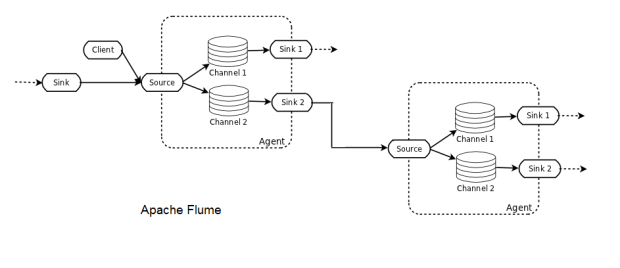
# Data Ingestion (Mengambil Data)

Saat membangun jaringan pipa (pipeline) data yang besar, kita perlu memikirkan bagaimana mengambil volume, variasi, dan kecepatan data dan memunculkan ke ekosistem Hadoop. Pertimbangan awal seperti skalabilitas, keandalan, kemampuan beradaptasi, biaya dalam hal waktu pengembangan, dan lain-lain akan mulai berperan saat menentukan alat mana yang harus digunakan untuk memenuhi persyaratan tersebut. Pada artikel ini, kita akan fokus secara singkat pada tiga alat Apache ingestion tools: Flume, Kafka, dan NiFi. Ketiga produk tersebut menawarkan kinerja yang hebat, dapat diskalakan secara horizontal, dan menyediakan arsitektur plug-in dimana fungsionalitas dapat diperluas melalui komponen kustom.

## Apache Flume

**Flume** adalah sistem terdistribusi yang bisa digunakan untuk mengumpulkan, menggabungkan, dan mentransfer acara streaming ke Hadoop. Muncul dengan banyak built-in sumber, saluran, dan sink, misalnya, Kafka Channel dan Avro sink. Flume berbasis konfigurasi dan memiliki interseptor untuk melakukan transformasi sederhana pada data.

Sangat mudah kehilangan data jika menggunakan Flume dengan tidak hati-hati. Misalnya, memilih saluran memori untuk throughput yang tinggi memiliki sisi negatif bahwa data akan hilang saat node agen turun. Saluran file akan memberikan daya tahan pada harga latensi yang meningkat. Bahkan saat itu, karena data tidak direplikasi ke node lain, saluran file hanya dapat diandalkan seperti disk yang mendasarinya. Flume memang menawarkan skalabilitas melalui aliran kipas-arus multi-hop / fan-in. Untuk ketersediaan tinggi (HA), agen dapat diukur secara horisontal.



Apache Kafka

Kafka adalah bus pesan terdistribusi dan high-throughput yang memisahkan produsen data dari konsumen. Pesan disusun menjadi topik, topik dipecah menjadi partisi, dan partisi direplikasi di node - disebut pialang - di cluster. Dibanding Flume, Kafka menawarkan skalabilitas dan daya tahan pesan yang lebih baik. Kafka sekarang hadir dalam dua rasa: model produsen / konsumen "klasik", dan Kafka-Connect baru, yang menyediakan konektor (sumber / sink) yang dapat dikonfigurasi ke toko data eksternal.

Kafka dapat digunakan untuk pemrosesan acara dan integrasi antar komponen sistem perangkat lunak besar. Lonjakan data dan tekanan balik (fast producer, slow consumer) ditangani out-of-the-box. Selain itu, kapal Kafka dengan Kafka Streams, yang dapat digunakan untuk pemrosesan arus sederhana tanpa memerlukan cluster terpisah seperti Apache Spark atau Apache Flink.

Karena pesan terus berlanjut pada disk dan juga direplikasi dalam cluster, skenario kehilangan data kurang umum dibandingkan dengan Flume. Konon, pengkodean kustom sering dibutuhkan untuk produsen / sumber dan konsumen / sink, baik menggunakan klien Kafka atau melalui Connect API. Seperti Flume, ada keterbatasan dalam ukuran pesan. Akhirnya, agar bisa berkomunikasi, produsen dan konsumen Kafka harus menyetujui protokol, format, dan skema, yang bisa bermasalah dalam beberapa kasus.

