Irion EDM
In-Depth

Irion®

Giovanni Scavino

Come ridurre tempi e costi dei progetti di Enterprise Data Management

Indice

L’autore

Giovanni Scavino

In qualità di cofondatore Giovanni ha avuto l’idea di creare la piattaforma Irion EDM®.

Dopo gli studi di ingegneria elettronica ad indirizzo informatico, a soli 22 anni, insieme al fratello Alberto, ha fondato una prima azienda produttrice di software, Dianos. Aveva già alle spalle una carriera di successo nello sviluppo di software finanziari (sistemi di front e middle office, Risk & Performance Management, Master Data Management) quando ha deciso, nel 2004, di avviare, sempre con il fratello e con Mauro Sturaro, una nuova iniziativa imprenditoriale, avvalendosi della grande esperienza accumulata in termini di Enterprise Data Management, qualità dei dati, organizzazione dei processi, view di prodotto e posizionamento strategico d’impresa.

Giovanni si è quindi concentrato sull’implementazione della piattaforma Irion EDM®, con l’obiettivo di riuscire a rispondere sempre in anticipo alle esigenze del mercato, cogliendo le priorità più urgenti e agendo di conseguenza.

La sua esperienza e la sua competenza hanno garantito a Irion di crescere rapidamente, affermandosi come un’eccellenza di successo a livello locale e internazionale.

Giovanni Scavino
Irion Chairman of the Board

Come ridurre tempi e costi dei progetti di Enterprise Data Management

Introduzione

Nel complesso e variegato mondo della misurazione delle prestazioni associate alle elaborazioni svolte da un determinato sistema software occorre, come prima cosa, chiarire e concordare gli indicatori di performance specifici a cui si fa riferimento.

I due principali KPI sono il **Response Time**, il tempo che uno specifico processo di elaborazione impiega per fornire gli output dopo che sono stati resi disponibili gli input, e il **System Throughput**, la quantità di lavoro che un dato sistema di elaborazione (hw+sw) è in grado di svolgere in una unità di tempo.

Entrambi gli indicatori sono ovviamente interessanti da monitorare e sono certamente correlati.

Tuttavia, in un contesto di Enterprise Data Management dove le operazioni sono quasi sempre **data intensive** e molto spesso di tipo **massivo ed analitico** (OLAP) piuttosto che "atomico e transazionale" (OLTP), occorre ottimizzare la quantità di lavoro svolto dal sistema in ragione delle risorse hardware (tipicamente CPU, RAM, I/O) messe a disposizione.

Veniamo ad un esempio concreto: in un sistema di controllo della qualità dei dati alimentanti o già presenti in un data lake o in un data warehouse, risulta abbastanza ovvio che ciò che più conta non è tanto il tempo impiegato ad eseguire un singolo controllo, quanto il numero di controlli che il sistema riesce ad effettuare in un'unità di tempo o, ancora più pragmaticamente, il tempo complessivo che impiega ad effettuare tutti i controlli necessari a garantire la qualità delle informazioni dopo ogni specifico ciclo di aggiornamento.

Chiarito quindi che il parametro di riferimento più rilevante da monitorare è il Throughput complessivo del sistema così come precedentemente definito, è anche importante sottolineare che i moderni sistemi di EDM si configurano come un'architettura complessa, spesso multi-tier e multilayer, scalabile sia orizzontalmente che verticalmente.

È composta da servizi e processi che implementano workflow operativi sia automatici che interallacciati a "human task" e agiscono su volumi molto significativi di dati spesso organizzati nei formati più eterogenei.

In questi scenari le variabili e le scelte che influiscono, anche significativamente, sulle prestazioni sono molte e a più livelli.

La tecnologia di Enterprise Data Management di Irion non nasce in vitro, non è il frutto astratto di qualche ragionamento accademico o di laboratorio. Anni di esperienza in contesti mission critical e data intensive ci hanno insegnato molte cose. La nostra piattaforma si sviluppa con l'esperienza, o meglio, ragionando sull'esperienza, partendo dalla **concretezza di centinaia di progetti veri**; quelli con tanti dati, con vincoli di tempo, con le specifiche che cambiano in corso d'opera, con problemi complessi in contesti architetturali e organizzativi molto articolati.

