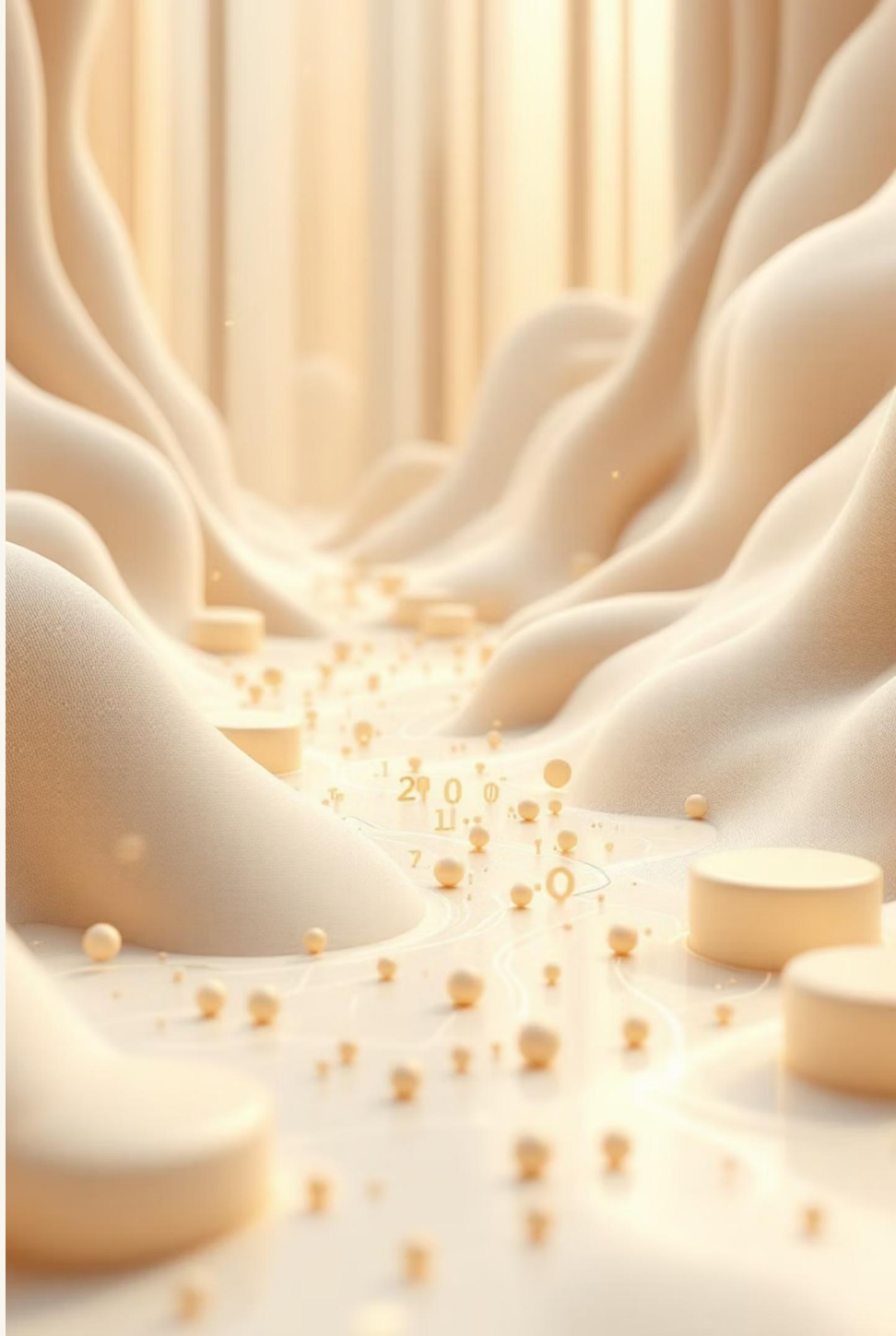
Mata Kuliah Web Information Retrieval (3 SKS)

Topik: IR dan Big Data – Skalabilitas, MapReduce, dan Spark



Tujuan Pembelajaran

1

Mengenali Tantangan IR Skala Besar

Memahami kompleksitas dan skala data yang dihadapi dalam sistem Information Retrieval modern.

2

Memahami Konsep Skalabilitas

Mempelajari bagaimana sistem dapat ditingkatkan untuk menangani volume data dan beban kerja yang terus meningkat.

