Module 5

Data Ingestion: Storage and Maintenance

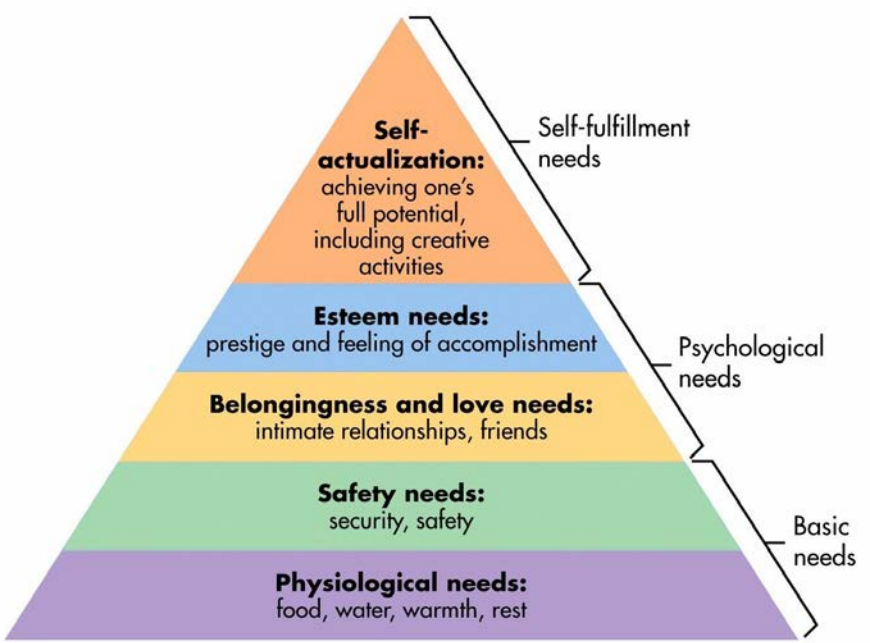
Bottom Line

Analytics solutions start with data ingestion

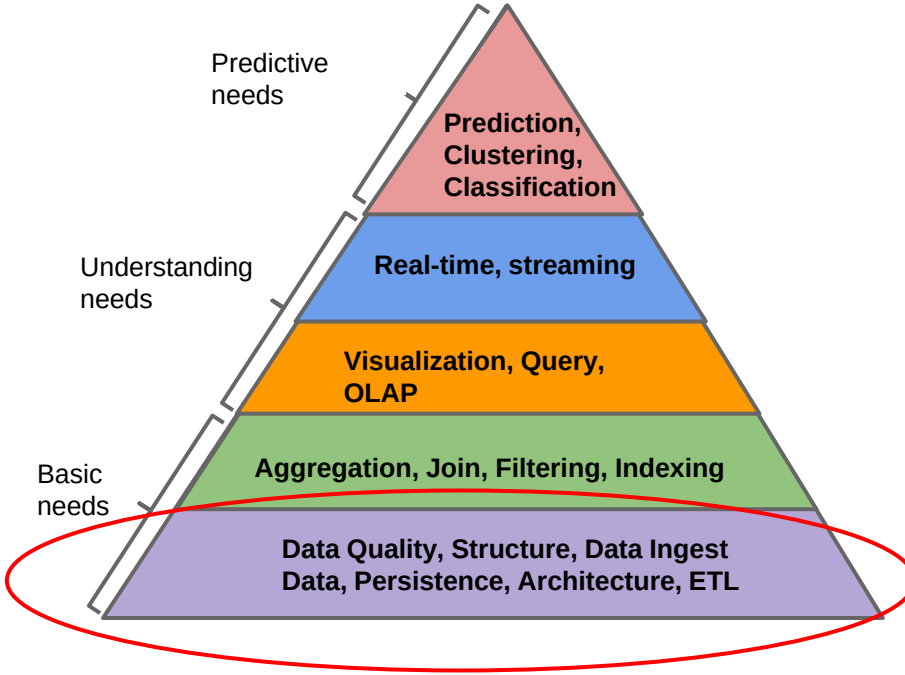
The problem can be that of **volume** (many similar integrations), **variety** (many different integrations) or **velocity** (batch v.s real-time) or all of the above.

