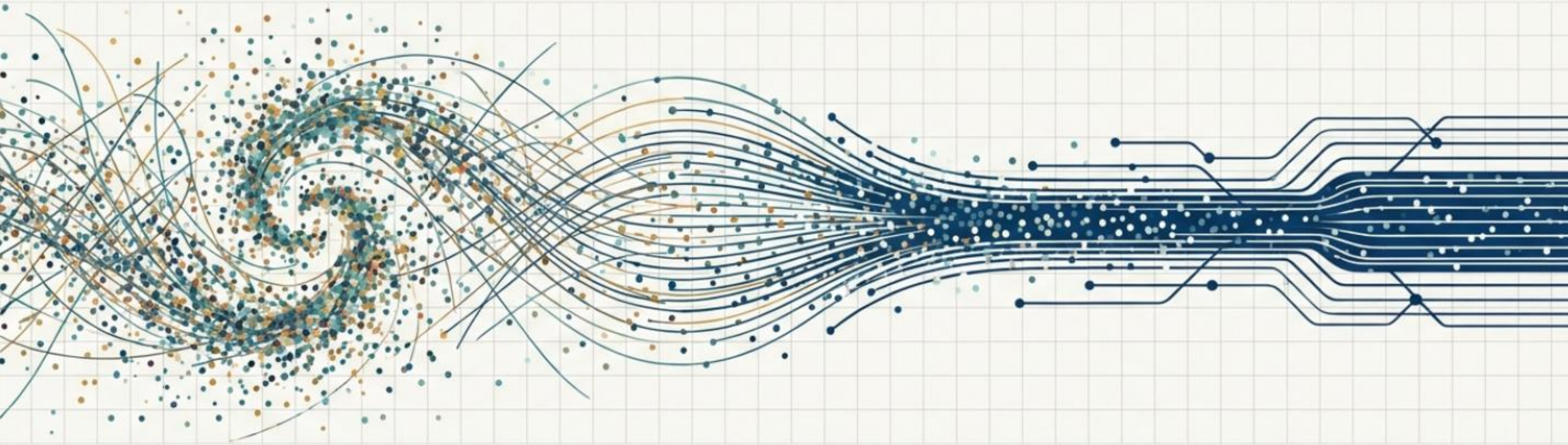
The Architect’s Journey: Principles of Modern Stream Processing

From Infinite Data to Real-Time Insight

المعالجة التدفقيّة - معالجة البيانات بشكل مستمر وفوري عند وصولها



بيانات الحساسات (Sensors / IoT) مثال: حساسات الطقس ترسل بيانات كل ثانية. // درجة الحرارة، الضغط، الرطوبة.

In an era of IoT, AI, and constant connectivity, data is no longer a static resource to be warehoused. It's a continuous, infinite stream. Our challenge as architects is to harness this flow in real-time. This deck is a journey through the core questions and architectural decisions required to build robust, intelligent streaming systems.

ما هو تدفق البيانات؟ جوهر البيانات اللامحدودة

تدفق البيانات هو تجريد لمجموعة بيانات غير محدودة (unbounded dataset). تُرسل البيانات بشكل مستمر في حزم صغيرة من المصدر إلى الوجهة.

قابل لإعادة التشغيل (Replayable):
يمكن إعادة قراءة التدفق من نقطة بداية محددة.

لامحدود (Unbounded):
مجموعة بيانات لا نهائية ومتزايدة باستمرار.

مرتب (Ordered):
تصل سجلات البيانات بتسلسل زمني محدد.

عام (Generic):
يحدد خصائص نقل البيانات، وليس نوع البيانات نفسه؛ يمكن استخدام أي أو نوع من البيانات.

غير قابل للتغيير (Immutable):
بمجرد إنشاء حدث في التدفق، لا يمكن تغييره.

من أين تأتي التدفقات؟ المصادر والأنواع

الأنواع الأساسية

تدفقات الفيديو والصوت
(Video & Audio Streams)

تدفقات الاتصالات
(Communication Streams)

تدفقات البيانات والبيانات الوصفية
(Data & Metadata Streams)

تدفقات الأحداث
(Event Streams)

المصادر الشائعة

- سجلات أنظمة تخطيط موارد المؤسسات (ERP Logs)
- معاملات التجارة الإلكترونية (Views, Orders, Baskets)
- تتبع الأحداث في تطبيقات الجوال
- بيانات الموقع الجغرافي من تطبيقات الويب
- معاملات بطاقات الائتمان
- أحداث أجهزة الاستشعار (IoT)
- بيانات تداول الأسهم
- أحداث الشبكة

نماذج المعالجة: من الدفعات إلى اللحظة الحالية

الطلب والاستجابة (Request-Response)

الوصف: نموذج متزامن (Blocking) يعالج الطلبات الفردية.