Apache NiFi

Tidak seperti Flume dan Kafka, NiFi. bisa menangani pesan dengan ukuran yang sewenang-wenang. Di balik UI berbasis drag-and-drop, NiFi berjalan dalam sebuah cluster dan menyediakan kontrol real-time yang memudahkan pengelolaan pergerakan data antara sumber dan tujuan manapun. Ini mendukung sumber yang berbeda dan terdistribusi dengan format, skema, protokol, kecepatan, dan ukuran yang berbeda.

NiFi dapat digunakan dalam arus data mission-critical dengan persyaratan keamanan dan kepatuhan yang ketat, di mana kita dapat memvisualisasikan keseluruhan proses dan segera melakukan perubahan, secara real-time. Pada saat penulisan ini, telah mendekati 200 prosesor out-of-the-box (termasuk prosesor Flume dan Kafka) yang dapat diseret & dijatuhkan, dikonfigurasi dan segera bekerja. Beberapa fitur utama NiFi diprioritaskan pada antrian, ketertelusuran data dan konfigurasi ambang batas tekanan belakang per koneksi.

Meskipun digunakan untuk membuat pipa produksi toleransi kegagalan, NiFi belum meniru data seperti Kafka. Jika sebuah simpul turun, aliran dapat diarahkan ke node lain, namun antri data untuk node yang gagal harus menunggu sampai node tersebut kembali muncul. NiFi bukanlah alat ETL lengkap, juga tidak ideal untuk perhitungan kompleks dan pemrosesan acara (CEP). Untuk itu, seharusnya malah terhubung ke framework streaming seperti Apache Flink, Spark Streaming atau Storm.

Kombinasi

Tidak ada alat satu-satunya yang bisa melakukan semuanya dengan baik dan memenuhi semua kebutuhan Anda. Menggabungkan alat yang melakukan hal yang berbeda dengan cara yang lebih baik memungkinkan peningkatan fungsi dan fleksibilitas yang meningkat dalam menangani serangkaian skenario yang lebih besar. Bergantung pada kebutuhan Anda, baik NiFi dan Flume dapat bertindak sebagai produsen dan / atau konsumen Kafka.

Integrasi Flume-Kafka cukup populer, namanya punya nama sendiri: Flafka (saya tidak membuat ini). Flafka termasuk sumber Kafka, saluran Kafka, dan wastafel Kafka. Menggabungkan Flume dan Kafka memungkinkan Kafka untuk menghindari pengkodean ubahsuaian dan memanfaatkan sumber dan sink yang diuji oleh Flume, sementara peristiwa Flume yang melewati saluran Kafka disimpan dan direplikasi di pialang Kafka untuk mendapatkan ketenangan.

Menggabungkan alat mungkin tampak boros, karena tampaknya mengenalkan beberapa fungsi yang tumpang tindih. Misalnya, baik NiFi dan Kafka menyediakan broker untuk menghubungkan produsen dan konsumen. Namun, mereka melakukannya dengan cara yang berbeda: di NiFi, sebagian besar logika arus data tidak berada di dalam produsen / konsumen, namun tinggal di broker, memungkinkan kontrol terpusat. NiFi dibangun untuk melakukan satu hal penting dengan baik: pengelolaan arus data. Dengan kedua alat yang dikombinasikan, NiFi dapat memanfaatkan penyimpanan data stream Kafka yang andal, sambil memperhatikan tantangan dataflow yang tidak dapat diselesaikan oleh Kafka.

|  |  |  |  |
| --- | --- | --- | --- |
| Ingestion Tool | Out-of-the box | Batasan | Penggunaan |
| Flume | * Berbasis konfigurasi * Banyak Sumber, chanel dan sinks * Interceptor | * kehilangan data ketika tidak menggunakan Kafka Chanel * Data size (KB) * Tidak ada replikasi data | * Collecting, aggregating dan memindahkan striming dengan volume yang besar kedalam hadoop |
| Kafka | * Back-pressure * Reliable stream data storage * Kafka streams * Sources/sink menggunakan kafka Connect | * Kustom koding diperlukan * Data size (kb) * Fixed protocol format/skema | * Streaming data * Messaging * Systems integration * Commit log |
| Nifi | * konfigurasi berbasis UI * banyak drag & drop processors * Back-pressure * Prioritized queuing * Flow templat | * Tidak untuk CEP atau komputasi window * Tidak ada replikasi data | * Data flow management dengan visual control |