OPEN ACCESS

Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Review Article

Engineering Resilient ETL: PowerCenter Performance Limits and Cloud Migration Pathways

Venkat Sunil Kumar Indurthy[1]*

[1]Software Developer, Compunnel Software Group Inc, USA

DOI: https://doi.org/10.36348/sjet.2026.v11i02.004 | **Received:** 02.12.2025 | **Accepted:** 26.01.2026 | **Published:** 06.02.2026

***Corresponding author:** Venkat Sunil Kumar Indurthy
Software Developer, Compunnel Software Group Inc, USA

Abstract

The goal of this project is to modernize BP's OPSD2 Downstream Logistics ETL Solution by moving away from Mainframe DB2 database systems and using Oracle-based DBs and incorporating multiple source data sources through Informatica Mappings using an Agile Governance framework for ETL solutions. This study uses InformaticaPowerCenter's Repository-Integration service architecture to analyze the performance characteristics of InformaticaPowerCenter and find that Read Operations can scale efficiently under high levels of concurrency while Write Operations experience rapid degradations in performance due to the use of exclusive locks on certain tables. Therefore, there is a risk of failure if high-volume deployments are attempted using a Write Operation approach. Performance analysis performed using multiple tools have identified that the write latency is the primary cause of the performance constraint. To alleviate this issue, it is recommended that cache tuning, repository clustering and asynchronous bulk logging be used to meet the operational demands of fuel distribution. Looking forward, OPSD2 intends to evolve into BP's Real-time Net-Zero Analytics Platform by leveraging cloud-native solutions based on Informatica IDMC and Advanced Streaming capabilities.

Keywords: Mainframe Modernization, Legacy-to-Cloud Migration, Oracle RDBMS, Informatica PowerCenter, DB2 to Oracle Transition.

INTRODUCTION

BP North America's only downstream application is OPSD2 (Operating People Software of Oil Distribution System 2). OPSD2 originally supported bulk gasoline delivery through large-scale mainframe systems. Major features of OPSD2 include Demand Estimation, coordination of supply orders to meet demand while still meeting inventory targets, and tracking of forecasts. Reports generated by OPSD2 include Material reports, Plant material reports, Contract reports, Inventory reports, and Third-party Pipeline Scheduling reports. OPSD2 was originally developed on a Mainframe computer and contained information stored in DB2, which has now been changed to Oracle to improve scalability. The migration of data from DB2 to Oracle had a number of technical and operational challenges including (a) ensuring data would be compatible through extensive ETL efforts to account for schema differences, (b) resolving performance issues related to real-time querying against mainframe batch processing, and (c) limiting the risk of downtime during the migration to prevent disruptions in the delivery of fuel [1,2].

Compliance with existing systems that do not easily support scalability have difficulty managing real-time environmental conditions due to an inability to forecast demand and manage inventory in real-time, resulting in time delays from one end of the supply chain to the other. In addition, proprietary configurations prevent easy integration of contemporary technology with existing systems and limit flexibility in shipping schedules. Additionally, continued reliance upon obsolete COBOL programming increases operating costs while adding to the risk of downtime in critical logistics networks. Furthermore, the use of batch-processing methods creates reporting delays, making it more difficult to comply with regulatory changes. To address these issues, BP North America has implemented the OPSD2 application dedicated to the mainframe and focused on improving efficiency within the downstream segment of the business by enabling data-driven decision-making, improving shipment scheduling,

monitoring execution and generating required reports relating to forecasts, inventory and contracts [3].

The primary change in OPSD2 is the migration from DB2 to Oracle, which has increased BP's ability to manage data effectively for improved logistics through improved forecasting, scheduling, reporting and using Oracle's enterprise relational database rather than proprietary DB2 storage. The primary advantages of this migration are a more scalable system to support the growth of BP's renewable energy business, improved performance through the execution of real-time query capability and ease of integration with cloud, AI and Enterprise Resource Planning (ERP) Systems. The migration does, however, present several challenges, including data compatibility issues due to differences in schema, the risk of outage during a phased migration and the high level of resources needed to complete the transition, including detailed testing and proficiency with both Oracle PL/SQL and COBOL/DB2 programming languages [4].

Numerous reliable sources can help businesses with their DB2 to Oracle database migration. Oracle's official resources give customers the tools and strategies needed for migrating their data to Oracle Cloud environments. Oracle Migration Center provides detailed Migration Case Studies and Compatibility Matrices that aid clients in migrating from mainframe systems to Oracle databases. Oracle has published a database migration guide designed specifically for Oracle Autonomous Database, which emphasizes optimally performing ETL processes and streamlining your data movement process.