زمن الاستجابة: منخفض جدًا (مللي ثانية).

الإنتاجية: مصمم للمعاملات الفردية.

حالة الاستخدام: أنظمة معالجة المعاملات عبر الإنترنت (OLTP) مثل معالجة بطاقات الائتمان.

معالجة التدفقات (Stream Processing)

الوصف: نموذج غير متزامن (non-blocking) تم للمعالجة المستمرة.

زمن الاستجابة: منخفض.

الإنتاجية: عالية ومستمرة.

حالة الاستخدام: يسد الفجوة بين النموذجين الآخرين، مما يتيح التحليلات واتخاذ القرارات في الوقت الفعلي.

معالجة الدفعات (Batch Processing)

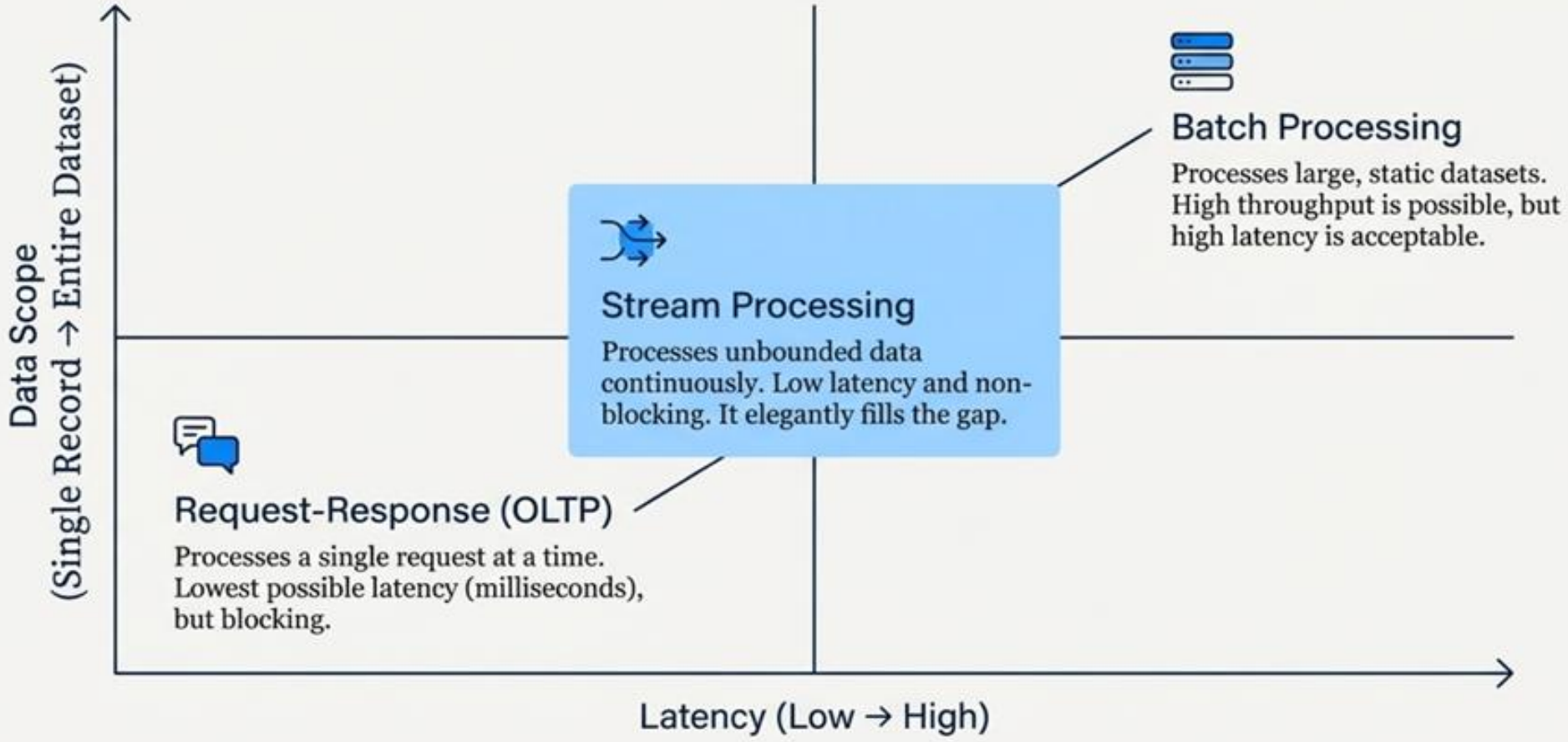
الوصف: تتم معالجة مجموعة البيانات "كاملة".

