Maximum Elastic Scheduling based on the Hose Model

基于软管模型的最大弹性调度

Jie Wu (吴杰)
Temple University (天普大学)
1. AI Takeoff

- Deep Blue
  - ICPP'96 panel: F. -H. Hsu (許峰雄) talked about LB instead.

- HPC-AI convergence
  - AI blackbox (黑箱子)
  - However, DARPA: **Explainable AI (XAI)**
    - Produce more explainable models
    - Enable human users to understand

- Back to fundamentals
  - Direct algorithmic/combinatoric solutions
  - A scheduling problem related to **maximum elasticity**
A Simple Illustration

- Given a cable connection in a graph, each household has an occupancy limit and each cable section has bandwidth limit.

- What is the maximum total occupancy that can support all possible simultaneous pairwise telephone conversations (hose model)?

- What is the schedule with the maximum elasticity (i.e., maximum uniform growth in occupancy)?

hose model (软管模型): statistical multiplexing
2. Model and Formulation

- **Maximum Admissible Load (MAL)**: 最大容许负载
  - Provisioning MAL of VMs in PMs for hose-model-based DCNs

- **Maximum Elastic Scheduling (MES)**: 最大弹性调度
  - A task assignment of a given load (< MAL) with potential maximum uniform growth in computation and communication

How to define elasticity?
A Simple 2-Level Tree

On DCN (数据中心网络), DCN cloud, or Internet cloud

\[ G = (V, E), \text{ } V: \text{ server (服务器)} \text{ or switch (交换器)}, \text{ } E: \text{ link (链路)} \]

Each VM has 1B Gbps aggregate bandwidth
How to Solve It (Polya)

If you can’t solve a problem, then there is an easier problem you can solve: find it

- Tree topology (typical DCN)

**Direct solutions**
- Shortest path problem (最短路径)
  - LP solution
  - Greedy solution: Dijkstra algorithm
- Maximum elastic scheduling (最大弹性调度)
  - LP solution
  - Greedy solution: Two-phase sweep
LP Solution

maximize \( e \) \hspace{1cm} \text{(1)}

\[
\begin{align*}
\text{s.t. } & \quad e \leq \min_i (1 - \frac{x_i}{N_i}) \quad \text{and} \quad x_i \leq N_i \quad \text{for } \forall i \\
& \quad e \leq \min_j (1 - \frac{y_j}{L_j}) \quad \text{and} \quad y_j \leq L_j \quad \text{for } \forall j \\
& \quad y_j = \min \left[ \sum_i \mu_{ij} x_i, \sum_i (1 - \mu_{ij}) x_i \right] \quad \text{for } \forall j
\end{align*}
\]

Eq. (1): objective function
Eq. (2) and Eq. (3): constraints on nodes \((N_i)\) and links \((L_j)\)
Eq. (4):

\( i \)th node on \( j \)th link
LP Solution (cont’d)

Variables: 3n−1
- n: # of leaf nodes
- 2n−2: # of links
- 1: objective function e

Constraints: 10n−8
- Eq. (6): 2n
- Eq. (7): 4n − 4
- Eq. (8): 4n − 4

Inefficiency: Simplex or Eclipse

maximize \[ e \]
subject to
\[ e \leq \min_i (1 - \frac{x_i}{N_i}) \text{ and } x_i \leq N_i \text{ for } \forall i \] (6)
\[ e \leq \min_j (1 - \frac{y_j}{L_j}) \text{ and } y_j \leq L_j \text{ for } \forall j \] (7)
\[ y_j \leq \sum_i \mu_{ij} x_i \text{ and } y_j \leq \sum_i (1 - \mu_{ij}) x_i \text{ for } \forall j \] (8)
3. Two-Phase Sweep Solutions

**Up phase:** Cal. MAL of a 2-level subtree

\[ \min\{N_L, L_L\} + \min\{N_R, L_R\} \]

**Down phase:** Given a load \( N(<\text{MAL}) \) at root

- **Left**
  \[ \min\{N_L, L_L\}/N \]

- **Right**
  \[ \min\{N_R, L_R\}/N \]
Why Simple Solution May Fail?

A simple solution

However

How to find the OPT Solution?

\[ 10 + 6 = 16 \]

\[ 10 + 6 > 14 \]
How to Calculate?

Hose-model tree orientation

- Directed tree: Link orientation is based on the selected root.
- Find a root with the maximum summation of branch values.
Optimal Solution

Insights

- Apply the simple solution to different orientations.
- Select the best orientation.