Needs of Data Analytics

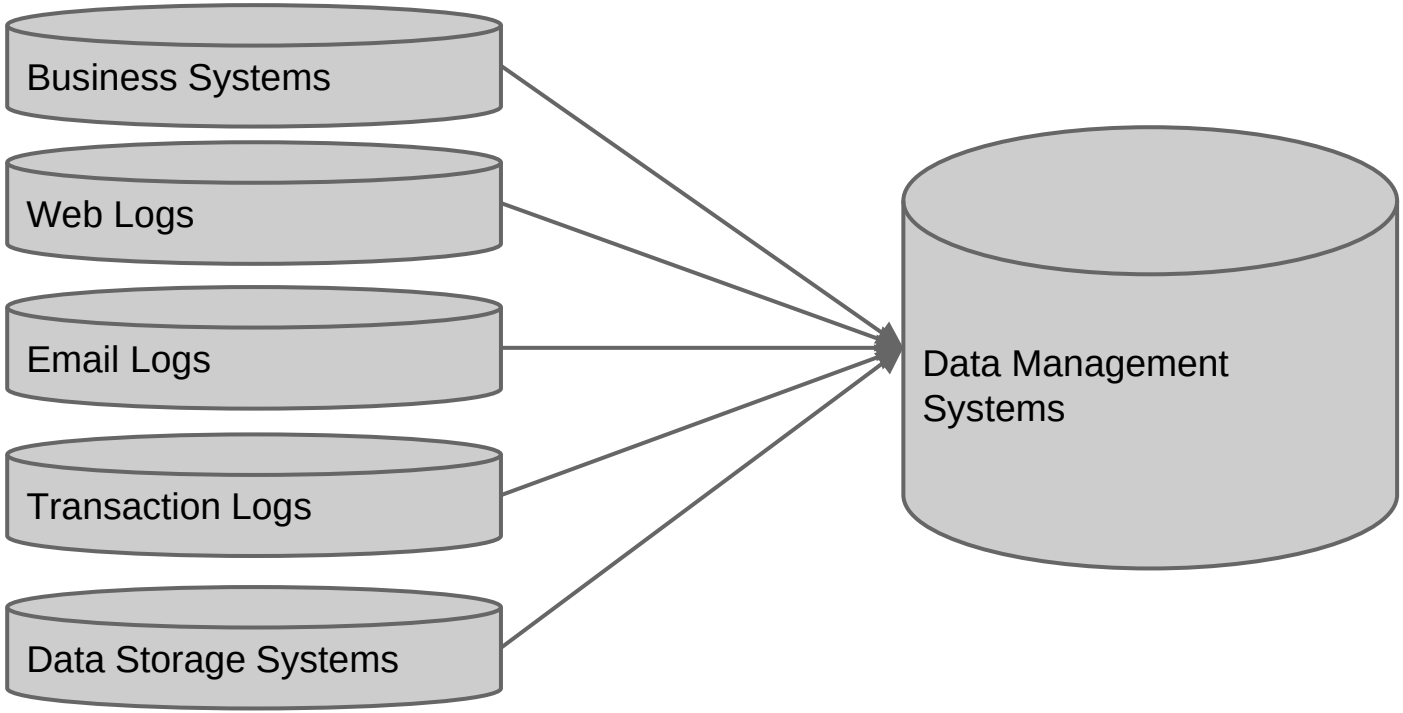
Maslow's Pyramid of needs



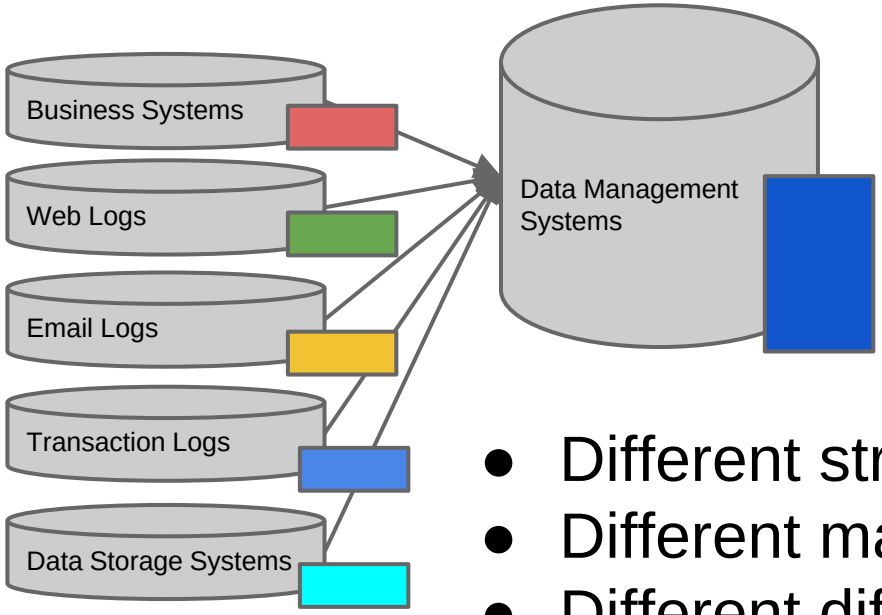
Pyramid of effective Analytics



Challenge: Many Sources