زمن الاستجابة: عالٍ، التقارير متاحة فقط بعد انتهاء المهمة.

الإنتاجية: عالية.

حالة الاستخدام: مثالي للتقارير غير العاجلة وأنظمة ذكاء الأعمال حيث يمكن أن تكون البيانات قديمة.

Choosing the right tool for the job: The Spectrum of Data Processing



Latency تعني زمن التأخير (زمن الاستجابة): هو الوقت الذي يستغرقه النظام من لحظة وصول البيانات أو الطلب إلى لحظة ظهور النتيجة .

Fault Tolerance - (تحمّل الأعطال) : هي قدرة النظام على الاستمرار بالعمل أو التعافي عند حدوث خطأ (مثل تعطل سيرفر، انقطاع شبكة، فشل مكوّن).

Takeaway: Choose stream processing when data is unbounded and low-latency, continuous computation is required.

نهجان للمعالجة الفورية: التدفق الأصيل مقابل الدفعات المصغرة

الدفعات المصغرة (Micro-Batching)

آلية العمل: يجمع الأحداث في دفعات صغيرة جدًا ويعالجها كل بضع مللي ثانية.

المزايا: تسامح أعلى مع الأخطاء بطبيعته. إنتاجية إنتاجية جيدة.

التحديات: زمن استجابة أعلى قليلاً (تأخير بسيط). إدارة الحالة بكفاءة تمثل تحديًا أكبر.

تجميع أحداث لفترة قصيرة ← معالجة دفعة صغيرة ← نتائج متكررة

Dashboards - عدد الزوار

Apache Spark

التدفق الأصيل (Native Streaming)

آلية العمل: يعالج كل سجل بيانات فور وصوله، دون انتظار.

المزايا: يحقق أقل زمن استجابة ممكن. إدارة الحالة (State management) تكون أبسط.

التحديات: قد يكون تحقيق التسامح العالي مع الأخطاء (Fault tolerance) صعبًا دورا، دون التأثير على الإنتاجية (بسبب الحاجة لكتابة نقاط الفحص).

الحدث يصل ← يُعالج فورًا ← النتيجة مباشرة

(Fraud Detection) اكتشاف الاحتيال

Apache Flink , Apache Kafka

محرك المعالجة: إطار عمل تدفق البيانات

"محرك لمعالجة البيانات مصمم مع أخذ مجموعات البيانات اللانهائية في الاعتبار. لا أكثر."

– تايلر أكيداو، مهندس برمجيات في جوجل

آلية العمل

- يستقبل إطار عمل التدفق (Streaming Framework) تدفقات البيانات من المصادر.
- يعالج المعلومات في الذاكرة (In-Memory) أولاً لتحقيق أقصى سرعة.
- يكتب النتائج النهائية إلى تخزين دائم (مثل الأقراص الصلبة أو قواعد البيانات).