Di seguito analizziamo alcune caratteristiche specifiche che descrivono Irion EDM®, lo differenziano sostanzialmente dalle altre proposte di mercato e rendono ragione delle **performance eccezionali** che abbiamo ottenuto in tantissimi scenari reali indirizzati. In particolare ci soffermeremo su una descrizione dettagliata delle specificità del modello di elaborazione implementato e sulle possibilità di scaling e utilizzo massivo parallelo delle risorse che l'architettura progettata abilita.

Impostazione logica e concettuale delle esecuzioni

Le principali caratteristiche che concorrono a definire le prestazioni del sistema sono riconducibili all'impostazione logica e concettuale delle esecuzioni. In sostanza, come si può facilmente intuire, il disegno logico-applicativo e tecnico a basso livello delle modalità di svolgimento di un'elaborazione è quello che incide in modo più significativo sulla efficienza del sistema. Da questo punto di vista è importante sottolineare che il nostro approccio è **disruptive**; Irion EDM® non è l'ennesimo sistema di ETL procedurale con qualche funzione in più, qualche connettore specifico, un editor più bello e motori magici per l'esecuzione veloce.

Abbiamo ridefinito il concetto di Data Management di cui i processi ETL sono una declinazione, ma che abbraccia anche molti altri ambiti (sistemi Rule-Based, Data Quality, Data Integration e Reporting, Data Governance etc).

Abbiamo preso spunto dal concetto dichiarativo storicamente applicato ad una categoria di linguaggi di programmazione e lo abbiamo trasformato in un paradigma originale che copre l'intera sfera delle pratiche di Data Management, rendendolo pervasivo nella nostra tecnologia e nelle nostre metodologie.

Una delle nostre principali tecnologie prende il nome DELT™, un acronimo che significa Declarative Extraction Loading & Transformation e rappresenta proprio il fatto che, oltre ad invertire le fasi di caricamento e trasformazione (ELT rispetto a ETL), cosa che trova già ampie argomentazioni in letteratura, abbiamo soprattutto fatto in modo che tutto il processo sia descritto e svolto in conformità ad un modello dichiarativo.

Ricordiamo qui in sintesi che per **approccio dichiarativo** intendiamo che non occorre specificare come un dato compito debba essere fatto, ma è sufficiente limitarsi a dichiarare cosa deve essere fatto!

Per un approfondimento e un confronto con il modello procedurale tipico degli ETL tradizionali rimandiamo all'articolo Declarative Thinking.

Gli aspetti più rilevanti, in particolare da un punto di vista prestazionale, a fondamento della nostra architettura tecnologica e del nostro approccio dichiarativo alle elaborazioni sono i seguenti:

- Ogni dataset utilizzato nell'elaborazione **viene riesposto virtualmente come se fosse una tabella** (o un insieme di tabelle)

nel modello relazionale secondo una tecnica che definiamo **EasT®** (Everything as a Table). E' compito della piattaforma operare implicitamente **tutte le trasformazioni necessarie** a far sì che un insieme di dati disponibile in qualsiasi formato (file CSV, Excel, XML, Cobol, DB, Web Services, API ecc.) sia mappato opportunamente. Le tabelle virtuali temporanee EasT® definite durante un processo di elaborazione sono **immutable object** cioè inizializzate una sola volta ed il loro contenuto non può essere modificato dopo che sono state create. Questa caratteristica garantisce coerenza e controllabilità al sistema. Inoltre, il fatto di aver normalizzato tutto secondo un modello relazionale, consente di adottare e far leva su tecniche e motori di Data Management consolidati e iper-ottimizzati negli anni.