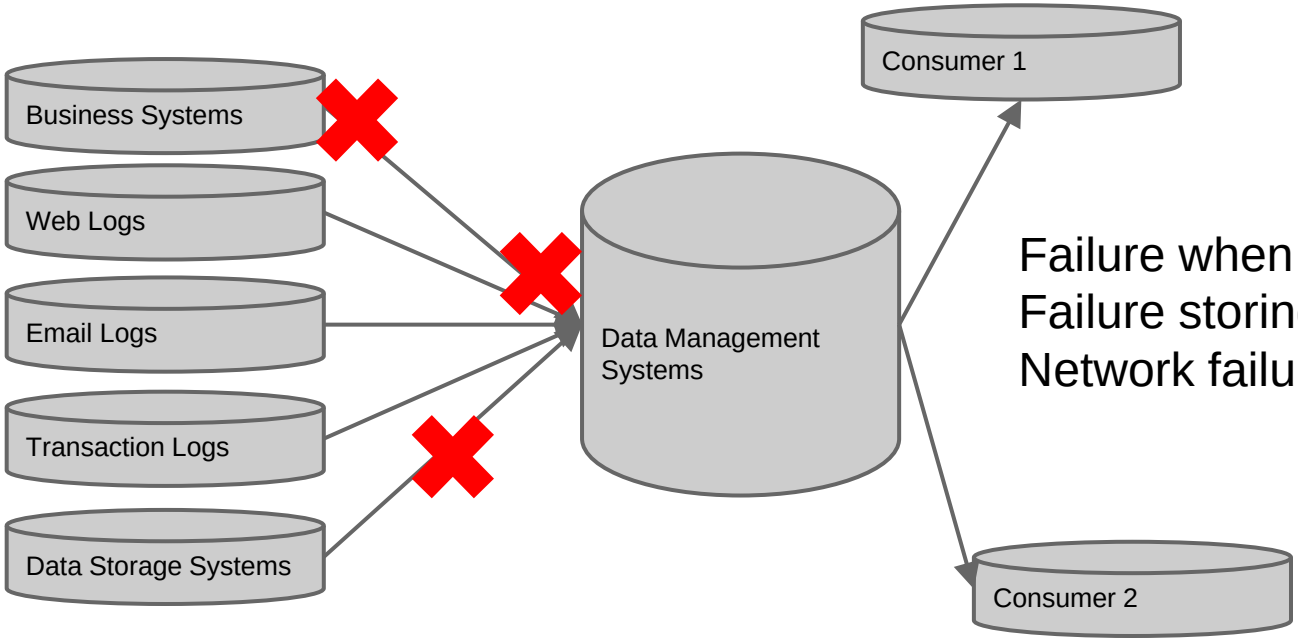


Challenge: Many Schemas



- Different structures
- Different manifest data types
- Different different literal for same data type.
- Different Keys