Case studies presented in Daffodil Software documents demonstrate the real-world difficulties and the benefits of significantly improved performance resulting from Logistics Application migrations to Oracle databases. Likewise, case studies from mLogica show how organizations have successfully completed Supply Chain migrations from existing on-premise DB2 environments to Oracle Cloud databases with little or no downtime. Ispirer's tools make it easy for computer programmers to create applications that are automatically converted from COBOL/DB2 into PL/SQL using automatic code generation. In addition, IBM's database migration instructions offer vendor-neutral methods for ensuring application compatibility and data migration from COBOL/DB2 applications to Oracle databases. SAP Community Insights provide insights to the community regarding issues related to Schema Mapping, indexing, and Testing that occur during on-premise migrations.

Modernizing ETL processes allows Oil Companies like BP to solve Supply Chain real-time issues by integrating Next-Generation Analytical Tools with Legacy Mainframe Data Warehousing Projects. This project involved Managing ETL processes used to Create Demand Prediction Spikes for Oil Price increase Decreases in 2022. To Manage this process, we collected user requirements for Bulk Shipping Scheduling through stakeholder workshops and created Data Flow Models to illustrate the flow of data through the Bulk Shipping Schedule. The Technical Design portion of this project consisted of creating Informatica Mappings that Extract Data from different Data Sources. Types of Data Sources include: DB2 Mainframe Log Files; Oracle Inventory Database; SQL Server Forecasting Systems. Conversion of flat files from non-volatile markets to reusable components enabled further ETL transformation of flat file contents to load to Oracle for near-real time reporting [5]. The testing of this process consisted of unit testing conducted within a UAT environment simulating continuous supply of fuels [5].

Incorporation of the Apache Kafka platform into our ETC is a component of OPSD2 evolution through the use of advanced ETL capabilities. By utilizing Kafka to stream real-time Internet-of-Things (IoT) sensor data from BP refineries into Informatica, we are able to achieve near-real-time streaming of events into the Informatica eventhub via Kafka Connect, which greatly enhances our ability to react quickly to changes in the fuel supply chain. This capability significantly reduces prediction latencies and substantially increases our ability to receive demand signals in near-real-time from the sensors in the refinery, particularly in the event of geopolitical unrest. By utilizing the Kafka Streams component to enable hybrid processing, we can filter and aggregate events, allowing us to reschedule shipments prior to experiencing any supply shortfall. This capability reduces costs incurred related to lost sales resultant from fuel supply shortfalls. In addition, enriched events will be sent to Informatica PowerCenter for Transformation of data via, amongst other things, complex event process modelling and transformation. The installation of lightweight Kafka clusters on remote oil and gas rigs allows for efficient data replication from the edge to the cloud, allowing for data to be mirrored at the central site, even during periods of intense use of the network. This capability is especially beneficial during geopolitical instability because Kafka can handle a significant surge in events leading to immediate changes to our inventory levels. Additionally, we have deployed real-time IoT feeds to provide automated ETL alerts to Logistics personnel, enabling identification of pump faults and minimisation of downtime costs [6].

Through near real-time calculation of features from dynamic inputs received via streaming integration, we are able to substantially reduce forecast latency in OPSD2 to enable sub-minute forecast for demand spikes. By using various streaming frameworks such as Apache Flink or Kafka Streams to perform our forecast calculations, we can perform extremely rapid incremental calculations, such as changing moving average calculations every 10 seconds rather than waiting for lengthy timeframes for each forecast.

Transformer features that are available can be cached in low-latency Redis caches, improving our ability to efficiently access them during our ETL processing. These features can also be stored in accordance with industry-standard frameworks such as Tecton or Feast. In addition to providing low-latency access to forecast features, we also employ on-demand computation through vectorized NumPy operations to derive lightweight features, which allows us to achieve sub-50 millisecond responses during critical events. In practical situations, such as when responding to geopolitical events that create surges in demand, the Flink disruption index significantly enhances the speed with which we can re-forecast our inventory. Additionally, the availability of edge-compute features for refinery alerts considerably reduces latency timeframes for proactive shipping.

Related Work

There have been a number of studies published in peer-reviewed journals demonstrating how successful Kafka is at enabling real-time extraction, transformation and loading of sensor data and failure detection within Oil Refineries. One of the more established examples (from 2017 [7]) is an example of a Lambda Architecture, which makes use of Kafka for processing real-time sensor data streams to identify anomalies and forecast demand. Another study analyzed the performance of Kafka within various streaming topology architectures and showed low latencies (less than one second) capable of handling thousands of events per second. This would be particularly useful for hybrid ETL environments that are disrupted during times of supply chain issues. A conference presentation (2021) discussed how Kafka is used within the refinery's sensor pipeline, highlighting the potential of smart grid operations and the ability to reduce ETL Latencies using Change Data Capture (CDC). A 2022 article [6] was published contrasting real-time ETL to Batch Processing, showing Kafka's advantages in managing unexpected spikes in demand globally. Finally, a 2025 document summarizes 98 ptimizing consumer lag and implementing best practice methods for partitioning and replicating data within high-throughput ETL applications with a focus on the monitoring of oil refinery operations.

