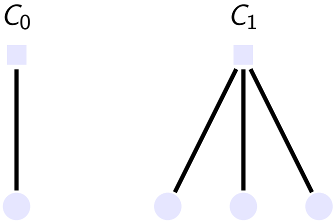
Load Migration Scheduling

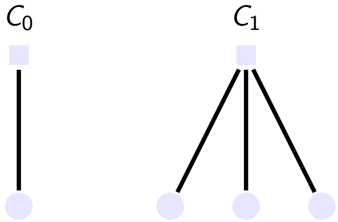
Mohammad Amin Beiruti, Yashar Ganjali

October 18, 2021

Load Migration



Load Migration



- Controllers can become overloaded
- Want to migrate switches from heavily loaded controllers to controllers with lower load
- Investigate migrating multiple switches

Constraints

Controller resource constraints

- Migration requires processing + memory at src + dst controllers
- src stops processing messages and must buffer them
- Limits number of concurrent migrations at controllers

Constraints

Controller resource constraints

- Migration requires processing + memory at src + dst controllers
- src stops processing messages and must buffer them
- Limits number of concurrent migrations at controllers

QoS constraints

- Messages between controller and switch paused during migration
- Don't want to interrupt network services
- Limit concurrent migration of switches in a particular group

Inputs

Assume k migrations $M = \{m_1, ..., m_k\}$

Migration m_i

- moves switch m_i^{SW}
- from controller m_i^{src}
- to controller m_i^{dst}
- has weight m_i^{w}

Inputs

Assume k migrations $M = \{m_1, ..., m_k\}$

Matrix P is kxn with

$$P_{ij} = \begin{cases} 1 & \text{if controller } c_j \text{ is the source for migration } m_i \\ 0 & \text{otherwise} \end{cases}$$

Matrix Q is kxn with

$$Q_{ij} = \begin{cases} 1 & \text{if controller } c_j \text{ is the destination for migration } m_i \\ 0 & \text{otherwise} \end{cases}$$

Constraints

Controller resource constraints

- Controller c_i has capacity a_i
- Combines memory and computational capacity

QoS constraints

- l groups of switches $\mathcal{G} = \{g_1, ..., g_l\}$
- group g_i can have at most α_i concurrent migrations

Binary matrix W

$$W_{ij} = \begin{cases} 1 & \text{if migration } m_i \text{ is in group } g_j \\ 0 & \text{otherwise} \end{cases}$$

Variables

Assume at most R rounds needed for all migrations

A is a kxR matrix where

$$A_{ij} = \begin{cases} 1 & \text{if migration } m_i \text{ occurs during round } j \\ 0 & \text{otherwise} \end{cases}$$

Objective

Want to minimize the number of rounds needed

Idea: Penalize later rounds more than earlier rounds

$$\min \sum_{i=1}^{k} \sum_{j=1}^{R} (A_{ij}) e^{j}$$

Penalties increase in exponential fashion

Model

$$\min \sum_{i=1}^{k} \sum_{j=1}^{R} (A_{ij}) e^{j}$$

s.t.

$$\sum_{r=1}^{R} A_{ir} = 1 \qquad \forall\, 1 \le i \le k$$

$$\sum_{t=1}^{k} (A_{tj})(m_t^w) Q_{ti} \le a_i \qquad \forall\, 1 \le i \le n, 1 \le j \le R$$

$$\sum_{i=1}^{k} (A_{ij}) W_{iu} \le \alpha_u \qquad \forall\, 1 \le j \le R, 1 \le u \le l$$

$$A_{ij} \in \{0, 1\} \qquad \forall\, 1 \le i \le k, 1 \le j \le R$$

Model

$$\sum_{r=1}^{R} A_{ir} = 1$$

Says each migration is scheduled in one round

Model

$$\sum_{t=1}^{k} (A_{tj})(m_t^w) Q_{ti} \leq a_i$$

Encodes resource constraints on destination controller

To incorporate failure resiliency, can replace with

$$\sum_{t=1}^{k} (A_{tj})(m_t^w)(P_{ti} + Q_{ti}) \leq a_i$$

Model

$$\sum_{i=1}^{k}(A_{ij})W_{iu} \leq \alpha_u$$

Encodes QoS constraints

Next Steps

- Show NP-hard via reduction from graph colouring
- Provide lower bound through LP relaxation
- Exponential objective becomes intractable for large number of rounds
  - Instead use integer variables for round number
- Need better motivation for QoS constraints
- Extensive experiments

Generalizations

- Destination may not be fixed
- New migrations could arrive / leave in later rounds
- Resource requirement could dynamically change
- Migrations could also require varying amounts of time
  - Strong connection to $R||C_{max}$ scheduling instance
  - But more complicated
- Look at online / stochastic scheduling variant