3

Memahami Peran MapReduce & Spark

Menganalisis kontribusi fundamental teknologi ini dalam memproses dan mengelola data untuk IR skala besar.

Apa itu Information Retrieval (IR) Skala Besar?

IR tradisional: Fokus pada pencarian dokumen dalam dataset yang ukurannya relatif terbatas dan statis. Pendekatan ini seringkali mengandalkan database terpusat dan algoritma pencarian yang tidak dirancang untuk distribusi.

IR skala besar: Menangani volume data yang sangat besar, mencapai miliaran dokumen, dan harus mampu merespons permintaan pencarian (query) secara real-time. Lingkungan ini jauh lebih dinamis dan kompleks.

Tantangan utama:

- **Volume data:** Jumlah data yang harus diindeks dan dicari sangat besar.
- **Kecepatan:** Kebutuhan untuk memberikan hasil pencarian dalam hitungan milidetik, meskipun datanya masif.
- **Kompleksitas data:** Data tidak hanya teks, tetapi juga gambar, video, dan berbagai format tidak terstruktur.

Tantangan Skalabilitas dalam IR

Pertumbuhan data global sangat pesat, dengan perkiraan 90% dari seluruh data global yang ada saat ini baru dibuat dalam dua tahun terakhir. Ini menciptakan tekanan besar pada sistem IR untuk terus beradaptasi.

1

Memproses Data Terdistribusi

Sistem harus mampu memecah dan memproses data di berbagai server secara bersamaan, bukan hanya pada satu mesin.

2

Menangani Data Beragam

Integrasi data terstruktur (misalnya, metadata) dan tidak terstruktur (misalnya, teks bebas) dalam satu sistem pencarian.

3

Menyediakan Hasil Cepat & Akurat

Keseimbangan antara kecepatan respons dan relevansi hasil pencarian menjadi krusial di tengah volume data yang sangat besar.

MapReduce: Paradigma Pemrosesan Data Terdistribusi

MapReduce adalah model pemrograman yang dikembangkan oleh Google untuk memproses dataset berukuran sangat besar secara paralel pada kluster komputer terdistribusi. Konsep ini kemudian diadopsi dan menjadi inti dari ekosistem Hadoop.

- **Tahap Map (Pemetaan):** Data input dibagi menjadi potongan-potongan kecil. Setiap potongan diproses secara independen oleh "mapper" yang menghasilkan pasangan kunci-nilai (key-value pairs).
- **Tahap Reduce (Pengurangan):** Hasil dari tahap Map kemudian dikelompokkan berdasarkan kunci (key) yang sama dan diteruskan ke "reducer". Reducer akan menggabungkan nilai-nilai yang terkait dengan kunci yang sama untuk menghasilkan output akhir.

MapReduce sangat cocok untuk **batch processing** data besar, seperti membangun indeks dokumen web atau melakukan analisis log berskala besar.

Contoh: Membuat indeks terbalik (inverted index) untuk miliaran dokumen web. Setiap mapper memproses sebagian dokumen, mengekstrak kata-kata, dan mencatat dokumen asal kata tersebut. Reducer kemudian mengumpulkan semua entri untuk setiap kata dan membuat daftar dokumen di mana kata itu muncul.

Hadoop Ecosystem dan MapReduce

MapReduce adalah salah satu komponen kunci dalam ekosistem Hadoop, yang menyediakan fondasi untuk penyimpanan dan pemrosesan data besar.

HDFS (Hadoop Distributed File System)

Sistem file terdistribusi yang dirancang untuk menyimpan data berukuran sangat besar secara redundan di banyak mesin. Ini adalah tulang punggung penyimpanan data di Hadoop.

YARN (Yet Another Resource Negotiator)

Manajer sumber daya kluster yang bertanggung jawab untuk mengalokasikan sumber daya komputasi (CPU, memori) ke berbagai aplikasi yang berjalan di Hadoop, termasuk MapReduce.

MapReduce

Framework pemrosesan paralel yang memungkinkan aplikasi untuk memproses data besar secara terdistribusi. Ini mengelola eksekusi tugas Map dan Reduce di seluruh kluster.

Meskipun powerful, salah satu **kelemahan** utama MapReduce adalah **lambat untuk proses real-time** atau iteratif karena banyaknya operasi I/O disk yang diperlukan di antara setiap tahap.

Apache Spark: Revolusi Pemrosesan Big Data untuk IR

Apache Spark muncul sebagai solusi untuk mengatasi keterbatasan MapReduce, menawarkan kecepatan dan fleksibilitas yang lebih besar dalam pemrosesan data besar.