Current literature discusses Kafka's impact on streaming ETL applications, specifically in low-latency pipelines using Kafka Streams and Kafka Connect, as well as with Flink and Spark in high-throughput scenarios like Fraud Detection and Internet of Things (IoT) Analytics Applications. Additionally, in 2024, it was shown that using Flink creates an average of 25% lower latency when compared to Kafka when processing high-throughput ETL processes, making it ideal for use in refinery sensor data pipelines. A second comparative study shows how Kafka is effectively partitioned to enable rapid transformations using changing data flow for fraud detection ETL. A 2021 review [8] also noted that Kafka provides event-driven ETL pipelines that scale up to millions of events per second. The use of Kafka Connect and Streams for Kappa architecture ETL in healthcare and energy supply chains as an example of what Kafka can do by integrating data in real time at scale [9]. Publication notes best practices for minimizing consumer lag by duplicating data in the production pipeline, which underlines Kafka's contribution to the fault-tolerant ETL. All of the aforementioned are evidence that Kafka has become a key element in ETL development, especially in systems needing near real-time feature engineering.

New research has identified some challenges in employing streaming ETL operations in Kafka. As of 2020-2025, the challenges of scaling up Kafka-based streaming ETL involve consistency and fault tolerance problems. Efficient recovery from failure and restoration of the state of Kafka Streams are not possible because the checkpointing process is lacking. Therefore, when the entire system fails, it results in a significant amount of time lost because all components must be brought back to the same point before proceeding. In addition, Kafka has issues surrounding internal consistency due to the fact that it offers depth-first processing instead of breadth-first processing. Therefore, when processing events that occur at very high rates, Kafka's depth-first processing will not yield real-time results. As stateful applications grow in use, it becomes increasingly difficult to manage the resources required to run those applications. Consumer lag occurs within multi-tenant ETL pipelines because Kafka does not provide an auto-scaling feature. Finally, additional issues arise when late-arriving data must be handled in rapidly changing energy environments, resulting in trade-offs between time and completeness, complicating the effect of event time attribution. The fragmented nature of end-to-end tracing between the multiple components in a hybrid architecture is preventing proper observability and ultimately complicating the debugging of those systems. The associated issues of observability and integration have been the motivators for many researchers looking into the development of hybrid engines (e.g. Flink + Kafka) and hybrid streaming databases that will facilitate building reliable, low latency ETL pipelines for real-time applications such as Oil Logistics [10,11].

The research carried out between 2020 and 2025 demonstrates that the scalability, consistency, and fault tolerance of Kafka-based streaming ETL pipelines continues to be a challenge. Furthermore, the ability to recover from failures and restore application state is severely hampered by the lack of an effective checkpointing mechanism, leading to substantial time wasted recovering from a complete failure of a pipeline. In addition, due to the processing method used in Kafka Streams being depth first, stateful applications will experience internal consistency issues when processing high-velocity events because Kafka Streams has a design that prioritises eventual consistency over strict correctness. As the size and complexity of stateful

applications continue to grow, the management of resources such as topic partitions becomes increasingly difficult. Additionally, the lack of built-in auto-scaling capabilities in Kafka makes it even more difficult for producers and consumers to respond to changing workloads and resource availability, resulting in the lagging behaviour seen in many multi-tenant ETL pipelines [12].

Lastly, due to the dynamic nature of energy consumption in today's world and the fact that we often have to deal with late arriving data, when building ETL pipelines in a hybrid environment, we have to weigh the trade-offs between latency and completeness, creating further complications with event-time attribution. The challenges resulting from the fragmented nature of the end-to-end traces in these hybrid setups make it difficult to diagnose and remedy the issues encountered when debugging these systems. Therefore, many researchers are focusing their efforts on finding ways to build reliable, low-latency ETL pipelines to service real-time applications in hybrid environments such as Oil Logistics [13].