Challenge: Failures

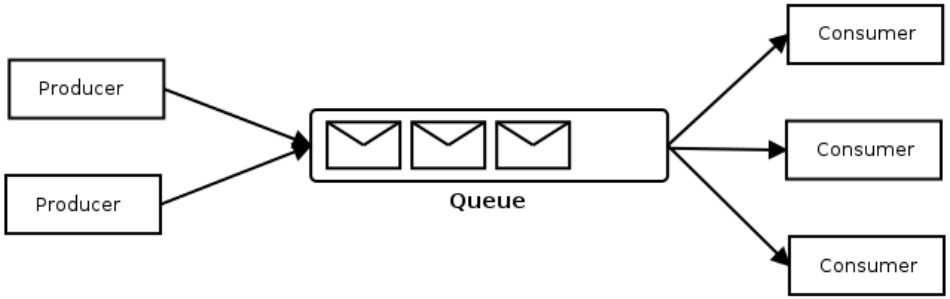


Failure when producers push data
Failure storing received data
Network failures while transferring

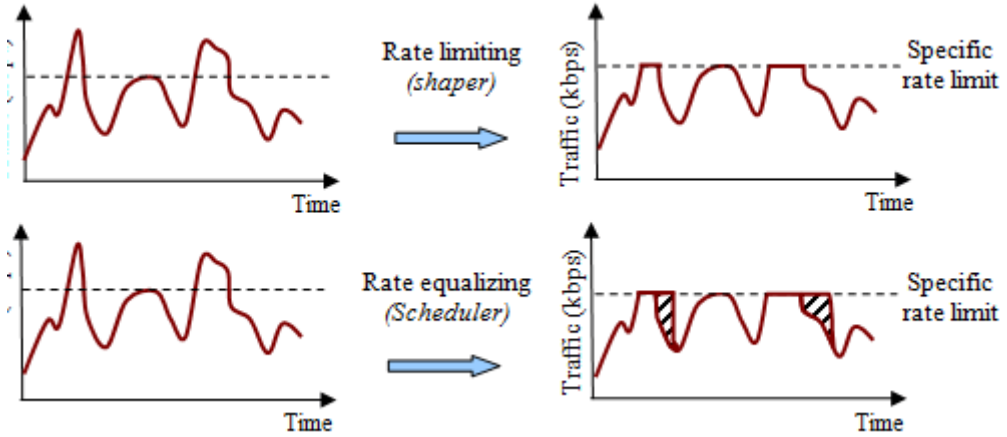
Data Bundle Semantics examples

Full dumps	All data is provided at once and replaces all previous data.
Incremental	Increments are provided in any order, data interval in an increment is provided as metadata. On an hourly, daily or weekly cadence.
Append	Always appended at the end of the dataset. Order is assumed to be correct.
Change stream	Stream of incoming data as individual rows or in small batches. On a sb second, minute or hourly cadence.

