- Heartbeat service updates periodically

Partitioning

View Progress – Task Result

Task Result

Details

Name	Refresh Identity Cube	Started By	The Administrator
Type	Identity	Started	12/1/13 10:36:23 PM
Description	Perform a full refresh of all the identities.	Completed	
Status	pending...	Progress	Beginning identity refresh scan with filter: workgroup == false

Return to Tasks

Partitioned Results

Name	Host	Status
Refresh 1 to 31,254	perftest-task1	Refreshing 565 1162
Refresh 125,015 to 156,267	perftest-task2	Refreshing 498 94865
Refresh 156,268 to 187,520	perftest-task2	Refreshing 492 133894
Refresh 187,521 to 218,773	perftest-task2	Refreshing 488 110346
Refresh 218,774 to 250,026	perftest-task2	Refreshing 424 149544
Refresh 31,255 to 62,508	perftest-task1	Refreshing 560 8929
Refresh 62,509 to 93,761	perftest-task1	Refreshing 571 16647
Refresh 93,762 to 125,014	perftest-task1	Refreshing 572 24361

Partitioning

Things to be Aware Of

- Limit partitions to fewer than 50
  - Be aware of diminishing returns when scaling up due to overhead – context switching, etc.
  - 6.4 aggregation exception: limit partitions to fewer than 250
- Watch for contention
  - DB contention
  - Target application contention
- Ensure ability to terminate tasks
  - For each server, if the Task Processor is running, run a Request Processor to ensure ability to terminate task
- Trial and error testing to fine tune is likely
- More information
  - Compass→IdentityIQ Whitepapers→Partitioning Best Practices

Purge Unneeded Database Objects

- Default for object retention is often “forever”
- These can fill up the database and impact performance
- Set “Days Before…” to non-zero values such as
  - Days before snapshot deletion = 365 (1 yr)
  - Days before task results deletion = 90-180
  - Days before certifications are archived = 720 (2 yrs)
  - Days before certification archive deletion = 1080 (3 yrs)
  - Identity request objects =
    - Task: performidentityrequestmaintenance

Performance Bottom Line

- Build performance into your implementation from the beginning
- Use your resources
  - Performance Team
  - Compass
    - Performance Optimization Checklist
    - Performance Management Guide
    - Sizing Calculator
- Build/leverage your team
  - Pull in resources to tune the infrastructure
- You have a lot of control – develop responsibly

Questions?

SailPoint

Putting it All Together

Section 5, Final Exercise

- Debug and resolve problem discovered during user acceptance testing
- Problem discovered in LCM
- Researching will utilize
  - Identity Cubes
  - Entitlement Catalog
  - Data sources
  - … and possibly more
- Resolving will utilize
  - Rules
    - Coding
  - Aggregation
  - Refresh
  - …and possibly more