RuleSet for Distributed Quantum Error Correction

> [time=Wed, Jan 19, 2022 22:23 PM]
>
> [name=Bernard Ousmane Sane (BOS)]

tags: 'quantum-error-correction` Encoding` Decoding` 'Stablizers'

Research Objectives:

In this work, we propose a solution-based quantum error correction for reducing nodes' memories requirement. Hence, we define a distributed erasure code that can do fast recovery while using a few nodes, unlike existing proposals. This allows:

- To reduce the computational complexity
- To reduce the number of nodes to contact during the recovery process

Construction of $[n, k, r]$ LRC code

Process 1 : the parameters

1. Prepare $n, q, r, and\ k$ such that $r + 1/n$ and r/k.
2. Prepare a polynomial $g(x) \in \mathcal{F}_q$ such as:
   - deg(g) is $r + 1$
   - $\mathcal{A} = \cup_{i=1}^{\frac{n}{r+1}} \mathcal{A}_i$ is a partition of $\mathcal{A} \subseteq \mathcal{F}_q$ with $n = |\mathcal{A}|$, $r + 1 = |\mathcal{A}_i|$
   - g is constant on each set $\mathcal{A}_i$ in the partition.
3. Send the result to each node

Process2: $[n, k, r]$ **LRC code** codeword

1. Wait for a message $m = m_1 ... m_k$ from the system
2. Wait for the code's parameters
3. form the codeword $c = c_1 ... c_n$ such as c_i is a function of a small number (at most r) of other symbols

- Distribute c_i over the nodes

Process3: recover an erasure symbol c_t^d of the codeword $c^d = c_1^d ... c_n^d$

1. Determine the nodes which contain the symbols c_i^d, $i \neq t$
2. Wait for symbols $c_1, ..., c_r$ from r Nodes
3. recover c_i by using at most r symbols

Process3: Blocks of qubits

1. Wait for N qubits from the system
2. Split the N qubits into m blocks of k qubits
3. Build a km matrix where the j^{th} column describes the j^{th} block.

Process4: Quantum local Recovery ceode $[[n, k, r]]$

1. Wait for $\mathcal{C} = [n, k, r]$ LRC code
2. Generates the parity-check and generator matrices
3. Build the quantum code using the theorem given by *A. R. Calderbank et al.*
4. Send the result to every node

Process5: Encoding a block $\mathcal{B}^i$ from a node $\mathcal{N}^j$

1. Wait for the block $\mathcal{B}^i = \{q_1^i, ..., q_k^i\}$
2. Produce a logical state $|\psi^i\rangle$ which is a combination of states $|\psi_j^i\rangle$ where $1 < j \leq n$.
3. Removes $\mathcal{B}^i$ and distribute $|\psi_j^i\rangle$ where $1 < j \leq n$ over the $\mathcal{N}$ nodes

Process 6: Quantum Local recovery code

1. Check the availability of the nodes
2. Determine the state $|\psi^i\rangle$ which contains the encoding value of $\mathcal{B}^i$
3. Launch a broadcast request for $\{|\psi_1^i\rangle, , ..., |\psi_n^i\rangle\}$ from nodes $\{\mathcal{N}^1, ..., \mathcal{N}^n\}$ (**remote gate**)
4. Recover the block $\mathcal{B}^i$ using $\{|\psi_1^i\rangle, ..., |\psi_n^i\rangle\}$