SAPIENZA
UNIVERSITÀ DI ROMA

Networking for big data
Data centers

Lecturer: Andrea Baiocchi

email: andrea.baiocchi@uniroma1.it

Department of Information Engineering, Electronics, and Telecommunications
SAPIENZA University of Rome

a.y. 2023/2024

Challenge #1: topology design

Definitions

- A graph is a couple $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a finite, non-empty set and $\mathcal{A}$ is a subset of $\mathcal{N} \times \mathcal{N}$.
  - A very useful model for representing network connectivity as seen at a given architectural level.
    - The meaning of nodes and arcs depends on the considered architectural level, e.g., nodes can be routers and arcs subnets at IP level or switches and physical links at MAC layer.
  - Elements of $\mathcal{N}$ are called nodes (also vertices).
  - Elements of $\mathcal{A}$ are called arcs (also edges, links).
- A graph is said to be undirected if $(i, j) \in \mathcal{A} \Rightarrow (j, i) \in \mathcal{A}$.
- We define:

$$\begin{aligned} n &= \text{number of nodes;} \\ k &= \text{number of arcs.} \end{aligned}$$

Adjacency matrix

- The $n \times n$ matrix $\mathbf{A}$ whose entries a_{ij} are defined as:

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$$

  is called the adjacency matrix of the graph.
- The adjacency matrix of an undirected graph is symmetric.
- The sum of rows (columns) of $\mathbf{A}$ is the out-(in-)degree of the graph nodes, i.e., the number of arcs outgoing from (incoming into) a graph node.

Connectivity

- A path between nodes s and t is an ordered sequence of nodes and arcs connecting the two nodes, i.e., $(s, k_1), (k_1, k_2), \ldots, (k_{\ell-1}, t)$ is a path of length ℓ.
- A graph is said to be connected if there exists a path connecting any two nodes.

Criteria to check graph connectivity for undirected graphs.

- Starting from an arbitrary node, all graph nodes can be visited (BSF algorithm).
- The adjacency matrix $\mathbf{A}$ is irreducible.
- The second smallest eigenvalue of the graph Laplacian matrix $\mathbf{L}$ is positive.

Breadth-First-Search algorithm

Finds all paths form a given node.

Given a graph 'Graph' and a node 'root' in it:

```
for each node n in Graph:
  n.distance = INFINITY
  n.parent = NIL
endfor
create empty queue Q
root.distance = 0
Q.enqueue(root)
while Q is not empty:
  current = Q.dequeue()
  for each node n that is adjacent to current:
  if n.distance == INFINITY:
   n.distance = current.distance + 1
   n.parent = current
   Q.enqueue(n)
  endif
endwhile
```

Irreducibility

- Say nodes of the graph are labeled with positive integers $1, \dots, n$.
- The adjacency matrix is reducible if and only if there exists a re-labeling of nodes such that the matrix can be partitioned as follows:

  $$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

  where $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are square blocks and $\mathbf{0}$ is a block of zeros.
- It can be shown that an $n \times n$ matrix $\mathbf{A}$ is irreducible if

  $$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^{n-1} > \mathbf{0}$$

Laplacian

- The Laplacian of a graph in an $n \times n$ matrix $\mathbf{L}$ defined as follows:
  - $L_{ii} = d_i$, where d_i is the out-degree of node i;
  - $L_{ij} = -1$ if and only if $(i,j) \in \mathcal{A}$.
- We can write:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

- $\mathbf{L}$ is a symmetric matrix, if the graph is undirected.
- The sum of each row of $\mathbf{L}$ is 0. Therefore 0 is always an eigenvalue of $\mathbf{L}$

Eigenvalues of the Laplacian

- The incidence matrix of a graph in an $2k \times n$ matrix $\mathbf{M}$ defined as follows:
  - $M_{\ell i} = 1$, if the directed arc ℓ is outgoing from node i.
  - $M_{\ell i} = -1$, if the directed arc ℓ is incoming into node i.
  - $M_{\ell i} = 0$, otherwise.
- It can be verified that:

$$\mathbf{L} = \frac{1}{2}\mathbf{M}^T\mathbf{M}$$

- Hence $\mathbf{L}$ is a positive semi-definite matrix, i.e., $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for any vector $\mathbf{x}$.
- Positive semi-definite and symmetric → the eigenvalues of $\mathbf{L}$ are real and non-negative.

Connectivity through Laplacian

Theorem (From algebraic graph theory)

For an undirected graph the number of connected components is equal to the algebraic multiplicity of the smallest eigenvalue of the graph Laplacian.

Corollary. An undirected graph is connected if and only if the smallest eigenvalue of the graph Laplacian is simple, i.e., the second smallest eigenvalue of the Laplacian is positive.

- The property stated in the Corollary can be used to assess the connectivity of a given graph.
- Let $\eta_1 = 0 \leq \eta_2 \leq \cdots \leq \eta_n$ denote the n real, non-negative eigenvalues of $\mathbf{L}$.
- The graph is connected if and only if $\eta_2 > 0$.

Random graphs

We consider only symmetric graphs.

- **Erdős-Rényi random graphs**: two models are possible.
  - $G(n, p)$ model: a graph is constructed by connecting nodes randomly.
  - An arc is included in the graph with probability p independently of any other arc.
- ***r*-regular random graph**: a graph selected from $\mathcal{G}_{n,r}$, which denotes the probability space of all r-regular graphs on n vertices, where $3 \leq r < n$ and nr is even.
  - An r regular graph is a graph where each vertex has the same number r of neighbors.
  - r regular graphs for $r = 0, 1, 2$ are trivial.