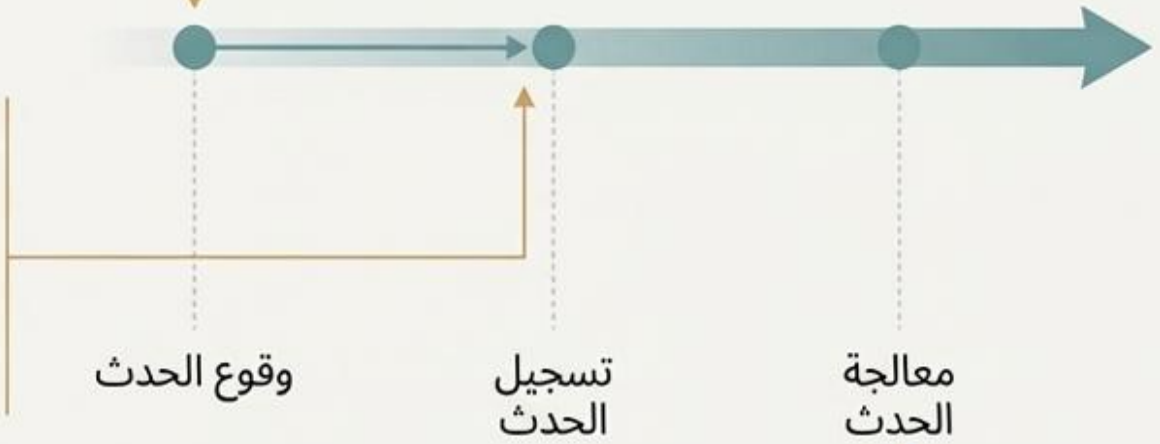
The Three Pillars of Stream Processing

To truly master stream processing, one must understand its three foundational mechanics: Time, State, and Windows.

مفهوم الزمن في معالجة التدفقات

الفكرة الأساسية:
معظم تطبيقات التدفق تنفذ عمليات ضمن نوافذ زمنية محددة.

المفهوم الرئيسي: الإطارات الزمنية (Windowing):
العمليات (مثل حساب المتوسطات أو المجاميع) تُجرى على مجموعات من الأحداث التي تقع ضمن إطار زمني محدد (مثال: متوسط سعر السهم في آخر 5 دقائق).



زمن الحدث (Event Time):
اللحظة التي وقع فيها الحدث في العالم الحقيقي الحقيقي (مثلاً، وقت نقر المستخدم على موقع ويب). **هذا هو الزمن الأكثر أهمية للمعالجة الدقيقة.**

زمن الإلحاق بالسجل (Log Append Time):
اللحظة التي تم فيها تسجيل الحدث في نظام المصدر (مثل وسيط الرسائل). يمكن استخدامه كبديل تقريبي لزمن الحدث إذا لم يكن الأخير متاحًا.

زمن المعالجة (Processing Time):
اللحظة التي استقبل فيها تطبيق المعالجة الحدث.

ذاكرة النظام: إدارة الحالة (State Management)

لماذا الحالة ضرورية؟

عندما تتطلب المعالجة معلومات من أحداث متعددة (مثل حساب 'عدد المشاهدات لكل نوع منتج خلال الساعة الماضية")، يجب على النظام
النظام تتبع النتائج الوسيطة. هذه "النتائج الوسيطة" تسمى **الحالة** (State).

الحالة الخارجية (External State)

- **الوصف:** تتم إدارتها في مخزن بيانات خارجي منفصل (مثل قاعدة بيانات NoSQL كـ Cassandra).
- **المزايا:** سعة تخزين غير محدودة تقريبًا وقابلة للمشاركة بين عدة نسخ من التطبيق.
- **العيوب:** يضيف زمن استجابة إضافي (network latency) وتعقيدًا للنظام.

الحالة الداخلية (Local / Internal State)

- **الوصف:** تتم إدارتها داخل ذاكرة التطبيق نفسه، غالبًا باستخدام قاعدة بيانات مدمجة.
- **المزايا:** سريعة جدًا في الوصول.
- **العيوب:** محدودة بسعة الذاكرة المتاحة في نسخة التطبيق.

وجهان لنفس البيانات: ازدواجية الجدول والتدفق Stream–Table Duality

** **التدفق (Stream):** هو السجل الكامل للتغييرات (log of events). يمثل *تاريخ* كيفية وصول العالم إلى حالته الحالية.
** **الجدول (Table):** هو *الحالة الحالية* للعالم، وهي نتيجة تطبيق جميع التغييرات في التدفق.

تدفق الأحداث

وصلت شحنة أحذية حمراء وزرقاء وخضراء
تم بيع حذاء أزرق
تم بيع حذاء أحمر
تم إرجاع حذاء أزرق
تم بيع حذاء أخضر

الجدول المتجسد (الحالة الحالية)

اللون	الكمية
أحمر	0
أزرق	1
أخضر	0

- **من الجدول إلى التدفق (Table to Stream):** يمكن تحويل جدول إلى تدفق عن طريق التقاط جميع التغييرات (إدراج، تحديث، حذف) المطبقة عليه. تُعرف هذه العملية باسم ****التقاط بيانات التغيير (Change Data Capture - CDC).**
- **من التدفق إلى الجدول (Stream to Table):** يمكن تحويل تدفق إلى جدول عن طريق تطبيق جميع أحداث التغيير من التدفق لإنشاء عرض محدث. تُعرف هذه العملية باسم ****التجسيد** (Materialization).