- L'intero processo di elaborazione (trasformazione, controllo, aggregazione, analisi ecc.) **viene scomposto in un insieme di Data Engine**, cioè di motori in grado di ricevere in ingresso una o *n* tabelle virtuali, di operare le opportune trasformazioni/verifiche e di produrre in output una o *m* tabelle virtuali secondo il modello EasT®.
- **I Data Engine possono essere di tante tipologie** (Query, Script, Rule, R, Python, Masking, Profiling ecc.). Eseguono elaborazioni semplici o molto complesse secondo quanto configurato da chi imposta la soluzione, incapsulano logiche e modelli varie, utilizzano i linguaggi e le tecnologie migliori e più efficienti in ragione del compito da svolgere. Se necessario ogni motore (es. R, Python ecc.) può utilizzare librerie esterne di calcolo e/o ingaggiare elaborazioni remote.
- **La configurazione dei Data Engine**, degli accessi ad altri oggetti EasT® e della produzione di tabelle virtuali in output **comporta l'implicita definizione di un grafo orientato** di dipendenze di esecuzione. Il sistema è quindi autonomamente in grado di organizzare l'insieme di passi necessari a produrre ogni output e non è richiesto al designer di descrivere l'algoritmo.
- Le varie tabelle virtuali prodotte come step intermedio delle elaborazioni **possono essere opzionalmente materializzate** in modo da produrre dati intermedi eventualmente utilizzabili più volte, scomporre il processo in step parziali più efficienti e fornire al sistema stime corrette sulla cardinalità e la distribuzione statistica dei dati in modo da ottimizzare al meglio le elaborazioni successive. Se necessario possono anche essere attivate tecniche evolute di indicizzazione, sia row-based che colonnari, e algoritmi di compressione dei dati al fine di ottimizzare gli accessi successivi.

- **L'elaborazione DELT™ segue uno schema dichiarativo Goal Driven** e **Backward Chaining**: vengono specificati i dataset attesi in output ed è compito del sistema rilevare automaticamente il **DELT™ Graph** in ragione delle dipendenze implicite, verificare che il grafo sia aciclico e avviare l'esecuzione ottimizzata e parallela di tutti i passi necessari per produrre il risultato richiesto. L'algoritmo di risoluzione ed attraversamento del DELT™ Graph fa anche uso di una serie di euristiche e regole specifiche di ottimizzazione per definire in che ordine e con quale livello di concorrenza è più opportuno avviare i vari nodi del grafo.
- In casi molto complessi il motore, grazie ad un modello fortemente **metadata-driven**, **è anche in grado di modificare attivamente il grafo** durante l'esecuzione in funzione delle proprietà dinamiche che lo determinano e che possono essere influenzate da passi precedenti (Dynamic Properties) o, addirittura, generare ed eseguire runtime i Data Engine necessari per specifiche elaborazioni configurate ed ottimizzate con riferimento ai parametri ed ai dati coinvolti nella specifica esecuzione (VEG Virtual Engine Generator).
- Se necessario una DELT™ Execution **può ingaggiare altre DELT™ Execution** a fini di incapsulamento, ottimizzazione e riusabilità della business logic.
- La tecnologia proprietaria utilizzata per rendere disponibile alle DELT™ Execution i dati coinvolti (gli oggetti EasT® e tutti i dati temporanei necessari) in uno spazio isolato e gestito **si chiama IsolData™**. Si tratta di una sorta di **sandbox**: uno spazio di lavoro virtualmente illimitato, dinamicamente allocato e liberato dal sistema, segregato in termini di permessi di accesso e spazio dei nomi, volatile o persistente.
- **Gli IsolData™ consentono di eseguire n DELT™ parallele**, sugli stessi dati o su dati completamente differenti e/o con parametri, regole e logiche diverse, senza che debba essere predefinita o gestita alcuna struttura particolare (es. non è necessario definire e rimuovere tabelle temporanee, spazi, script, ecc.); tutte le operazioni di supporto alla gestione infrastrutturale sono coordinate e gestite dal framework in modo ottimizzato e trasparente all'utilizzatore. Grazie agli IsolData™ il sistema minimizza la "contention" fra le varie elaborazioni parallele avviate e consente di sfruttare al massimo le risorse hardware disponibili. Dato che le tabelle temporanee contenute in un'IsolData™ sono "immutable object", ovvero vengono inizializzate una sola volta e lette molte volte senza che possano essere modificate, l'accesso a questi dataset avviene in modalità NoLock e vengono prodotte

minimizzando il Log. Le attività di "housekeeping" del sistema sono eseguite in modo centralizzato ed asincrono per non impattare sui tempi di risposta delle elaborazioni.

