

Link Scheduling Notes

mdskrzypczyk

August 2019

1 Cost functions

Objective functions to optimize with choice of link scheduling.

1.1 Minimized Weighted Idle Time

$$\min d = \sum_{i=1}^{n-1} w_{i+1} |e_i - e_{i+1}| + \max_i(e_i)(w_1 + w_{n+1}) - w_1 e_1 - w_{n+1} e_n$$

Where:

- n is the number of edges in the chain
- e_i is the time at which link i is expected to be generated
- w_i is a decoherence weight associated with node i
- \sum term represents the weighted decoherence introduced by the nodes internal to the chain (2:n)
- $\max_i(e_i)$ is the completion time of the last link
- Final terms represent the decoherence introduced on the qubits held by the end nodes as they must wait until the full link has been established to begin using

Observations:

- Preferable to have end links generated last as generating them earlier accumulates more decoherence (dependent on weights)

1.2 Minimized Total Fidelity Loss

$$\min F_{loss} = \sum_{i=1}^n F_{0,i} e^{-w_i t_i}$$

Where:

- F_{loss} is the total lost fidelity
- $F_{0,i}$ is the initial fidelity of a generated link
- w_i is the rate of decoherence of a link
- t_i is the amount of time the link decoheres for

Problems:

- Metric does not consider how the links decohere after they have been SWAPPED, w_i is a property of the originally generated link, not the nodes holding the qubits

2 Greedy Heuristics

- Place links that take less time to generate ahead of current link and links that take less time behind the current link.
- Place links that have a higher fidelity by the time both links should be done earlier in the schedule

3 Coordination

3.1 Timeslot Scheduling

Share schedules of nodes along path and insert slots for when time should be spent generating a link.

3.1.1 Execution Techniques

- Get individual machine schedules.
 - Schedule cannot be represented infinitely into the future (messages are finite in size, base size on the max time of generation request?)
 - Need an efficient representation that permits taking intersections for creating link schedules
 - Prune slots that are occupied or too short for generating a given link. "Too short" can have a few meanings, below average or below some fraction of average time to generate. (Check probability distribution, geometric?)
 - Represent schedule for each communication qubit on the link
- Take an intersection to form availability schedules for each link.
 - How can we take into account several communication qubits for this part?

- Reserving a slot in a link schedule affects the adjacent link schedule availability.
- Determine the order of the links and which slots should be used.
 - Prune to isolate valid pairs of adjacent slots that allow SWAP-ing to achieve reasonable fidelity.
 - * "Window" adjacent schedules to find pairs of slots. Window size should tolerate occupied slots between available slots. May be based on achievable fidelity and how low we can accept the fidelity to become before the second link generates.
 - * Can we order these slot pairs by a greedy heuristic?
 - * Can also order based on how much faster/slower a link would degrade the sooner we can push a SWAP through to a device that decoheres less.
 - There may be several "starting" points within a link availability schedule, how to pick which one?
 - * Consider the timespan of generating all the links?
 - * Construct a naive order assuming full availability and then slot?
 - * Construct the order based on the available slots? Start with the earliest slot? Start with the worst link?
 - * Brute force is $O(l2^n)$ if an end node's schedule has l starts and we use the closest two slots (before/after) for n nodes
 - * Greedy has $O(nl)$ if an end node has l different starts and we always select the next one simply
- How to handle requests for several pairs?
 - Instruct nodes to have a decided link scheduling recurring to meet the max time requirement. "Do schedule X every 10 slots".
 - Using multiple routes? This can incur a significant communication overhead if a request is cancelled.
- Token Bucket (?) - Investigate

4 Questions

- Could decoherence at intermediate links be more important because it affects the success of SWAPs?
- Wait some time before requested to generate entanglement to gather more information about other desired links?