Materializationيعني:
تحويل Stream تدفّق أحداث إلى Tableحالة حالية.

مقاييس الجودة: معايير تقييم أنظمة التدفق (QoS)

لفهم نقاط القوة والضعف في أطر عمل التدفق المختلفة، يجب تقييمها بناءً على معايير جودة الخدمة (Quality of Service) الأساسية.

ضمانات المعالجة (Processing Guarantees)

الضمانات التي يقدمها النظام حول كيفية معالجة كل سجل بيانات (سيتم تفصيلها في الشريحة التالية).

إدارة الحالة (State Management)

القدرة على حفظ وتحديث الحالة بفعالية وموثوقية أثناء المعالجة.

التسامح مع الأخطاء (Fault Tolerance)

قدرة النظام على استئناف المعالجة من النقطة التي توقف عندها بعد حدوث فشل (مثل فشل عقدة أو مشكلة في الشبكة)، غالبًا باستخدام نقاط الفحص (Checkpoints).

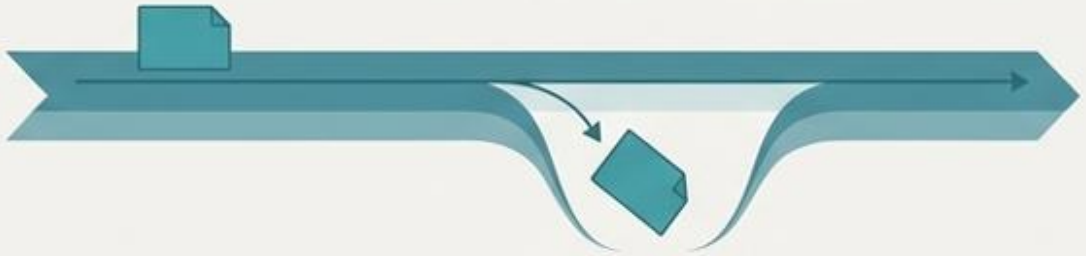
النضج والاعتمادية (Maturity & Adoption)

قوة مجتمع المطورين، الدعم المؤسسي، وحالات الاستخدام الناجحة في الشركات.

السرعة والأداء (Speed & Performance)

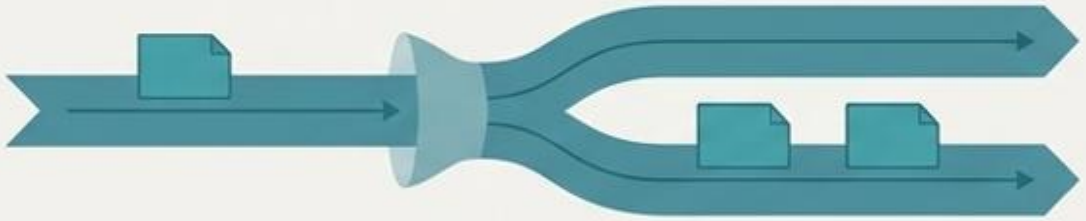
التوازن بين زمن الاستجابة المنخفض (Low Latency) والإنتاجية العالية (High Throughput).

ضمانات المعالجة: من "مرة على الأكثر" إلى "مرة واحدة بالضبط"



مرة على الأكثر (At-Most-Once)

- **الوصف":** قد يتم فقدان السجلات في حالة حدوث فشل، ولكن لن يتم تكرارها أبدًا.
- **المفاضلة:** الأسرع أداءً ولكنه الأقل موثوقية.



