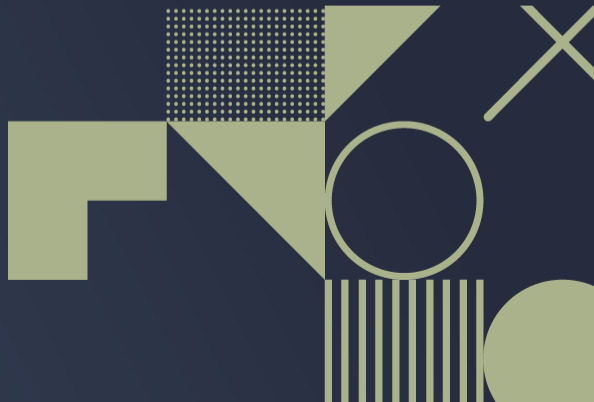
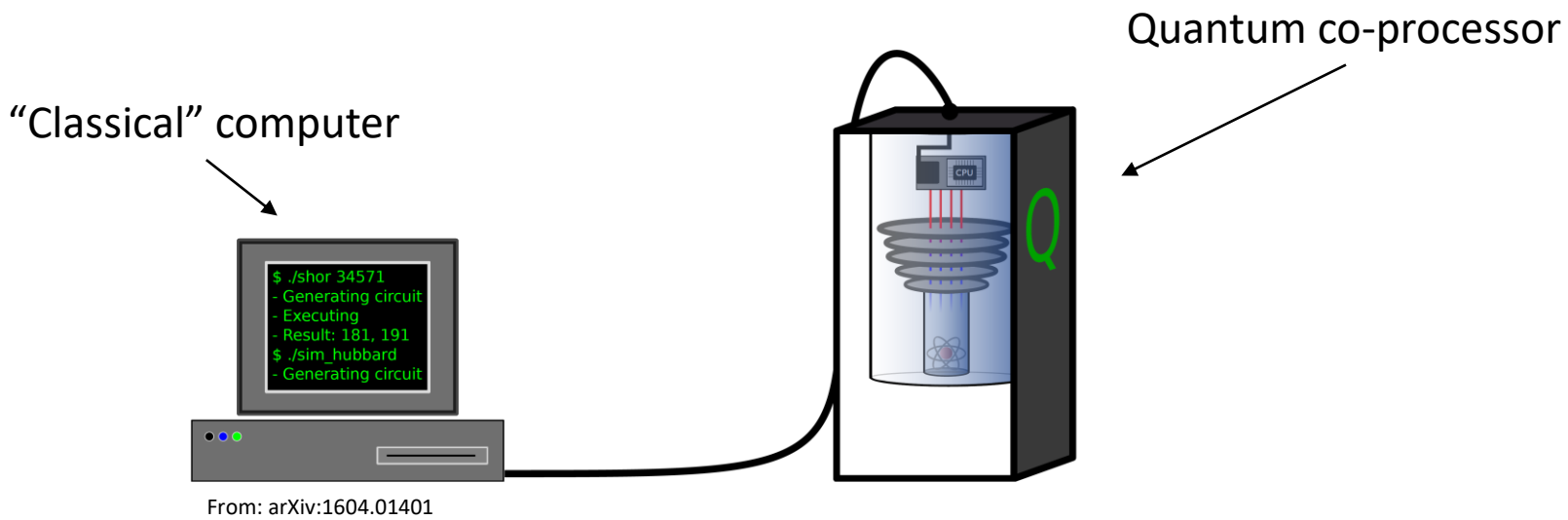
Distributed Quantum Computing with QMPI