- Durante un'esecuzione DELT™ **tutte le elaborazioni sui dati sono eseguite a basso livello sul server** e secondo la modalità "set oriented" ovvero considerando un insieme complesso a piacere di dataset contemporaneamente e non "una riga alla volta". In definitiva risulta ovviamente molto più performante "spostare le regole che non i dati", ingegnerizzare il processo e minimizzare le operazioni svolte dal sistema.
- Il modello dichiarativo di rappresentazione di un processo di elaborazione dati **consente al motore DELT™ di scegliere automaticamente e "run-time"**, in base alle statistiche attuali dei dati, quale dovrà essere l'algoritmo ottimale e il livello di parallelismo opportuno da porre in essere.
- Al contrario, un modello procedurale costringerà a prevedere "design-time" tutti gli step e l'eventuale parallelismo da avviare, a prescindere dall'effettiva distribuzione dei dati che si incontrerà durante le differenti esecuzioni.
- Il motore di regole di controllo e trasformazione presente nel sistema **è in grado di generare dinamicamente le istruzioni** necessarie ad eseguire in parallelo e a basso livello tutte le regole afferenti ad uno stesso dataset evitando in questo modo di doverlo leggere ed analizzare più volte.
- Il server **minimizza le onerose operazioni di data movement** ed eventuali elaborazioni complesse che producono dati intermedi da utilizzarsi molte volte (aggregati parziali, lookup ecc.). Infatti sono eseguite una sola volta senza la necessità di salvarle esplicitamente in file o tabelle di appoggio.

La maggior parte delle elaborazioni dei dati avvengono "server to server", senza la necessità di trasferire i dati ad un client perché siano elaborati e successivamente riscritti sul server stesso.

Quando si rendesse necessario passare comunque per un elaborazione client-side, opportune tecniche di pipelining con buffer circolari in memory e bulk-load in streaming minimizzano i tempi di elaborazione ed attesa.

Se necessario una parte delle logiche di elaborazione possono essere direttamente demandate alla sorgente dati (tecniche di "push-down") in

modo da minimizzare il trasferimento dei dati (es. filtri, preaggregazioni ecc.)

In determinate situazioni il sistema è anche predisposto per utilizzare, in maniera trasparente, delle "local cache" dei dati in modo da minimizzare il costo di accesso e disaccoppiare le sorgenti dati, eliminando gli impatti su sistemi esterni e superando eventuali vincoli temporali alla disponibilità dei dati.

- La conoscenza di una serie di statistiche sui dati consente al motore di **operare specifiche ottimizzazioni durante l'esecuzione** dei motori. Per esempio, nel caso di tabelle con una sola riga il sistema è in grado di utilizzare i valori attuali del record in modo da sfruttare meglio le informazioni eventualmente disponibili sulla distribuzione dei dati stessi e decidere ad es. quali strategie di lettura dei dati adottare (i.e. Scan vs. Seek). Nei casi opposti di tabelle con un significativo numero di righe, il motore di ottimizzazione cerca di indirizzare l'utilizzo del "batch-mode" per gli operatori di risoluzione delle query; in questo modo ogni operatore, anche a bassissimo livello, opera contemporaneamente su un set di dati con l'ausilio di specifiche operazioni vettoriali messe a disposizione dalle CPU.

In determinati casi le prestazioni ottenibili adottando questi accorgimenti cambiano anche di un ordine di grandezza.

Conclusione

L'approccio dichiarativo all'Enterprise Data Management, le tecniche e gli algoritmi "fine-tuned", i livelli di scalabilità sia orizzontale che verticale, l'efficiente utilizzo delle risorse, l'orchestrazione e l'ottimizzazione cost-based del grado di parallelismo rendono il throughput complessivo del sistema assolutamente inconfrontabile con gli approcci tradizionali ETL-based.

Cos'è Irion EDM®

Overview e value proposition

Irion EDM® permette di realizzare rapidamente applicazioni per l'Enterprise Data Management. Basata su un innovativo paradigma dichiarativo e con funzionalità end-to-end, è una piattaforma potente e aperta che offre **prestazioni e scalabilità**, usufruibile sia in cloud che on premises.

Le Data App create con Irion EDM® possono coprire e governare **tutte le fasi** del processo di Data Management, compresi mapping, acquisizione e profiling, normalizzazione e trasformazione, validazione e arricchimento, auditing, masking, editing, persistenza e pubblicazione dei dati. La piattaforma è nativamente e interamente **metadata-driven**, garantisce la governance delle soluzioni e consente di sviluppare **interfacce utente specializzate** per adattarsi alle esigenze specifiche dell'azienda.