- **Memproses data di memori (in-memory computing):** Ini adalah keunggulan utama Spark. Dengan menyimpan data di RAM selama pemrosesan, Spark dapat 10-100x lebih cepat daripada MapReduce, terutama untuk beban kerja iteratif atau interaktif.

- **Mendukung batch dan real-time processing:** Spark tidak hanya unggul dalam pemrosesan batch, tetapi juga menyediakan fungsionalitas **Spark Streaming** untuk memproses aliran data secara real-time, sangat penting untuk IR modern yang membutuhkan respons instan.
- **Library MLlib untuk machine learning:** Spark dilengkapi dengan MLlib, sebuah perpustakaan pembelajaran mesin yang kaya fitur untuk data besar. Ini memungkinkan pengembangan model ranking, klasifikasi, dan personalisasi dalam IR.
- **Kompatibilitas:** Spark dapat berjalan di atas HDFS dan berintegrasi mulus dengan ekosistem Hadoop, memungkinkan organisasi untuk memanfaatkan infrastruktur yang sudah ada.

Perbandingan MapReduce vs Spark untuk IR

Model Pemrosesan	Batch processing disk-based	Batch, streaming, interaktif, in-memory
Kecepatan	Lambat untuk iterasi (banyak I/O disk)	Sangat cepat (in-memory computing)
Fleksibilitas	Sederhana, fokus pada Map/Reduce	API kaya, banyak library (SQL, ML, Graph)
Kasus Penggunaan	Pembangunan indeks awal, analisis log besar	Pencarian real-time, ranking personalisasi, analisis sentimen
Persyaratan Sumber Daya	Hardware komoditas (disk-heavy)	RAM tinggi (lebih mahal)

Pilihan antara keduanya sangat bergantung pada kebutuhan spesifik sistem IR dan ketersediaan sumber daya.

Studi Kasus: Implementasi IR Skala Besar dengan Spark

Mari kita lihat bagaimana Spark dapat diimplementasikan dalam skenario dunia nyata untuk IR skala besar.

Analisis Sentimen Real-time Media Sosial

Perusahaan memantau jutaan tweet dan postingan setiap hari untuk memahami sentimen publik terhadap merek mereka. Diperlukan kemampuan untuk memproses data ini secara instan.

Spark Streaming untuk Pemrosesan Data Cepat

Spark Streaming digunakan untuk menyerap aliran data dari platform media sosial. Data diproses dalam micro-batch secara real-time, diekstraksi fitur, dan sentimen dianalisis menggunakan model MLlib.

Integrasi HDFS untuk Penyimpanan Jangka Panjang

Data mentah dan hasil analisis disimpan di HDFS untuk analisis historis dan pelatihan model ulang. Spark dapat membaca dan menulis ke HDFS dengan efisien.

Hasil: Respons Cepat & Akurasi Tinggi

Dengan Spark, perusahaan mendapatkan wawasan sentimen hampir secara instan, memungkinkan respons cepat terhadap tren atau krisis. Sistem ini juga skalabel untuk menangani peningkatan volume data selama event besar.

Kesimpulan & Arah Pembelajaran Selanjutnya

IR Skala Besar: Kebutuhan Skalabilitas & Kecepatan

Web Information Retrieval modern dihadapkan pada volume data yang masif, menuntut solusi yang tidak hanya skalabel tetapi juga mampu memberikan respons cepat.

MapReduce & Hadoop: Fondasi Batch Processing

MapReduce dan ekosistem Hadoop tetap relevan untuk beban kerja batch processing berskala besar, terutama untuk tugas-tugas seperti pembangunan indeks awal.

Spark: Kecepatan & Fleksibilitas untuk IR Modern

Apache Spark menawarkan lompatan besar dalam kecepatan berkat in-memory computing dan fleksibilitas untuk memproses data secara real-time dan menjalankan algoritma machine learning yang kompleks.

Pilihan Teknologi Sesuai Kebutuhan

Mahasiswa diharapkan mampu menganalisis karakteristik proyek IR dan memilih teknologi Big Data yang paling sesuai, apakah itu MapReduce, Spark, atau kombinasi keduanya.

Eksplorasi Implementasi & Optimasi

Langkah selanjutnya adalah mendalami implementasi praktis, mempelajari optimasi kinerja, dan memahami integrasi sistem IR dengan ekosistem Big Data yang lebih luas.