T. Häner, D. Steiger, T. Hoefler, M. Troyer



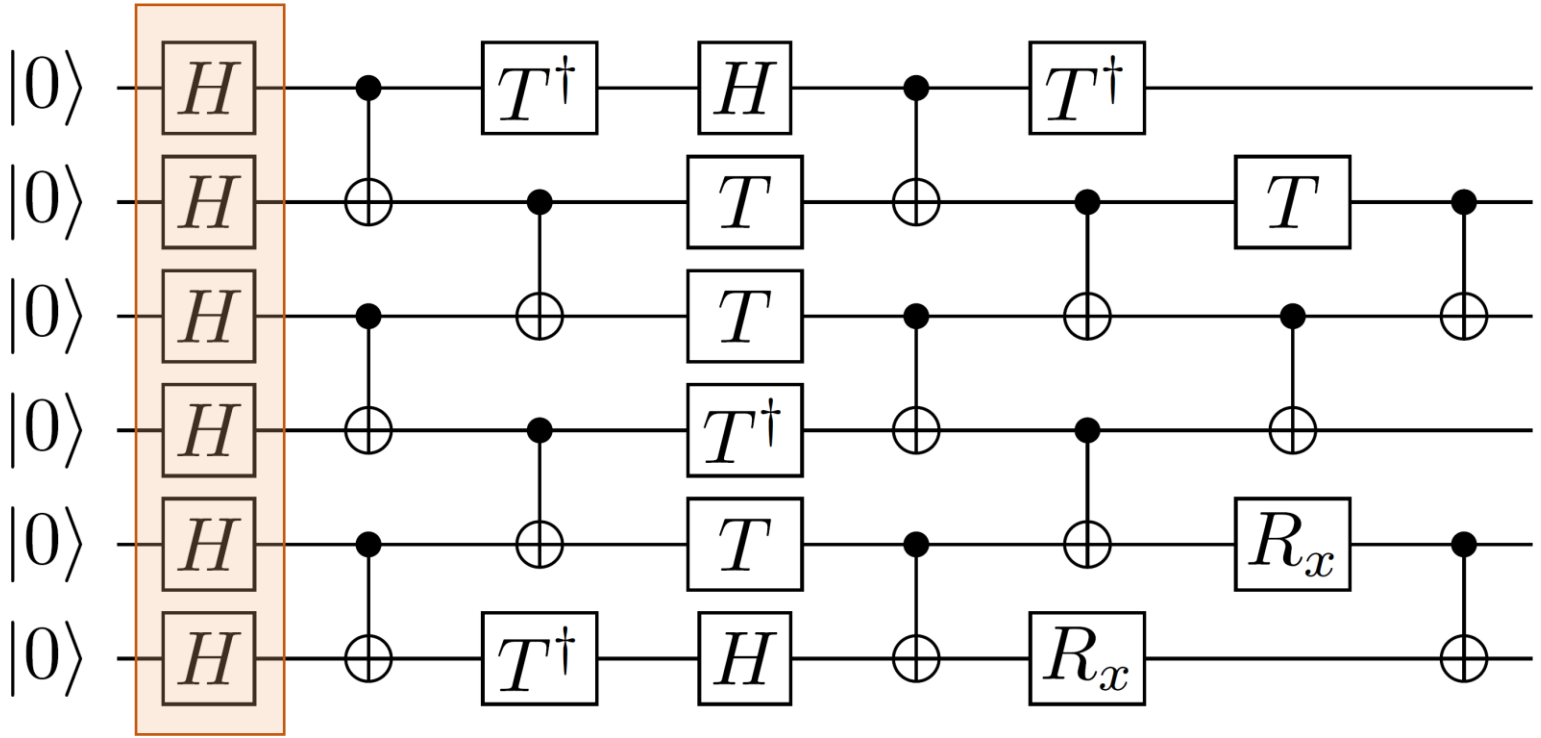
Quantum computing

> Using quantum phenomena to solve certain problems faster



From: arXiv:1604.01401

Quantum bits, gates and circuits

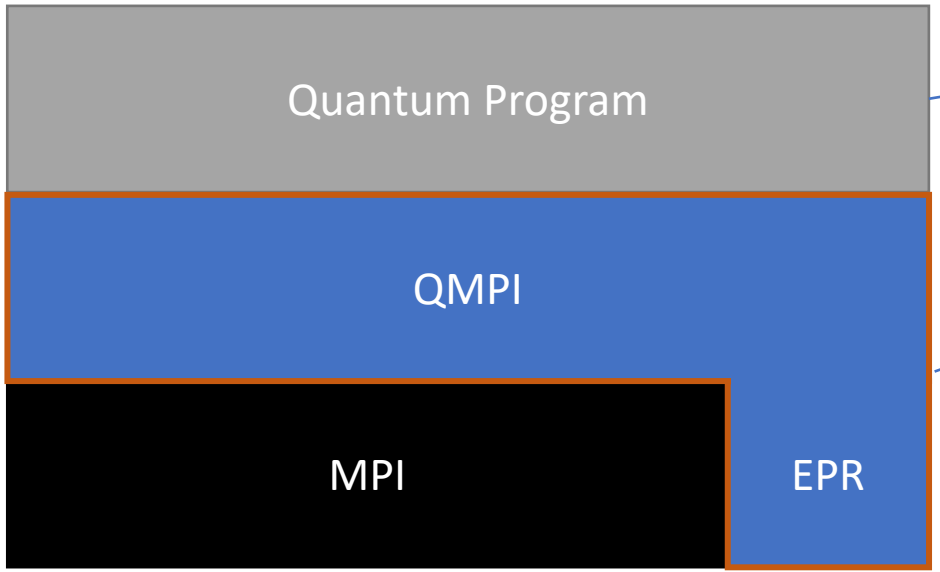