More specifically to the OPD2 real-time ETL workflows, when looking at how scalable Kafka is, some of the key indicators you can look for indicating that Kafka may be struggling with scaling include consumer lag, broker resource utilization, under-replicated partitions, unbalanced partitioning, and throughput drops. When consumer lag reaches levels of 10,000 records or higher, consumers are showing that they are no longer keeping pace with the producers and, therefore, will require either more partitions or increased levels of parallelism. Broker resource utilization metrics such as CPU and disk usage being over 70-80% and long wait times for I/O indicates that these brokers could potentially be overloaded; this is especially true in geo-distributed environments where both network and broker resources may be constrained. If you see frequently changing metrics for under-replicated partitions, you are likely experiencing either failure of brokers or failure of the network and therefore require scaling to keep your partitions balanced. Metrics for unbalanced partitioning may signify hot partitions as well as. The last metric to review is the drop in throughput evidenced by a decrease in the rate of BytesInPerSec / OutPerSec regardless of stable rates of production. This can be indicative of a bottleneck occurring in a controller (either for ZooKeeper or Kraft) [14].

System Architecture

The focus of the design is on the convergence of multiple data sources into an Oracle Data Warehouse (DWH) following Agile principles in support of BP's downstream logistics and utilizing Service Oriented Architecture (SOA) components from InformaticaPowerCenter. The strategy starts with the setup of the PowerCenter Domain to provide a Load Balancing and High Availability solution and then through the Configuration of Integration and Repository Services. The requirements were acquired from workshops and resulted in Logical Data Models and Physical Data Models with a list of prioritized backlog items within the Repository Manager. Data ingestion begins by bringing in Data in different Formats, defining the data in Source Analyzer, and using Filtering and Optimization techniques on the incoming Files. The transformation process is to Create Mappings that can be reused in the Data Processing and Validation Process and set session properties such as session type and session frequency to improve performance. The Workflow Manager orchestrates the Workflows and the Workflows run in a daily Batch Process, with automatic notification of errors via email. The Workflow Monitor monitors and executes the Workflow, Logging Performance and enabling Debugging and Repository management. The multi-environment Test Pipeline includes Unit Tests, End-to-End Integration Tests and Deployment to Production with Rollback options. The ongoing support of Agile in Production Continuously supports the Change Management and the Engagement of stakeholders while maximizing the ability to Support Delivery of Services and developing Scalable ETL Processes from the Legacy Mainframe systems to the Modernization of the DWH via Informatica Power Center as an ETL tool used for Data Processing to transfer Data from Multiple Sources (DB2, Oracle, etc.) to the Target Locations. The mappings and workflows will contain all the integration and Transformation components, within the Service Oriented Architecture (SOA), of the Domain, Repository, and Integration Services, in order to facilitate the ETL Pipeline. The Roadmap is depicted in the accompanying Figure 1.

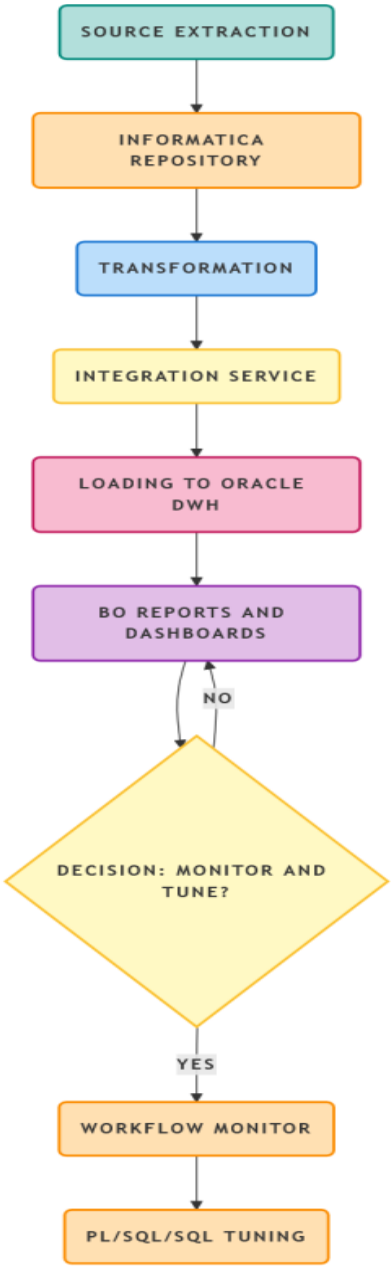

Figure 1: PowerCenter Domain Environment Setup

1. PowerCenter Domain High-availability (HA) Grid Installation and Configuration:
   - Setup Integration Services which allow ETL (Extraction, Transformation, Load) operations and also Repository Service which manages metadata associated with all the Integration Services.

2. Define Source using Designer Source Analyzer:
   - Import Sources that may be from a number of locations i.e.(MS Access, Oracle, SQL Server, DB2).
   - Configure Source Qualifier based on your requirements and need (pushdown and SQL overrides).

3. Create Mappings using Mapping Designer:
   - Create the mappings that are required based on the transformations you are going to include i.e., Joiner, Lookup, Aggregator, Filter/Rank, and Sequence Generator.
   - Validate mappings that you create and reuse transformations whenever possible in other mappings.