Assignment - Part 1

1. Use library scripts to generate p-ER random graphs and r-regular random graph. Let K denote the number of nodes.
2. Write a script to check the connectivity of a given graph.
   - algebraic method 1 (irreducibility);
   - algebraic method 2 (eigenvalue of the Laplacian matrix);
   - breadth-first search algorithm.
3. Compare the complexity as a function of K of the methods above by plotting curves of a complexity measure vs K.
4. Let $p_c(\mathcal{G})$ denote the probability that a graph $\mathcal{G}$ is connected. By running Monte Carlo simulations, estimate $p_c(\mathcal{G})$ and produce two curve plots:
   - $p_c(\mathcal{G})$ vs. p for Erdős-Rényi graphs with $K = 100$.
   - $p_c(\mathcal{G})$ vs. K, for $K \leq 100$, for r-regular random graphs with $r = 2$ and $r = 8$.

Computation job: local versus distributed run

- Let us consider a server A that has to run a computation job that takes a time $T_0 + X$, where T_0 is a fixed set-up time and X is a random variable with mean $\mathrm{E}[X]$.
- The job is applied to a file of data of size L_f.
- The output of the task is an amount of data L_o.
- If the job is split into N parallel tasks, that are run over N servers (other than A), then
  - Each task takes a time $T_0 + X_i$, where X_i is a negative exponential random variable with mean $\mathrm{E}[X_i] = \mathrm{E}[X]/N$.
  - Each server receives an amount of input data L_f/N.
  - The amount of output data produced by each server is a random variable $L_{o,i}$, uniformly distributed in $[0, 2L_o/N]$.

Communications among servers

- Data is transferred to and from server i $(i = 1, \dots, N)$ via a TCP connection between server A and server i, having average throughput given by

  $$TH_i = C \frac{1/T_i}{\sum_{j=1}^{N} 1/T_j}$$

  where T_i is the RTT between the origin server A and server i and C is the capacity of each link of the DC network.
- $T_i = 2\tau h_i$, where h_i is the number of hops between server A and server i.
- The N servers selected to split the job are *the N closest servers to A in the given Data Center network topology*.
- In the following, we assume the N servers are numbered from 1 to N so that $h_1 \leq h_2 \leq \cdots \leq h_N$.

Metrics

- The **response time** R is the time elapsing since when the job is submitted to server A until the time that the output of the job is available at server A.
- In case of splitting of the job into N tasks, the output of the job is available only when *all* servers have delivered their output files to server A.

- The **Job running cost** S is defined as

$$S = \mathrm{E}[R] + \xi\,\mathrm{E}[\Theta]$$

where $\mathrm{E}[\Theta]$ is the average server time used to run the job.

- Θ is the time that server A is used, if the job runs locally on A. Otherwise, Θ is the sum of the times that all N servers are used to run their respective tasks.

Additional details

- In building DC network topologies assume that switches with n ports are used.
  - Fat-Tree DC network topology is defined as usual.
  - Jellyfish DC network topology is build assuming that $r = n/2$, i.e., each switch devotes $n/2$ ports for connections to local servers and the other $n/2$ ports for connections to other switches.
  - The two topologies must be built so that they have the same number of servers $(n^3/4)$.
- When communicating data back and forth from server A to the N servers, using TCP connections, there is some overhead involved. Specifically, the amount of bits sent through the TCP connection is given by the sum of the application job data plus an additional overhead, which is a fraction f of the original data.

Numerical values

- $C = 10$ Gbit/s.
- $\tau = 5\ \mu$s.
- $L_f = 4$ TB.
- $L_o = 4$ TB.
- $\mathrm{E}[X] = 8$ hours.
- $T_0 = 30$ sec.
- $\xi = 0.1$.
- $f = 48/1500$.
- $n = 64$.

Assignment - Part 2

1. Give a concise and accurate formal statement of the algorithm you use to evaluate the mean response time (i.e., how you conduct the statistical experiment to collect samples of R).
2. Plot the mean response time $\mathrm{E}[R]$ as a function of N for N ranging between 1 and 10000.
    - Let R_{baseline} be the response time in case only server A is used, i.e., the job is run locally on A.
    - In the plot normalize $\mathrm{E}[R]$ with respect to $\mathrm{E}[R_{\text{baseline}}]$.
3. Plot the Job running cost S as a function of N for N ranging between 1 and 10000.
    - Let S_{baseline} be the Job running cost in case only server A is used, i.e., the job is run locally on A.
    - In the plot normalize S with respect to S_{baseline}.
4. Give the numerical value of the optimal number of servers (the one that minimizes the job running cost).
5. Explain in a concise and accurate way the results you have got. Highlight the takeaways of your analysis.

Delivery of the assignment

The delivery of the assignment consists of a **max 4 pages written report** in the form of a pdf file (Font size: 12 pt).

1. **PAGE 1** – Cover Page: nickname of your group, your given and family names, and enrollment numbers.
2. **PAGE 2** – Three curve plots plus explanations: (i) Complexity versus number of nodes K, for the three connectivity checking algorithms; (ii) Probability of a connected ER random graph as a function of p for $K = 100$ nodes; (iii) Probability of a connected r-regular random graph as a function of K for $r = 2$ and 8.
3. **PAGE 3-4** – (i) Formal statement of algorithm for the evaluation of the mean response time; (ii) plot of the normalized mean response time as a function of N; (iii) plot of the normalized Job running cost as a function of N; (iv) optimal value of N that minimizes S; (v) analysis of results and takeaways. (note that this part must be done for both Fat-Tree and Jellyfish topologies)

Accuracy and conciseness of texts, as well as accuracy, readability and quality of the graphs will be fundamental in the evaluation.