Why *distributed* quantum computing?

Applications require millions of qubits

> Challenging to produce on a single chip!

Quantum MPI (QMPI)



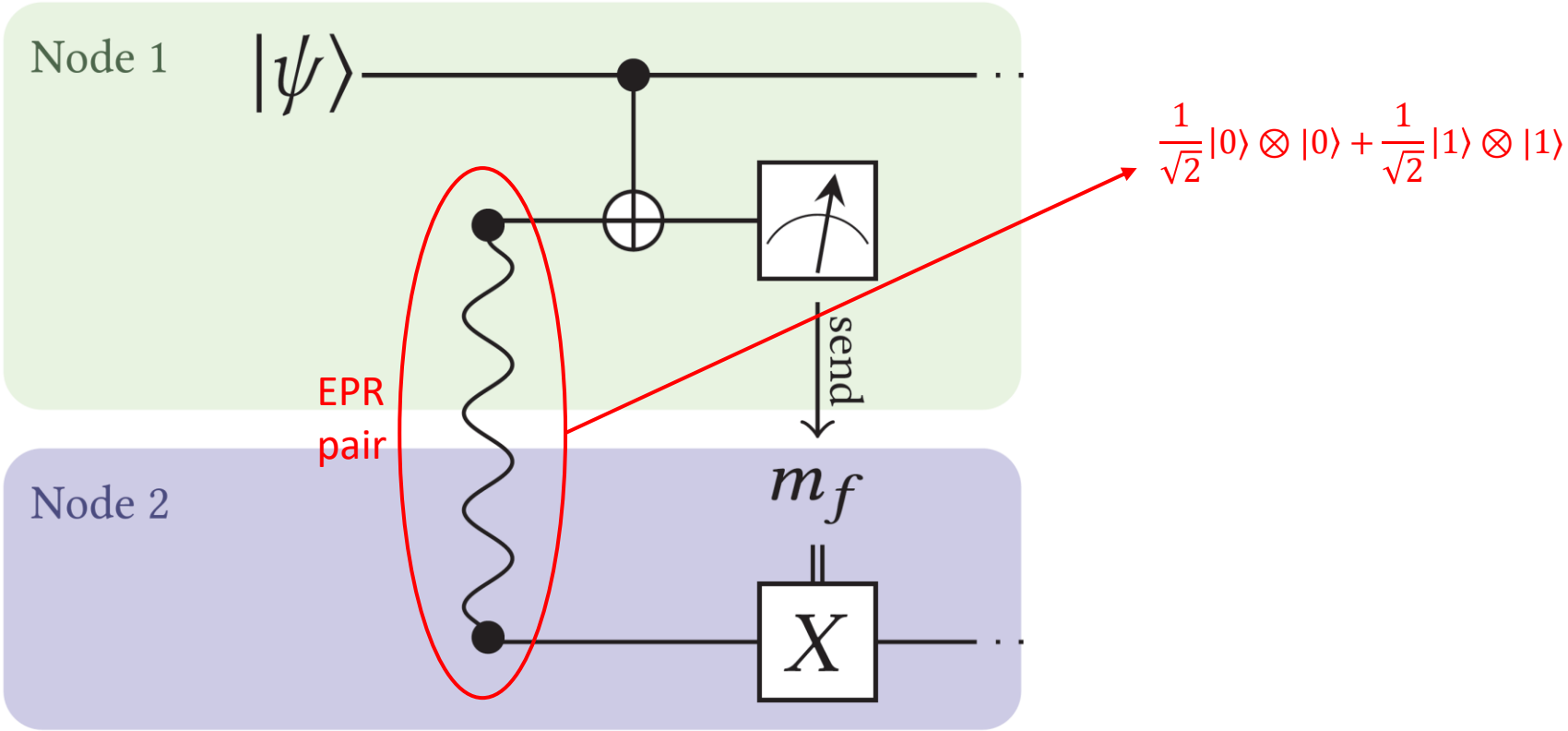

```
…

H(qubit); // send |+> state
QMPI_Send(qubit, other, tag, QMPI_COMM_WORLD);

…
```