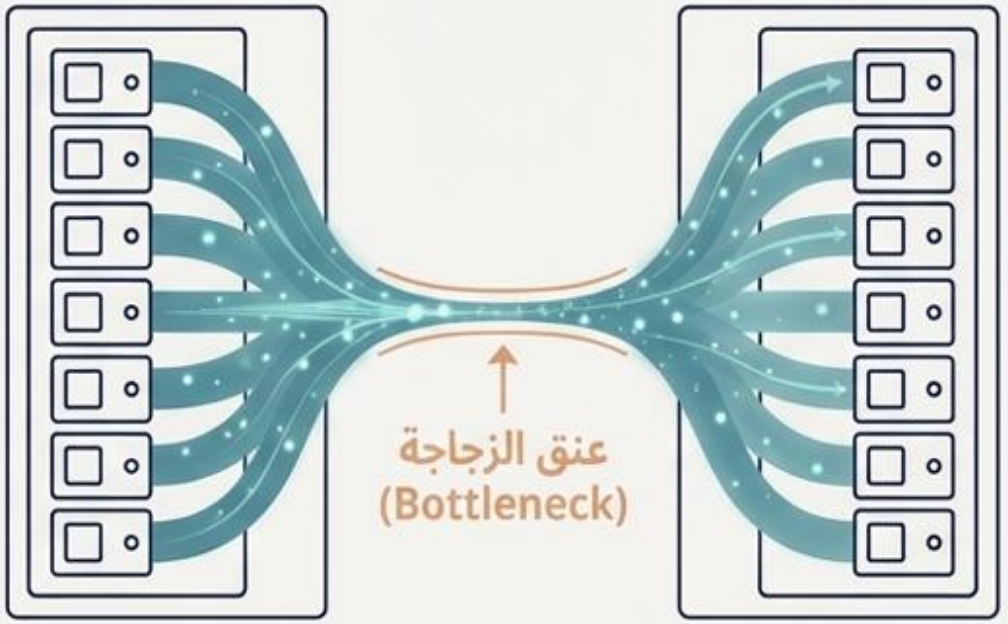
مرة على الأقل (At-Least-Once)

- **الوصف:** لن يتم فقدان أي سجل، ولكن قد تتم معالجة بعض السجلات أكثر من مرة في حالة استرداد الفشل.
- **المفاضلة:** يضمن عدم فقدان البيانات، ولكن يتطلب التعامل مع التكرارات.

مرة واحدة بالضبط (Exactly-Once)

- **الوصف:** يتم معالجة كل سجل مرة واحدة فقط، حتى في حالة وجود أخطاء.
- **المفاضلة: الخيار المفضل** لمعظم التطبيقات الحرجة. غالبًا ما يأتي على حساب الأداء بسبب التنسيق الإضافي المطلوب.

السعي نحو الأداء: الإنتاجية، زمن الاستجابة، والاختناقات



تحدي الاختناق (The Bottleneck Challenge)

في أي نظام موزع، أبطأ جزء يحدد الأداء الكلي تحت الحمل العالي.

مثال شائع: يمكن أن تكون النطاق الترددي للشبكة بين خزائن الخوادم (Racks) أقل بكثير من النطاق الترددي داخل الخزانة الواحدة، مما يخلق 'عنق زجاجة' يحد من الإنتاجية الإجمالية.

منظوران للأداء

منظور المستخدم (User Perspective):
الهدف هو **أقل وقت ممكن للوصول** إلى النتيجة النهائية. هذا يشمل كل شيء من استيعاب البيانات إلى تحليل النتائج.

منظور النظام (System Perspective):
الهدف هو **أقل تكلفة إجمالية للملكية (TCO)**. هذا يشمل سهولة النشر والإدارة، واستخدام الأجهزة بكفاءة.

التطبيق العملي: حالات استخدام معالجة التدفقات

تخصيص التجارة الإلكترونية (E-commerce Personalization)

لتقديم توصية المنتج المناسبة في صفحة الدفع، يحتاج النموذج إلى بيانات الجلسة الحالية (مثل سجل النقرات وسلة التسوق). تتيح معالجة التدفقات ذلك في لحظة اتخاذ القرار.

التحليلات المتدفقة وتعلم الآلة (Streaming Analytics & ML)

تطبيق نماذج تعلم الآلة المدربة مسبقًا على البيانات الحية لتسجيل النتائج (Model Scoring) بشكل فوري. يتيح ذلك الكشف عن الاحتيال، أو الصيانة التنبؤية، أو التحليلات السلوكية في الوقت الفعلي.

تحويل وتحميل البيانات المتدفق (Streaming ETL)

بدلاً من انتظار عمليات التحميل الليلية (Nightly Builds)، يمكن معالجة البيانات وتحويلها فور وصولها. هذا يوفر وقتًا ثمينًا ويمكّن من اتخاذ قرارات أسرع بناءً على أحدث البيانات.