MAL at the left leaf

MAL at the right leaf

MAL at the center
Distributed Implementation

At each node

<table>
<thead>
<tr>
<th>Step</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
<th>$v_6$</th>
<th>$v_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>send $\min{5,4}+\min{6,7}=10$ to $v_2$</td>
<td>send $\min{5,4}=6$ to $v_1$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Step 2</td>
<td></td>
<td></td>
<td></td>
<td>send $\min{6,2}$ to $v_2$</td>
<td>send $\min{6,7}=10$ to $v_3$</td>
<td>send $\min{6,6}=4$ to $v_3$</td>
<td>send $\min{6,6}=4$ to $v_3$</td>
</tr>
<tr>
<td>Step 3</td>
<td>send $\min{6,6}=6$ to $v_2$</td>
<td>send $\min{6,8}=8$ to $v_3$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Step 4</td>
<td>send $\min{6,8}=12$ to $v_4$</td>
<td>send $\min{6,8}=12$ to $v_4$</td>
<td>send $\min{8,6}=10$ to $v_5$</td>
<td>send $\min{8,6}=10$ to $v_6$</td>
<td>send $\min{8,6}=10$ to $v_6$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAL</td>
<td>$\min{10,8}+\min{6,6}=14$</td>
<td>$\min{5,4}+\min{6,7}+\min{8,6}=16$</td>
<td>$\min{6,2}+\min{4,5}+\min{8,6}=12$</td>
<td>$\min{12,4}+\min{5,\infty}=9$</td>
<td>$\min{10,7}+\min{6,\infty}=13$</td>
<td>$\min{10,2}+\min{6,\infty}=8$</td>
<td>$\min{8,5}+\min{4,\infty}=9$</td>
</tr>
</tbody>
</table>
4. Properties and Extensions

**Theorem 1:** The up-phase determines the MAL.

**Theorem 2:** The two-phase solution generates a schedule with maximum elasticity.

**Theorem 3:** The two-phase solution uses $2\log n + 1$ parallel steps. The computation complexity is $5(n-1)$, and the communication complexity is $4(n - 1)$. 
Extensions

- General trees
  - Any k-nary trees

- Optimal simple solution
  - Trees with computational-bottleneck

- Fat trees (used in DCN)
  - Still work!
5. Performance Comparisons

- **Basic setting**
  - Binary trees with levels: $k = 4, 5, \text{ and } 6$
  - Node capacity: 0 to 100 units
  - Link bandwidth: 0 to 100 GB
  - Bandwidth demand: 1 Gbps

- **Comparison algorithms**
  - Equally Distributed Placement (EDP)
  - Proportion to PM Capacities (PPMC)
  - Proportion to Physical Link Capacities (PPLC)
  - Proportion to PM and Channel Capacities (PPCC)
Binary Tree Simulation

Comparison of the elasticities
- Three comparison algorithms and PPCC
- Capacity ratio: average link capacity / node capacity

![Graphs showing comparison of elasticities for different values of k: (a) k = 4, (b) k = 5, (c) k = 6.](image)
Equal-cost multi-path routing (ECMP) with m=4 (ports)
Fat Tree Equivalence
Fat Tree Simulation

- **Settings**
  - $m = 4, 6, 8, \text{ and } 10$

- **Node capacity**
  - PM: 0 to 100 slots
  - VM comm. bandwidth: 1 Gpbs

- **Link bandwidth**
  - edge layer: [0, 10] Gbps
  - aggregation layer: [0, 15] Gbps
  - core layer: [0, 30] Gbps
Tree Testbed

- Central server: Grnlntrn
- Cisco switch: 8-port connector
- Pica8 switch: 48 ports
- Server: Dell Power Edge R210 (2.4 GHz CPU, 4 GB memory)
- Maximum link capacity: 1 Gbps

EDP 25% 25% 25% 25%
PCC 41% 24% 19% 16%
Testbed Results

- One-to-all comm.
- Stress-test on a hose: Map (comp.), shuffle (scatter/gather comm.), and reduce (comp.)

![Graph showing transmission time vs. file size for EDP and PPCC.]

- EDP: ≈1.2 min
- PPCC: ≈0.3 min
6. Conclusions

- **Models**
  - Hose model on trees

- **Elastic scheduling**
  - Maximum admissible load (MAL)
  - Maximum elastic scheduling (MES)

- **Future work**
  - Other topologies
  - Applications: Hadoop and Spark