Large Scale Systems
CS 410 / 510

Lecture 8:

Evaluating Large Scale System

Suyash Gupta

Assistant Professor

Distopia Labs and ONRG

Dept. of Computer Science

(E) suyash@uoregon.edu

(W) gupta-suyash.github.io

UNIVERSITY OF OREGON

Assignment 2 is Out!

- **Assignment 2 is out!**
- Please work with your groups to understand the underlying system.
- Assignment 2 report **deadline** → April 30, 2026 at 11:59pm.

Last Class

- Last class we looked at:

- Deterministic Databases
- Calvin

Evaluating Large-Scale System

- What is meant by evaluating a large-scale system?

Evaluating Large-Scale System

- What is meant by evaluating a large-scale system?
  - Evaluating a system means testing it on different parameters and metrics.

- Why should you evaluate your system?

Evaluating Large-Scale System

- What is meant by evaluating a large-scale system?
  - Evaluating a system means testing it on different parameters and metrics.

- Why should you evaluate your system?
  - Evaluation helps to prove a system's performance and capabilities.

Evaluation Comparison

- What should you compare your system against?

Evaluation Comparison

- What should you compare your system against?
  - **Prior Works!**
  - Any state-of-the-art system in production!

Evaluation Comparison

- What should you compare your system against?
  - **Prior Works!**
  - Any state-of-the-art system in production!
  - The emphasis is on **state-of-the-art:**
    - Decades old system may not include recent design strategies or optimizations.
  - Several state-of-the-art systems have accompanied manuals, published papers, and/or documentations.

Evaluation Inputs

- For evaluating different systems, what is the best input to these systems?

Evaluation Inputs

- For evaluating different systems, what is the best input to these systems?
  - Standard Benchmark Suites
  - Standard Datasets

Evaluation Inputs

- For evaluating different systems, what is the best input to these systems?
  - Standard Benchmark Suites
  - Standard Datasets
  - Synthetic datasets/ benchmarks not preferred unless none exists!
    - If so, try publishing your synthetic dataset/benchmark for public scrutiny.

Evaluation Methodology

- How to know if your evaluation methodology is **correct**?

Evaluation Methodology

- How to know if your evaluation methodology is **correct**?
  1) Create a blind **mental map**.
     - What is your expectation from your system?
     - How do you expect graphs to look like?
       - For example: Do you expect linear, quadratic, or exponential trend.
     - How much performance gains do you expect.

Evaluation Methodology

- How to know if your evaluation methodology is **correct**?
  1) Create a blind **mental map**.
     - What is your expectation from your system?
     - How do you expect graphs to look like?
       - For example: Do you expect linear, quadratic, or exponential trend.
     - How much performance gains do you expect.
  2) Check the mechanisms **prior works** use to evaluate their systems.
     - Use same evaluation metrics and graphs as prior works.

Evaluation Metrics

- What are different metrics to consider during evaluation?

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Throughput** → Number of transactions processed per second.
  - Higher the throughput, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Throughput** → Number of transactions processed per second.
  - Higher the throughput, more desirable the system.

- **Latency** → Time between client sent a transaction to client received a response.
  - Lower the latency, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Throughput** → Number of transactions processed per second.
  - Higher the throughput, more desirable the system.

- **Latency** → Time between client sent a transaction to client received a response.
  - Lower the latency, more desirable the system.

- **P99** (99 percentile latency) → Time in which 99 percent of requests are completed.
  - Lower the latency, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Throughput** → Number of transactions processed per second.
  - Higher the throughput, more desirable the system.

- **Latency** → Time between client sent a transaction to client received a response.
  - Lower the latency, more desirable the system.

- **P99** (99 percentile latency) → Time in which 99 percent of requests are completed.
  - Lower the latency, more desirable the system.

- **Bandwidth** → Network bandwidth consumed during an experiment.
  - Ideally, the system should try to utilize as much available bandwidth

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Energy** → Energy or power consumed by the system.
  - Lower the energy, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Energy** → Energy or power consumed by the system.
  - Lower the energy, more desirable the system.

- **Compute** → **C**omputational resources (e.g. cores, memory, GPUs) used.
  - Smaller the computational resources, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Energy** → Energy or power consumed by the system.
  - Lower the energy, more desirable the system.

- **Compute** → **C**omputational resources (e.g. cores, memory, GPUs) used.
  - Smaller the computational resources, more desirable the system.