qmpi_send.cpp:

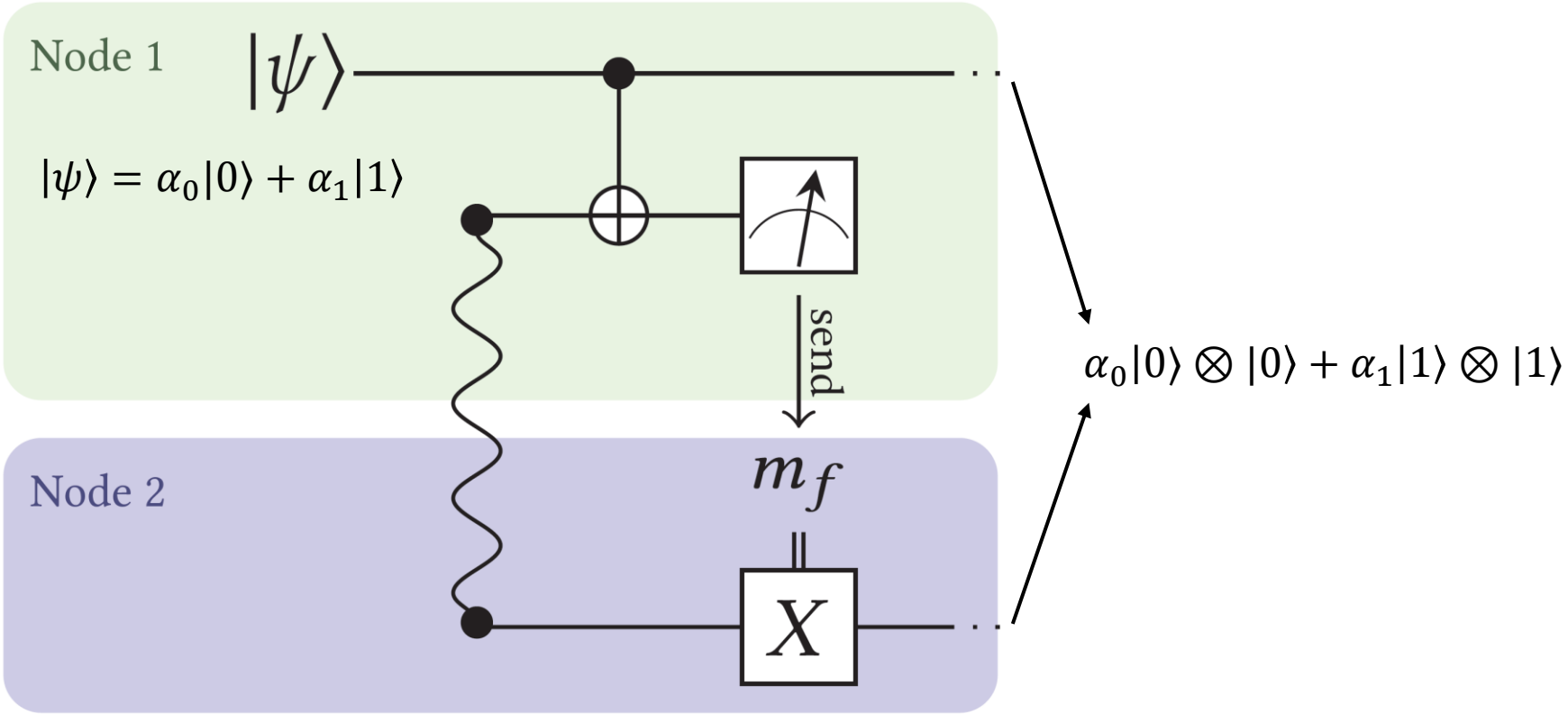
```
…

QMPI_Prepare_EPR(qubit, other, tag, comm);

…
```