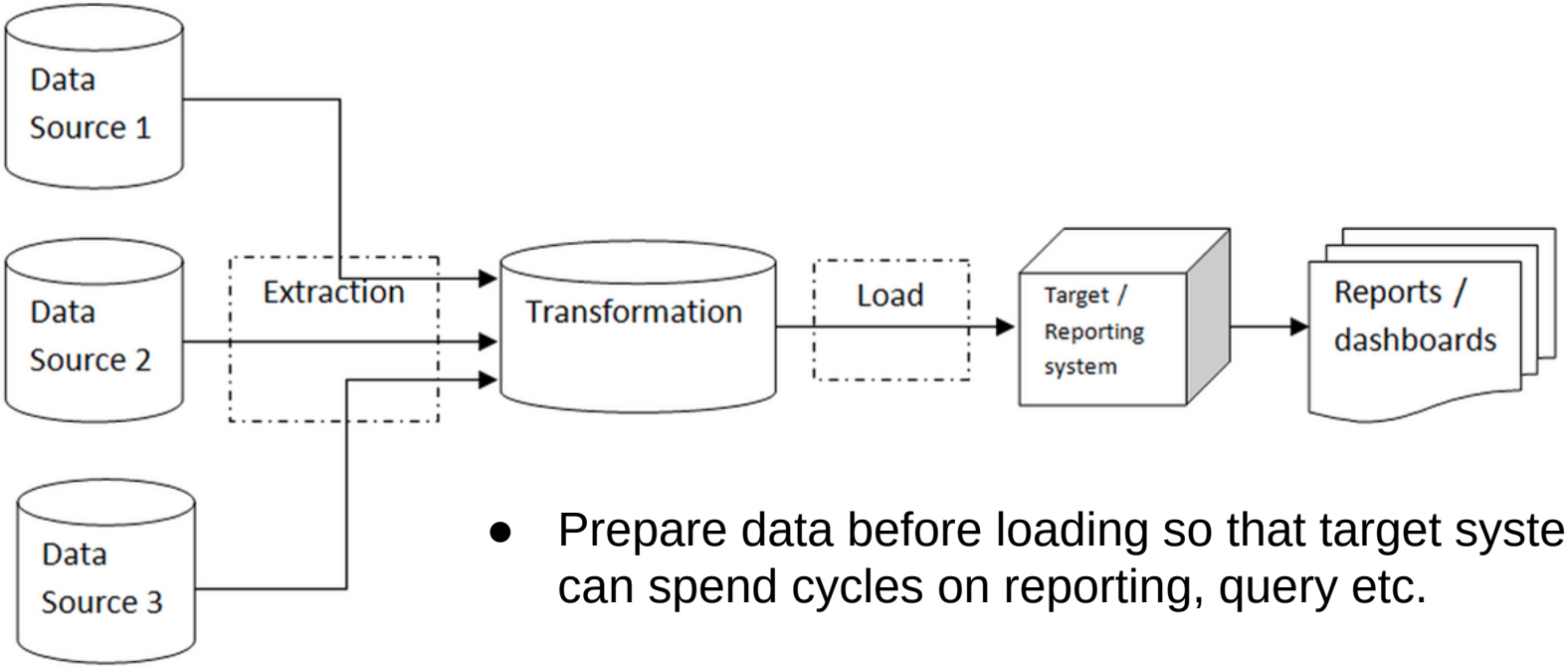
Bursts



- Data is produced in Bursts
- Consumers can only consume at a certain rate
- Dropping data can be a problem

