Final Project

• In place of a final exam
• “Ratov” – programming
• In the field of network /service management with emphasis on:
  – Efficiency
  – Cloud computing
  – Possible use of **Amazon Web Services** (AWS) or **Microsoft Azure**
• How much work?
Placement impact on intra-service communication

- **Goal:**
  - measure the ability of VMs within a service to communicate and try to evaluate the impact of the VM placement on this ability

- **More detailed description:**
  - create a service with 3 VMs such that the traffic matrix varies (e.g. Map-Reduce workflow)
  - measure the performance (ability to send / receive the required amount of data)
  - change the VM roles (in-house relocation) and measure again
  - create a “self optimized mechanism” that put the VMs in the “best” place

- **Environment**
  - EC2
Performance of the HP-HP load balancing scheme over heavy tailed load distribution

• Goal:
  – measure the performance of the HP-HP load balancing scheme where the load distribution is heavy tailed

• More detailed description:
  – create an in house event driven simulation for the HP-HP load balancing scheme as described in [1].
  – generate workload using heavy tailed distribution and evaluate the expected performance

• Environment
  – JAVA or C++

Amir Nahir, Ariel Orda, Danny Raz, Technion, CNSM 2012.
Reactivity and Cost in Cloud Elasticity

• Goal:
  – evaluate cost and reactivity of elastic mechanisms in cloud where the main idea is to maintain the unused VM in a “sleep” mode instead of removing them and thus reducing the “reaction time” when the load increases

• More detailed description:
  – in this project we will install VMs running a benchmark application on a server and measure the performance of the system when load increases
  – we also will measure the time it takes to activate a new VM with its application and the cost of having the VM alive in a “sleep” mode

• Environment
  – Open Stack and simulations
Introducing delay into the \textit{Join-idle-queue} model

\begin{itemize}
  \item \textbf{Goal:}
    \begin{itemize}
      \item evaluate the performance of the Join-idle-queue model in the present of network latency
    \end{itemize}
  \item \textbf{More detailed description:}
    \begin{itemize}
      \item the Join-idle-queue model is a two tier architecture for efficient load balancing described in [1]
      \item in this model, when servers become idle they explore several queues and join the shorter queue.
      \item in the original model this queue selection process does not take time
      \item in this project we will simulate this model when it does take time to monitor the size of the queues and evaluate the expected performance in this case
    \end{itemize}
  \item \textbf{Environment}
    \begin{itemize}
      \item Java or C++
    \end{itemize}
\end{itemize}

Efficient monitoring of aggregated traffic in an SDN environment

• Goal:
  – implement an efficient monitoring infrastructure for aggregated flows in a software defined Network environment

• More detailed description:
  – create an emulated SDN environment using Mininet
  – create aggregated flows in this environment using the OpenFlow controller.
  – implement an efficient aggregated monitoring infrastructure (from [1]) over this environment
  – evaluate the performance

• Environment
  – Mininet

Questions?