4. Oracle Target Definition using the Target Designer:
   - Create fact table(s) for example: FACT_SHIPMENTS in the Oracle Database.
   - You can also optimize the loading process via direct path and/or bulk load.

5. Workflow Manager - Create Sessions using the Workflow Manager:
   - Configure the session task with the Source and Target link(s) and set up pre/post SQL commands.

6. Workflow Management in Workflow Manager:
   - Create Workflows which include the Session task(s), Command task(s), and Email task(s).
   - Schedule workflows and make use of Worklet(s) wherever possible to reduce the amount of duplicative work.

7. Monitoring the Execution of your Workflows:
   - Start the workflow process; the extraction will be completed row-by-row; your transformations will take place in the workflow; after the transformations will be done the records will be loaded into your Target.
   - Monitor the session log(s) as issues arise and debug as necessary.

8. Deploying and Testing of your Workflows:
   - Conduct unit testing of your mappings and make sure that your mappings pass QA before rolling them into production.
   - Use Agile Sprints to assist with production rollout support and to provide organization and support for requests for changes to the product.

Informatica PowerCenter has developed a Service-oriented Architecture (SOA) in its ETL Pipeline called the "OPSD2", consisting of a single domain housing all the services including gateway computers and worker computers for the multiple nodes that run the services across the entire architecture. I hope this has clarified any questions you have with regards to setting up the Informatica PowerCenter environment, creating mappings, and testing. Client tools that connect to and communicate with the core components of the architecture, such as the Service Manager, Workflow Manager, Repository Service, and Integration Service, use the same TCP/IP protocol to communicate with one another. All these components make use of an Oracle Database to manage the actual repository (the repository is maintained in a centralized location).

When using the client tools (e.g., Designer and Workflow Manager), you will authenticate against the Service Manager and establish a connection to the Gateway Node (s) from which you can access metadata over ODBC/JDBC. In addition to maintaining the metadata, the Repository Service also maintains workflows and mapping that support Agile teams working together simultaneously because each of these workflows and mappings has a version. Workflows are executed on the Integration Service based on requests that come from the Service Manager. The Integration Service uses round-robin or least-loaded processing methods to process multiple Data Sources (parallel processing). The Workflow Monitor connects to the Integration Service to get real-time information about the load on that service, while the Admin Console gathers health metric data about the domain and generates structured reports from that data.

The ETL process follows this flow: Mapping is performed in the repository. Data is extracted, transformed, and loaded into the target(s). The additional interactions with the above components include retrieval of metadata, internal domain administration of the Service Manager, and observation of the workflow in real-time. The entire system is designed in a way that allows it to be highly expandable to support the extremely high-volume Data Warehouse activities anticipated by International Teams.

The Integration Service acts as a repository client (in the context of OPSD2) to support the building of a scalable, metadata-driven pipeline and to communicate with the PowerCenter Repository Service using a traditional client-server model. Once the Integration Service is launched, it initiates an authentication request to the Service Manager to obtain a connection to the Repository Service. Once the Service Manager provides the connection information and the route to the Repository Service, the Integration Service and the Repository Service are connected over TCP/IP and are capable of retrieving various types of metadata (i.e., source metadata, transformation metadata, workflow metadata) from the repository databases.

The Integration Service will load the metadata it is provided at execution time into memory and will perform conversion on the data without transmitting any actual data. The Integration Service will manage the source independently from the target. The Integration Service will return statistical information on the execution of the ETL job back to the Repository Service, which allows Agile teams to modify an existing mapping and create multiple versions of that mapping simultaneously. Once the integration process is complete or has failed, the connection will be closed, with the Repository Service remaining the Hub, and the load balancing process distributing the work to the various Integration Service nodes.

Some examples of how the Integration Service utilizes this architecture include using TCP/IP for communication, implementing enforced read/write locks, providing scalability through Shared Repository Services, and providing mechanisms to handle Failures to support continuous ETL Operations without a single point of failure. As such, the Integration Service provides centralized metadata storage and distributed computation.

The Integration Service will connect directly to the Repository Service over TCP/IP to obtain the Metadata it requires to execute ETL Jobs, including

Workflows, Sessions, and Mappings. The Integration Service will call the Domain's Service Manager for every Workflow. The Service Manager Will Direct the Integration Service request to the Repository Service and will provide the Integration Service with the Host and Port information it must use. The Connection will be established as persistent. It will be strictly for the Transmission of MetaData. The Integration Service will use stored Repository Credentials to Authenticate itself to the Repository Service. The Integration Service will request a specific set of MetaData items from the Repository Service, i.e., Workflow Details, Session Features, Mappings, etc. The Repository Service will ensure that all requested MetaData items have been locked and that all Items are consistent when it approves the request by using Locking Mechanisms and searching its underlying Database. The Repository Service serializes the MetaData into a proprietary format, and it transfers the serialized value to memory for efficient access by the Integration Service during processing.

