Batch Systems

A **batch system** is a computing paradigm where work (jobs or tasks) is accumulated into groups (“batches”) and then processed all at once, rather than handling each job immediately as it arrives.

Analogy..

Imagine you have a big pile of dirty dishes:

1.Collect: You let dishes stack up throughout the day.

2.Load: Once the dishwasher is full (or it’s a set time), you start the cycle.

3.Clean in bulk: All the dishes get washed together.

4.Retrieve: When it’s done, you empty the clean dishes and move on.

vs

A **batch system** (computing works the same way):

1.Gather tasks instead of running each one the moment it’s ready.

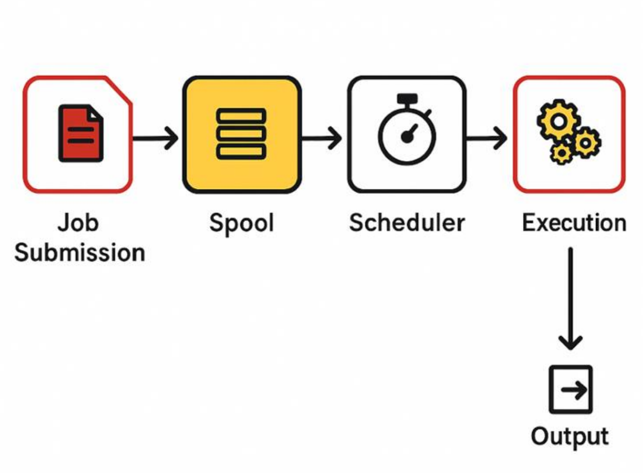
2. Hold them in a queue until there’s enough work or a scheduled time arrives.

3. Execute the whole batch in one go, maximizing the computer’s efficiency.

4. Collect results afterward—no need to watch each step as it happens.

Batch systems are everywhere (Examples):

- Banking and Finance: End-of-Day Batch Posting
- Manufacturing: Production Scheduling
- Data Warehousing: Extract-Transform-Load Batching
- Telecommunications: Call Detail Record (CDR) Billing
- E-commerce: Inventory Update
- Scientific & High-Performance Computing: Map Reduce
- Machine Learning Model Training: Stable Learning



Fun Fact: “Batch” *comes from Old Norse bọt, meaning “a quantity” or “lump,” reflecting the idea of grouping items for collective processing.*

Key Characteristics of a Batch System:

•**Accumulation**: Jobs build up in a queue until the system has enough work or reaches a scheduled time to run them.

•**Throughput-oriented**: By processing many jobs together, the system maximizes resource utilization (CPU, I/O, memory).

•**Non-interactive**: Users submit jobs and come back later for the results, rather than watching them execute in real time.

•**Scheduling & Prioritization**: A scheduler decides the order and time slices for each batch based on priorities, resource availability, or time windows.

Batch vs. Real-Time Processing:

Aspect	**Batch System**	**Real-Time System**
When	Scheduled intervals or when queued	Immediately, as events occur
Interaction	Submit & forget; check later	Interactive; provides instant feedback
Ideal for	Large-scale data processing, reports, backups	Live monitoring, user interfaces, control loops
Resource use	High throughput, optimized bursts	Low latency, continuous availability

Summary:
Batch System accumulates jobs into groups and processes them at once
Efficiency: Running many jobs together saves overhead (just like one dishwasher cycle uses less water per plate than washing each plate immediately).
Convenience: You don’t have to babysit each task—you just “load up” and let it run.
High throughput: Keeps the processor busy, tackling many jobs back-to-back for peak performance.

Note: Throughput is** the **number of jobs (or tasks) completed per unit of time