- **Accuracy** → Some systems may state % accuracy of the system/algorithm.
  - Higher the accuracy, more desirable the system.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Failure Recovery** → Time to recover from failures.
  - Desirable for a system to have small time to recover from failures.

Evaluation Metrics

- What are different metrics to consider during evaluation?

- **Failure Recovery** → Time to recover from failures.
  - Desirable for a system to have small time to recover from failures.

- **Failure Resistance** → Resources needed to prevent/guard against failures.
  - Desirable for a system to require fewer resources to guard against failures.

Evaluation Measurements

- When evaluating a system, what tasks prove the reliability of the results?

Evaluation Measurements

- When evaluating a system, what tasks prove the reliability of the results?
  1) **Multiple Runs:**

Evaluation Measurements

- When evaluating a system, what tasks prove the reliability of the results?
  1) **Multiple Runs:**
     - Always run each experiment multiple times (say three times).
     - Average the results of these multiple runs.
     - Helps to minimize noise.

Evaluation Measurements

- When evaluating a system, what tasks prove the reliability of the results?
  1) **Multiple Runs:**
     - Always run each experiment multiple times (say three times).
     - Average the results of these multiple runs.
     - Helps to minimize noise.
  2) **Experiment Phase:**

Evaluation Measurements

- When evaluating a system, what tasks prove the reliability of the results?
  1) **Multiple Runs:**
     - Always run each experiment multiple times (say three or five times).
     - Average the results of these multiple runs.
     - Helps to minimize noise.
  2) **Experiment Phase:**
     - Every system has a warmup phase, actual experiment phase, and a cool down phase.
     - Do not collect your results during the warmup and cool down phase.
     - Give sufficient time for these phases to run.

System Plots

Scalability

Scalability

- The goal of scalability plots is to tell how does your system/algorithm perform with addition of resources.

- Following are some possible designs:

Scalability

- The goal of scalability plots is to tell how does your system/algorithm perform with addition of resources.
- Following are some possible designs:
- In a **replicated system**, on increasing the number of replicas (1,2,3,…) does the system throughput, client and P99 latency increase or decrease.
- In a **sharded system**, on increasing the number of shards (1,2,3,…) does the throughput for intra-shard (or inter-shard) transactions increase or decrease

Resource

- The goal of resource plots are to tell how does your system/algorithm perform with change of resource.

Resource

- The goal of resource plots are to tell how does your system/algorithm perform with change of resource.
- Does the increase in resources of a type, cause an increase or decrease in the system throughput, client and P99 latency.
- If the resources are kept constant, then what is the impact on system throughput, client and P99 latency.

Different Types of Resources

Different Types of Resources

- Cores
- Memory
- Network Bandwidth
- Clients
- Nodes

Deployment

- In several systems, deployment of nodes/clients matter.

Deployment

- In several systems, deployment of nodes/clients matter.
- Nodes/clients deployed in same datacenter (or location) may incur less cost than geographical deployment.
- Is it important to study geographical deployment?

Deployment

- In several systems, deployment of nodes/clients matter.
- Nodes/clients deployed in same datacenter (or location) may incur less cost than geographical deployment.
- Is it important to study geographical deployment?
- Geographical deployment presents a view of real-world deployment, for example, to handle failures, some systems place nodes across the globe.

Code Impact

- Should you take into consideration the impact of code?

Code Impact

- Should you take into consideration the impact of code?
  - Yes, because code fragments can introduce performance bottlenecks!

- What are different code aspects to consider?

Code Impact

- Should you take into consideration the impact of code?
  - Yes, because code fragments can introduce performance bottlenecks!

- What are different code aspects to consider?
  - Optimizations
  - Iterations
  - Hyper-parameters
  - Queue Size
  - Locks and Critical sections
  - Available Parallelism (embarrassingly parallel program or shared resources)

Input Impact

- Should you take into consideration the impact of input?

Input Impact

- Should you take into consideration the impact of input?
  - Yes, a client input can impact the logical flow, which in turn can impact systems performance.

- Input consideration examples?

Input Impact

- Should you take into consideration the impact of input?
  - Yes, a client input can impact the logical flow, which in turn can impact systems performance.

- Input consideration examples?
  - Data accessed by client transactions → Can cause conflicts.
  - Order in which transactions arrive.

Lets Look at Calvin