The Integration Service also returns status logs to the Repository Service via the same connection, which are then saved in the database. Once the session is complete, the connection is closed. In grid mode, this Connection Pool is shared between multiple Instances of the Integration Service. This architecture provides several advantages such as: Data consistency with trapping; Clustering is optimised for Performance Improvement; The Integration Service supports robust failure recovery through Failover Mechanisms. This architecture can perform multiple concurrent ETL's on a Global Scale without database contention; thus, separating Storage and Computing.

The main objectives of this project are to optimize the OPSD2 ETL processes with Session Logs and External Monitoring, particularly by leveraging Informatica's ability to monitor the performance of Metadata Retrieval between the Repository Service and the Integration Service. Workflow metrics used to monitor Workflow performance are the "Service Initialization Time" and the "Metadata Fetch Duration". Significant spikes in Fetch Duration (greater than 5 seconds) are indicative of potential bottlenecks within Workflow.

Real-time Monitoring Latency Charts are used to monitor the different latency levels experienced as Data are being Transformed to and from the Integration Service. Verbose Logging is enabled to determine the time it takes to Complete TCP Handshakes and fetch Metadata from the Repository Service, particularly for those requests that exceed 2 seconds. Dashboards within the Admin Console provide detailed metrics related to the Integration Service's "Metadata Cache Hit Ratio" and other metrics captured within the Repository Service. CPU and Memory utilization for all connected Services within the Domain are also monitored. During peak Agile Deployment times, Oracle View Reports and AWR Reports are used to monitor all Queries within the Repository Database and to gather Performance Metrics associated with Integration Services using DTM Process Logs.

For integration with External Tools, we monitor Connection Times and Fetch Errors. Consumer Lag Spikes may be correlated with both TCP Latency and the Serialization Size of Data being loaded. Prometheus is being leveraged to monitor Latency and AppDynamics are used for End-to-End Tracing to decrease Metadata Lookup Times. To maximize the optimization thresholds for the OPSD2 ETL processes, we will Scale the number of Repository Service Nodes to support Higher Concurrent Loads, Tune Cache Size to support Reusable Mapplets and establish Alert Rule Sets for Fetch Failures and Latency during the testing Phases. High volume ETL processes should also have a regular Purge Process for the Repository to maintain Efficient Metadata Retrieval. A summary of recommendations and Optimization Thresholds can be seen in Table 1 on the following page.

Table 1: External Tools Integration

Tool	Key Metrics Monitored	OPSD2 Use Case
PMCMD/Infacmd	Connection Time, Fetch Errors	Script alerts for nightly DWH loads
Splunk/ELK	TCP Latency, Serialization Size	Correlate with consumer lag spikes
Prometheus	Custom Exporter for REP Latency	Grafana dashboards for global teams
AppDynamics	End-to-End Trace (Client→Repo→DTM)	Debug Joiner/Lookup metadata delays

PowerCenter provides metrics to help identify areas where metadata retrieval performance may be impacted through delays from the Repository Service to the Integration Service: Startup Time of Sessions, Duration of Metadata Fetches, and Service Level Response Times. The Startup Time of the Service is defined as the elapsed time from the point at which the Workflow was started until the time when the Metadata is fully loaded, and any delay greater than five seconds is indicative that there may be additional issues. The optimal time for performing Metadata Fetches should be less than 2 seconds for more complex mappings, and the Duration of DTM Process Startup Latencies may see a significant spike for very large Repositories.

The performance of the Latency Indicators can be assessed through the use of Percentiles, so any latency greater than 10 seconds for the P99 Fetches in high volume deployments would be considered an Outlier. If the average Connection Pool Wait Time is 1 second or greater, it may indicate that the Repository Service is overloaded and requires additional resources. The

additional metrics available to support Resource Performance include Oracle V$SQL Statistics which provide measurements regarding the Latencies associated with querying as well as the potential for Database Contention. Similarly, a Low Cache Hit Ratio will also delay the retrieval of Metadata. It is recommended that alerts should be implemented for any P95 Latencies that exceed 3 seconds, or Startup Times that exceed 10 seconds. Also, it is advisable to provide on-going maintenance and perform Cache Size Adjustments to help improve the performance of ETL Workflows.