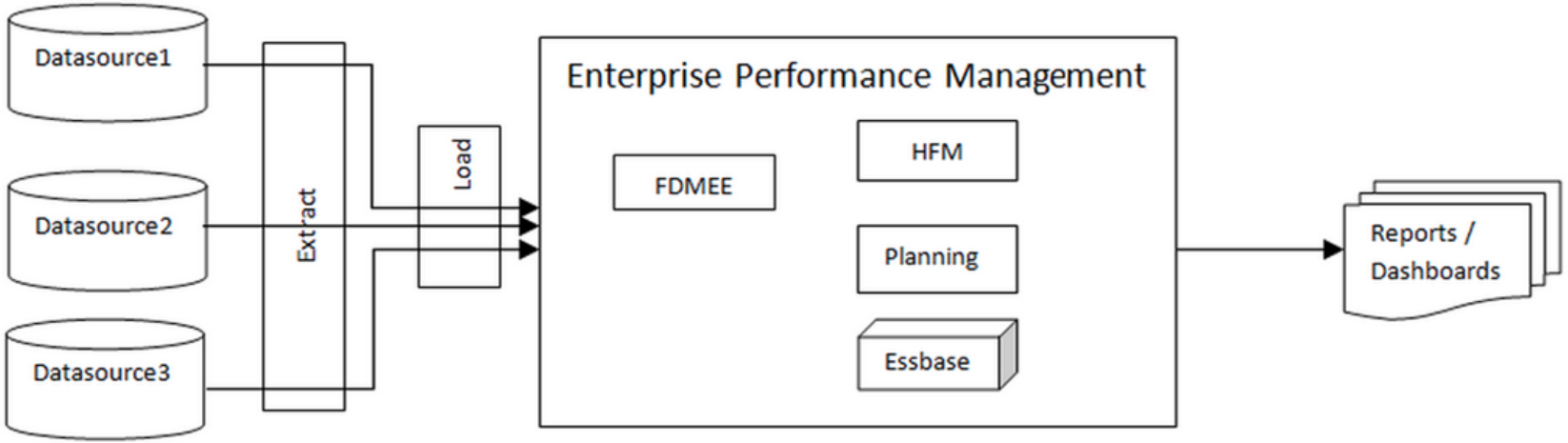


Traditional: Extract, Transform, Load



- Prepare data before loading so that target system can spend cycles on reporting, query etc.
- Requires transforms to know what reporting, query to enable.

Traditional: Extract, Load, Transform

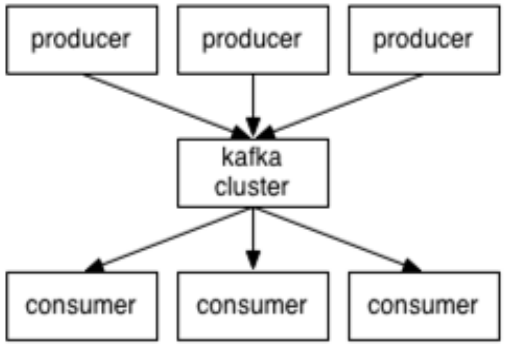


- Made possible by more powerful target systems.
- Provides more flexibility at later stages than ETL.

High Velocity Technologies

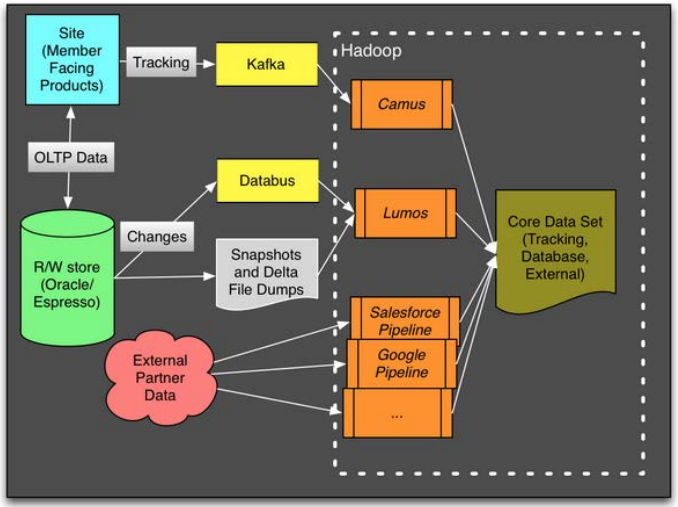
Kafka	Kafka is a distributed, partitioned, replicated commit log service. It provides the functionality of a messaging system, but with a unique design.
Kinesis	Amazon Kinesis is a fully managed, cloud-based service for real-time data processing over large, distributed data streams. Amazon Kinesis can continuously capture and store terabytes of data per hour from hundreds of thousands of sources.
S4	S4 is a general-purpose, distributed, scalable, fault-tolerant, pluggable platform that allows programmers to easily develop applications for processing continuous unbounded streams of data.
Storm	Apache Storm is a distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.
Samza	Apache Samza is a distributed stream processing framework. It uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management.