Approccio e tecnologie distintive

Grazie all'approccio **dichiarativo**, agile e intuitivo, l'utente può concentrarsi esclusivamente sul dichiarare ciò che desidera ottenere: Irion EDM® si occupa integralmente dell'esecuzione del processo, della selezione dei passaggi da intraprendere e della costruzione dinamica degli oggetti e delle strutture necessarie. L'approccio dichiarativo **semplifica** al massimo il modo in cui gli utenti interagiscono con il mondo dei dati e rende l'esperienza più efficiente e accessibile.

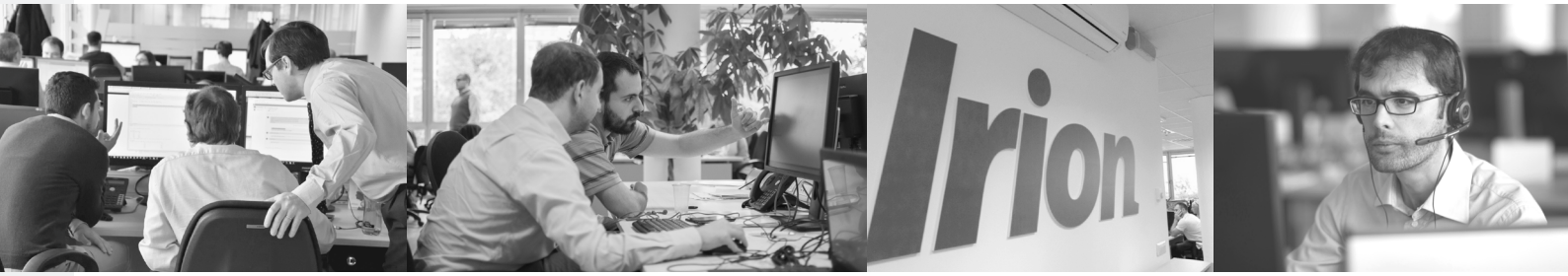
L'approccio dichiarativo è supportato da **tecnologie proprietarie** che trasformano radicalmente il modo di concepire il data management. **DELT®, EasT®, Isoldata®**, sono solo alcune delle innovazioni di Irion EDM®, progettata per **far convergere** nella stessa applicazione tutte le **capabilities** necessarie per rispondere ai requirements in ambito **Data & Analytics**. Supporta i moderni pattern e **design concept**, tra cui Data Fabric (cfr. Gartner®), DataOps e Data Mesh, consentendo alle aziende di adottare approcci all'avanguardia nella gestione dei dati e nell'analisi.

Augmented e versatile

Irion EDM ® è strutturalmente pensata in ottica Augmented Data Management: grazie all'integrazione nativa con R e Python, consente di introdurre logiche di calcolo basate su modelli di Machine Learning & Artificial Intelligence nei processi a tutti i livelli. Così è possibile sfruttare la sua potenza per arricchire e migliorare i processi di gestione dei dati in tutta l'organizzazione.

Irion EDM ® è versatile e può essere applicato con successo in qualsiasi settore e su qualunque processo data intensive: gli ambiti di applicazione che la piattaforma indirizza sono molteplici e coprono diverse industry di mercato.

La piattaforma supporta i principali "Data Journey" che indirizzano e attraversano le diverse discipline del Data Management: ad esempio Data Governance, Data Integration, Data Quality, Data Reconciliation, Data Aggregation & Reporting, Data Privacy & Masking.



Scopri di più!

Se desideri approfondire la Data Governance e l'approccio dichiarativo alla gestione dei dati, scarica dal sito web irion-edm.com gli ebook gratuiti su ***Data Governance***, ***Regulatory Reporting e Irion EDM® In-Depth***.

Sul sito irion-edm.com troverai inoltre tutte le novità
e le soluzioni Irion sui seguenti argomenti:

Data Governance - Data Quality

Metadata Management - Data Preparation

Data Analytics & Reporting

Master Data Management

...e molto altro!

Irion EDM®

La piattaforma
dichiarativa End-to-end
e Data Fabric ready
che ti aiuta a estrarre
valore dai tuoi dati.

PER SAPERNE DI PIÙ

Irion®

GO BEYOND,
BE DECLARATIVE.

www.irion-edm.com/it/

#MadeinItaly