PowerCenter reads typically have shorter latencies associated with the retrieval of Metadata because of the different Transaction Types and Types of Consistency Checks that occur during Reads and Writes. The general latencies for Reads and Writes have been observed to occur between 100 milliseconds and 500 milliseconds for Reads and 500 milliseconds to 2000 milliseconds for Writes. This difference in latencies is primarily due to the different mechanisms that are employed for Locking on both the Read and Write Operations. For example, when you perform a Read Operation, you will use a "Shared Lock" mechanism which allows multiple services to leverage the same Data instance concurrently. Conversely, a Write Operation will use an "Exclusive Lock," so only 1 User or Service can write to a particular Record or Sequence. This allows for far greater read efficiency while Writes incur a further workload burden due to the use of Triggers and the Audit Trail. Furthermore, Reads are further supported by SQL Indexed Views and Cache. Due to the predominance of Service Contacts being Metadata Fetches, to take advantage of these efficiencies, both Read and Write Thresholds should be monitored with an emphasis on Read Optimization. Tuning Cache can also yield efficiencies in the Daily Load Process of Data Warehouses.

PowerCenter's unique nature as a Shared Access Read Service to Metadata and Cache Storage allows for faster processing as a result of fewer issues associated with Concurrency of Writes (Exclusive Locking Mechanism) as compared to the Latency of Writes. When the number of Concurrent Write Requests surpasses the number of Concurrent Read Requests within the same time period, it results in a larger overall increase in Read Latency and a corresponding increase in Write Latency during peak usage. Shared Locking will allow Reads to complete successfully with Low Contention, but during High User Contention, TCP Queues will be backed up causing delays to Reads. Conversely, Writes will always be Serial when Low User Contention exists due to the additional overhead associated with Redo Logging and TCP Contention. Excessive Writing to the Database will cause spikes of Excessive Write Latency during Peak Usage times, while at the same time Read Times remain acceptable but experience delays when Writes occur, resulting in Stale Reports. A strategic approach to alleviate these issues is to create a Repository Service Cache that has a high Hit Rate/cached size in combination with a Batch Processing methodology to write Sessions. The architecture that is created will provide a balanced environment in which Reads can be effectively achieved with a high volume of Concurrent Requests, while providing for a limited quantity of Writes to eliminate potential failures. Additionally, the use of Regionally Dispersed Repository Services to provide access for Multi-Region Operations can also improve overall Processing performance as outlined below.