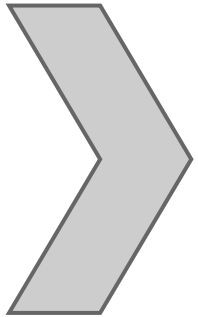
Kafka Approach

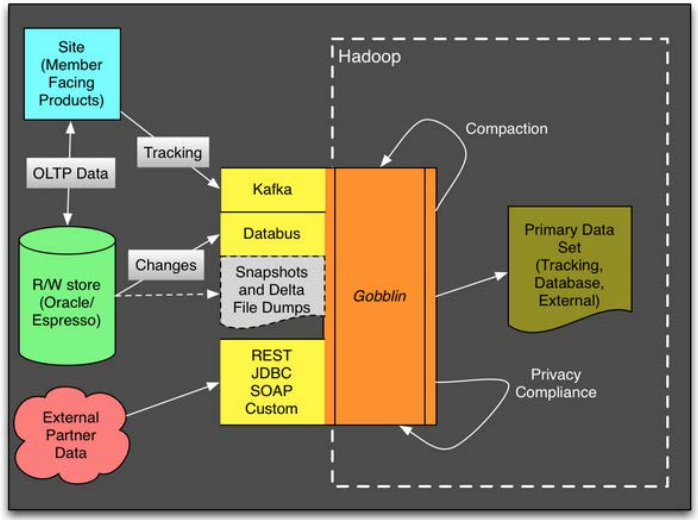


Scalability	Allows many producers and consumers. Partitions are the unit of scale.
Schema Variety	Does not really solve this.
Network Bottlenecks	Can handle some variability without loosing messages.
Consumer Bottlenecks	Kafka acts as a buffer allowing producers and consumers to work at different speeds.
Bursts	Handles buffering of messages between producers and consumers.
Reliability, Fault Tolerance	Allows reading of messages if a consumer fails.

Big Data Ingest: Logs + ETL







Linkedin
Gobblin

- Reduce complexity
- Reuse
- Self-service

Moving Large Data Considerations

If data is distributed how can you leverage parallelism?

What kind of source and sink is involved?

How do you use network bandwidth efficiently?

How to handle different formats and structures?

Large files take a long-time, how are failures handled?

As a data scientist you need to understand how to think about data transfer and movement

Performance Measures

Bandwidth measured in bits/sec is the maximum transfer rate.

Throughput is the actual rate that information is transferred

Latency the delay between the sender and the receiver, this is a function of the calls/signals travel time and intermediate processing time.

Jitter variation in the time of arrival at the receiver of the information

Error rate # of corrupted bits as a percentage or fraction of the total sent

Transfer Times

Limiting Factors:

- Available Network Bandwidth.
- Read/Write performance.
- Error rates.

Some info (for Disk storage we use base 10)

- 1 byte = 8 bits
- 1000 Bytes = 1 Kilobyte
- 1000 Kilobytes = 1 Megabyte
- 1000 Megabytes = 1 Gigabyte
- 1000 Gigabytes = 1 Terabyte

2 TB file, transferred over a 100 Mbit / sec connection. How long would it take?

2TB = 2*1000*1000*1000*1000*8 = 1.1.6E+13 bit

100 Mbit = 100*1000*1000 = 100000000

seconds = 1.1.6E+13 / 100000000 = 160000

hours = 160000 / 60 / 60 = 44 hours

Example Tools

Tool	What
Sqoop	RDBMS, BDW to Hadoop
distcp2	HDFS to HDFS copy
Rsync	FS to FS copy, FS to FS synchronization

Tools Comparison

	Sqoop	rsync	distcp
Type(s) of source and sink	RDBMS to HDFS	A unix / linux system inter transfer tool	HDFS to HDFS Storage Service (S3) to HDFS
Network usage	High	Smart use of bandwidth for sync	Uses available bandwidth
Level of parallelism	Low	No built-in parallelism.	High. Configurable
Structures and formats	Per table or free form SQL	Agnostic to file content	Agnostic to structure in files
Resilience to failures	If task fails it will be rolled back and result in a partial export	Need to be reinitiates of fails. Needs to restarted.	High, will retry per job. If one job fails it will be restarted.

Summary: Moving Large Data

Understand the bottleneck and your use case

Define your time constraints, can you move all the data or do you need to segment

Pick the right tool