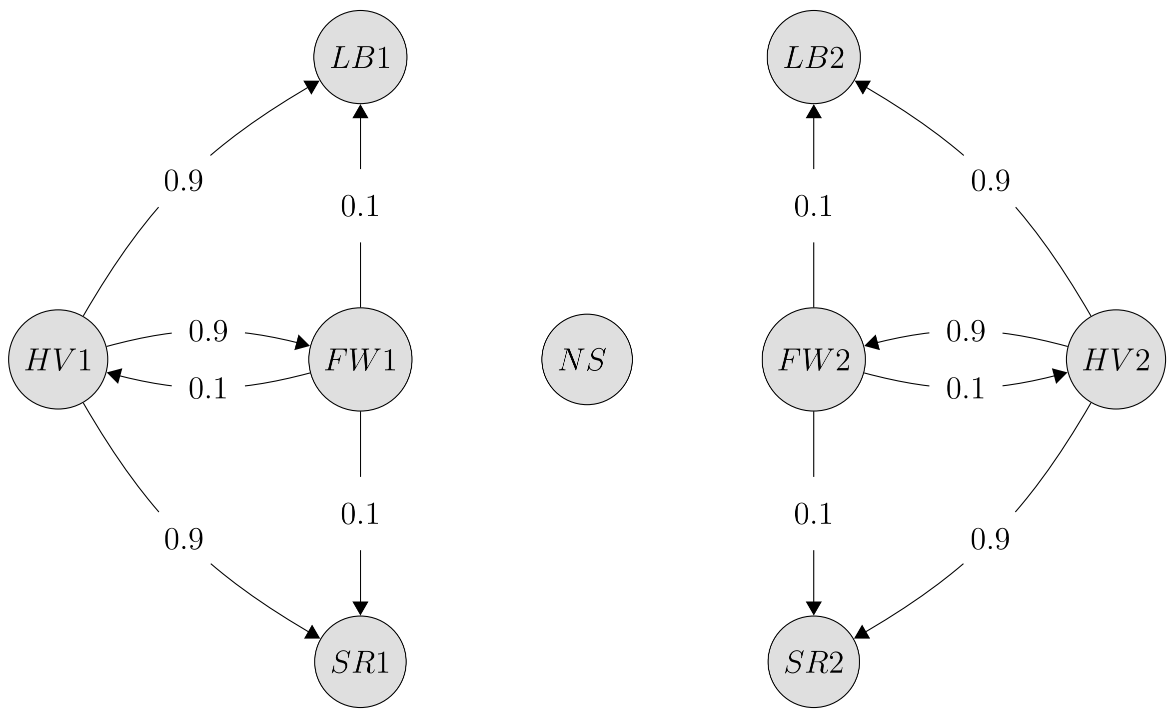

Figure 1: Cloud-computing architecture

We now consider a model depicting a cloud-computing architecture shown in Fig. 1. There is a directed edge from component type i to component type j if $j \in \Gamma_i$, and the label on the edge is $\phi_{i,j}$. NS represents a network switch; $LB1$ and $LB2$ are two types of load balancers; $FW1$ and $FW2$ denote two types of firewalls; $HV1$ and $HV2$ signify two types of hypervisors; and $SR1$ and $SR2$ are two types of server racks. The two types mean different components in the sense that we use in the paper, i.e., they can be the same model of component but they are different types because they are in different parts of the system. This organization of components specifically shows a firewall "sandwich" as described in [1].

We set $r_{NS} = r_{LB1} = r_{LB2} = r_{FW1} = r_{FW2} = r_{HV1} = r_{HV2} = 2$ and $r_{SR1} = r_{SR2} = 3$. We assume the system is operational if at least υ_i components of each type i are up. We need at least two server racks of each type $SR1$ and $SR2$ to be up to be able to parallel process across server racks. For other component types, we require at least one component of each type of the system to be up for the system to remain operational. Thus, $\upsilon_{SR1} = \upsilon_{SR2} = 2$

and $\upsilon_{NS} = \upsilon_{LB1} = \upsilon_{LB2} = \upsilon_{FW1} = \upsilon_{FW2} = \upsilon_{HV1} = \upsilon_{HV2} = 1$. Additionally, our system operates in two environments: high demand ($e = 0$) and low demand ($e = 1$). This gives us a state space of size 69984. Solving this using the previous version of DECaF takes xxx seconds, whereas using the new version takes xxx seconds.

We work with a system such that all component types that are located to the left of NS (i.e., types with a "1" in their names) handle SMTP requests only, whereas all component types located to the right of NS (i.e., types with a "2" in their names) handle HTTP requests only. Thus, both groups of components (i.e., to the left and to the right of NS) need to be operational for the system to be able to handle HTTP and SMTP requests. The decoupling ensures that hardware failure on one side does not immediately propagate to the other.

As stated in [2], "hypervisors almost always cause other system components to fail and certainly cause server racks to fail because of state corruption." Thus, we assume $\phi_{HVi,LBi} = \phi_{HVi,FWi} = \phi_{HVi,SRi} = 0.9$ for $i = 1, 2$. We also assume $\phi_{FWi,LBi} = \phi_{FWi,HVi} = \phi_{FWi,SRi} = 0.1$ for $i = 1, 2$.

To estimate the component failure rates, we assume that in a high-demand (resp., low-demand) environment, a hardware component is going to fail on average in twice (resp., eight times) the time of its warranty. The time unit is hours. Typical commercial network switches, load balancers and hypervisors have warranties of 90 days (2160 hours). Thus, $\lambda_{NS,0} = \lambda_{LB1,0} = \lambda_{LB2,0} = \lambda_{HV1,0} = \lambda_{HV2,0} = 1/4320$ and $\lambda_{NS,1} = \lambda_{LB1,1} = \lambda_{LB2,1} = \lambda_{HV1,1} = \lambda_{HV2,1} = 1/17280$. Commercial server racks often have 3-year warranties giving us $\lambda_{SR1,0} = \lambda_{SR2,0} = 1/52560$ and $\lambda_{SR1,1} = \lambda_{SR2,1} = 1/210240$. Since firewalls fail due to numerous reasons such as software bugs or from targeted attacks, it is difficult to find a single representative failure rate. Thus we assume a software firewall fails on average once in five years, and set $\lambda_{FW1,0} = \lambda_{FW2,0} = \lambda_{FW2,1} = \lambda_{FW1,1} = 1/43800$.

As given in [1], failed hardware components are repaired with a rate of $1/24$ regardless of the environment, i.e., we can replace one component a day on average. Firewalls being non-hardware components need to be rebooted after they fail, and this occurs with a rate of 10, which is also taken from [1]. The environment switches once every 12 hours on average, giving us the environment transition rates $\nu_{0,1} = \nu_{1,0} = 1/12$.

References

[1] S. Goddard, R. Kieckhafer, and Y. Zhang. An unavailability analysis of firewall sandwich configurations. In *High Assurance Systems Engineering, 2001. Sixth IEEE International Symposium on*, pages 139–148, 2001.

[2] M. Ye and Y. Tamir. Rehype: Enabling vm survival across hypervisor failures. *ACM SIGPLAN Notices*, 46(7):63–74, 2011.