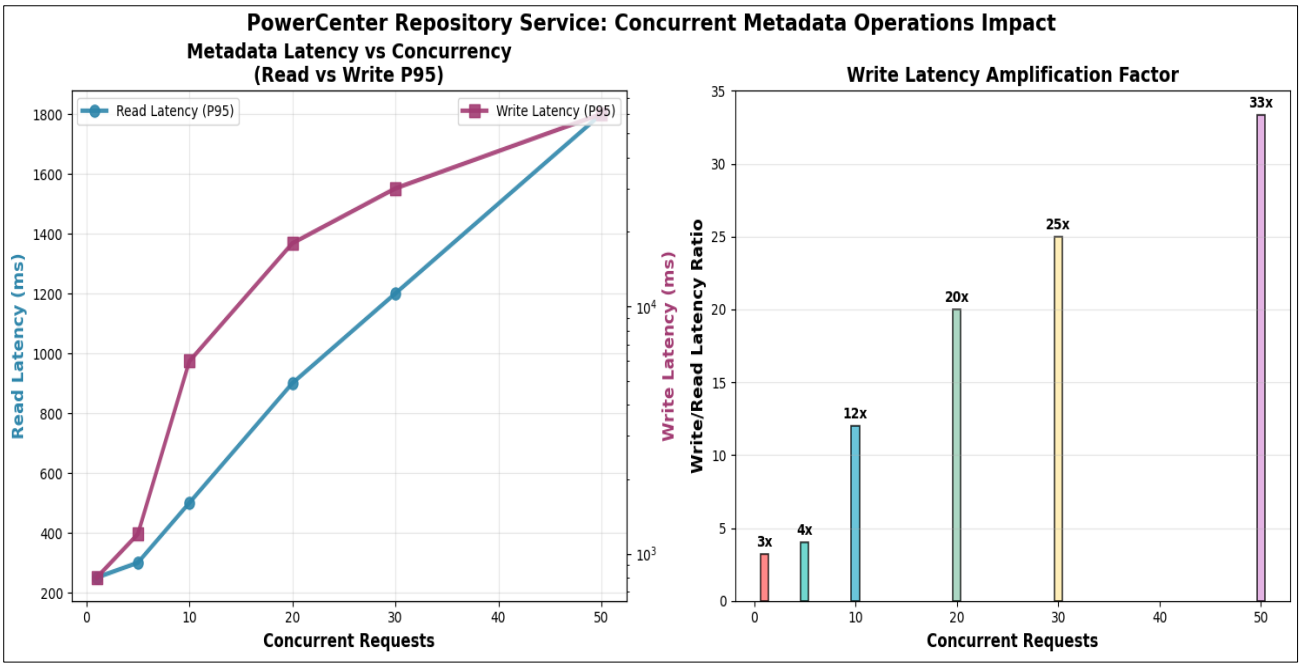

Figure 2: PowerCenter Repository Service: Concurrent Metadata Operations Impact

CONCLUSION

PowerCenter's Repository-Integration Service architecture experiences a bottleneck in terms of scalability due to the way exclusive locks on OPB_SESSLOG tables are used when multiple users create concurrent records, thus creating significant delays in write operations. In contrast, when multiple users access the system via read operations, PowerCenter can scale via the use of shared locks and read caching. When records are concurrently written to (due to) concurrent operations on the same tables, this process creates an additional hurdle for ETL. Current Repository-Integration Service architecture and design is capable of providing optimal integration services to support logistics at BP North America. However, as Agile teams continue to grow and real time operations continue to expand, current designs may become less scalable to support those demands. Therefore, further enhancements to Repository-Integration Service architecture will be necessary to support optimal levels of integration service delivery to internal and external customers. Specifically, plans include implementing migrating all data processing to Informatica IDMC for enhanced levels of concurrent processing and elastic scaling; implementing a hybrid architecture for streaming and batch processing to reduce repository hits during high volume requests; implementing an observability-first design to improve monitoring and predictive scaling; and utilizing GenAI technology for effective metadata management. Collectively, these key initiatives will help transition OPSD2 toward a more agile and responsive analytics platform, which supports BP's overall strategy of achieving net-zero emissions by 2030.

REFERENCES

1. "Challenges in Downstream Oil & Gas Distribution", Lasse Jiborn, April 2022, https://www.amcsgroup.com/resources/blogs/challenges-in-downstream-oil-gas-distribution/.
2. "Solutions in oil downstream logistics", December 28, 2023, https://inspenet.com/en/articulo/logistics-downstream-oil/.
3. "Oil and Gas Major Retires Mainframe After Upstream Downstream Split", Amit Garag, Jul 01, 2021, https://www.scribd.com/document/513778639/Oil-and-Gas-Major-Retires-Mainframe-After-Upstream-Downstream-Split.
4. "DB2 to Oracle On-premise Migration challenges", 2020 Dec 16, https://community.sap.com/t5/technology-q-a/db2-to-oracle-on-premise-migration-challenges/qaq-p/12310862.
5. "How to Integrate Data in the Oil and Gas Industry", September 25, 2023, https://dataforest.ai/blog/how-to-integrate-data-in-the-oil-and-gas-industry.
6. "Building Real-Time ETL Pipelines with Apache Kafka", Stefan Sprenger, February 11, 2022, https://datacater.io/blog/2022-02-11/etl-pipeline-with-apache-kafka.html.
7. "Cloud–based Fault Detection and Classification for Oil &Gas Industry", Athar Khodabakhsh, Ismail Ari, Mustafa Bakir, 11 May 2017, https://arxiv.org/pdf/1705.04583.
8. "Literature Review: A Comparative Study of Real Time Streaming Technologies and Apache Kafka", Shubham Vyas, Dr Rajesh Kumar Tyagi, DrCharu Jain, Dr Shashank Sahu, 2021, https://cict21.bmiet.net/proceed/pdfs/CCICT2021-5zzaqZBA554QZHi78VvEEj/239200a146/239200a146.pdf.
9. "Streaming ETL with Apache Kafka in the Healthcare Industry", Kai Waehner, April 2022, https://www.kai-waehner.de/blog/2022/04/01/streaming-etl-with-apache-kafka-healthcare-pharma-industry/.
10. "Migrating Batch ETL to Stream Processing: A Netflix Case Study with Kafka and Flink", Daniel Bryant, Feb 08, 2018, https://www.infoq.com/articles/netflix-migrating-stream-processing/.
11. "Top 3 Kafka Streams Challenges", Apr 11, 2024, https://www.voltactivedata.com/blog/2024/04/top-3-kafka-streams-challenges/.
12. "Scaling Kafka for High-Throughput Data Pipelines: Techniques and Tools", Ashnik Team, Feb 27, 2025, https://www.ashnik.com/scaling-kafka-high-throughput/.
13. "Best practices for scaling Apache Kafka", Tony Mancill, Jan 24, 2024, https://newrelic.com/blog/best-practices/kafka-best-practices.
14. "Kafka performance metrics: How to monitor", Evan Mouzakitis, David M. Lentz, Apr 6, 2016, https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics/.