System Model

**M** different types of jobs where each type is specified by a chain of tasks. Edge of the chain represents a precedence constraint. In the processing network, there are **J** servers.

**Definition:** A scheduling policy is throughput-optimal if it stabilizes the system when there exists a stabilizing scheduling policy.

**Goal:** Find a throughput optimal scheduling policy.

Capacity Region

**Capacity region** is determined by an LP:

\[
\text{Maximize } \delta \\
\text{subject to } \sum_{j} \left( \mu_{k,j}(k) - \rho_j \right) \forall 1 \leq j \leq J, \forall (k,j) \in \mathcal{E} \\
1 \geq \sum_{j} \rho_j \forall 1 \leq j \leq J \\
1 \geq \sum_{k,j} \rho_j \forall 1 \leq j \leq J \\
\mu_{k,j} \geq 0, \forall 1 \leq j \leq J \\
\rho_j \geq 0, \forall 1 \leq j \leq J \\
\delta > 0.
\]

Arrival rates are in the capacity region if \( \delta > 0 \)

Queuing Network

**Processing queue:** virtual queue called \((k,j)\) for type-**k** tasks which are processed at server **j**

**Communication queue:** one virtual queue called \((k,j),c\) for processed type-**k** tasks sent from server **j** to other servers

Equivalent Capacity Region

Corresponding to the queueing network, we define an optimization problem:

**Max-weight policy**:

Given the lengths of virtual queues at time \( n \), Max-weight policy allocates the allocation vectors \( p, q, u \) and \( s \), i.e.

\[
\arg \max_{p,q,u,s} \left( Q(n) \right)^T E[\Delta Q(n) | F^n] - (Q(n))^T E[\Delta Q(n) | F^n] \\
= \arg \min_{p,q,u,s} \left( Q(n) \right)^T E[\Delta Q(n) | F^n] + (Q(n))^T E[\Delta Q(n) | F^n].
\]

The Max-Weight policy is the choice of \( p, q, u \) and \( s \) that minimizes the drift of change of a Lyapunov function:

\[ V = \sum_{k,j} Q_{(k,j)}(n)^2 + \sum_{k,j} Q_{(k,j)}(n) \]

**Theorem:** Max-Weight policy for the queueing network is throughput-optimal.

Max-Weight policy is rate stable for all the arrival vectors in the capacity region presented in optimization problem.

Max-weight policy makes the underlying Markov process of the queue-lengths positive recurrent for all the arrival rate vectors that are in the interior of the capacity region.

Ongoing Work and Future Directions

- Consider jobs modeled as directed acyclic graphs (DAG) and design queueing network with low complexity.
- Simulations on the real world scenario, ex: Amazon EC2, Digital Ocean, etc.

*For more questions, please contact: chienshy@usc.edu*