How to send/receive qubits?

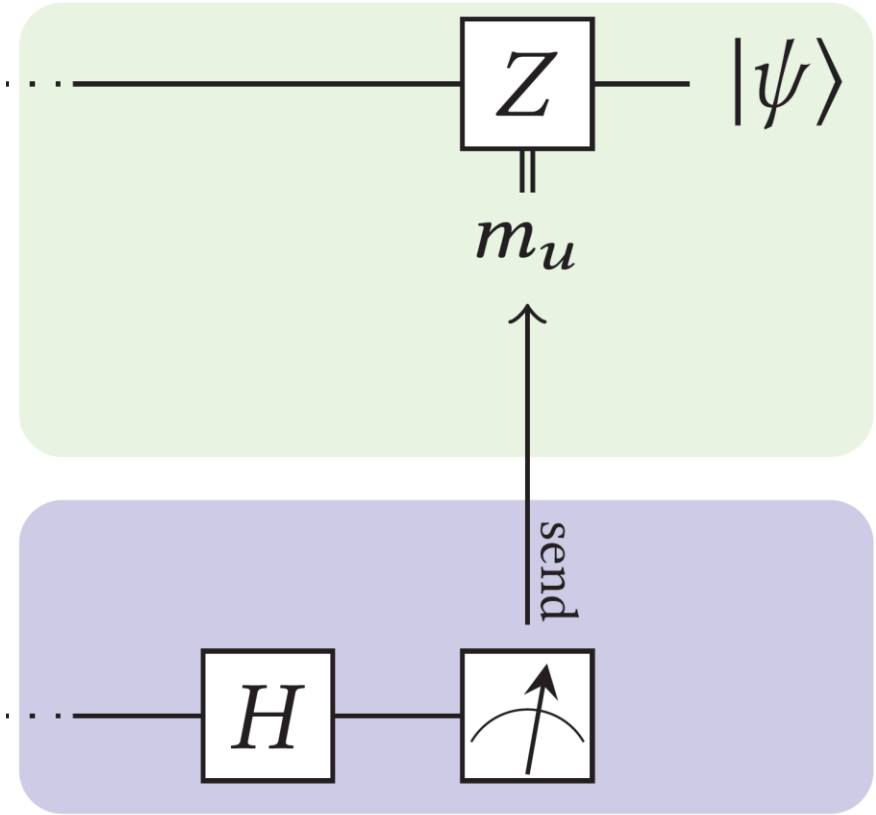
Step #1: Entangled copy



Step #1: Entangled copy

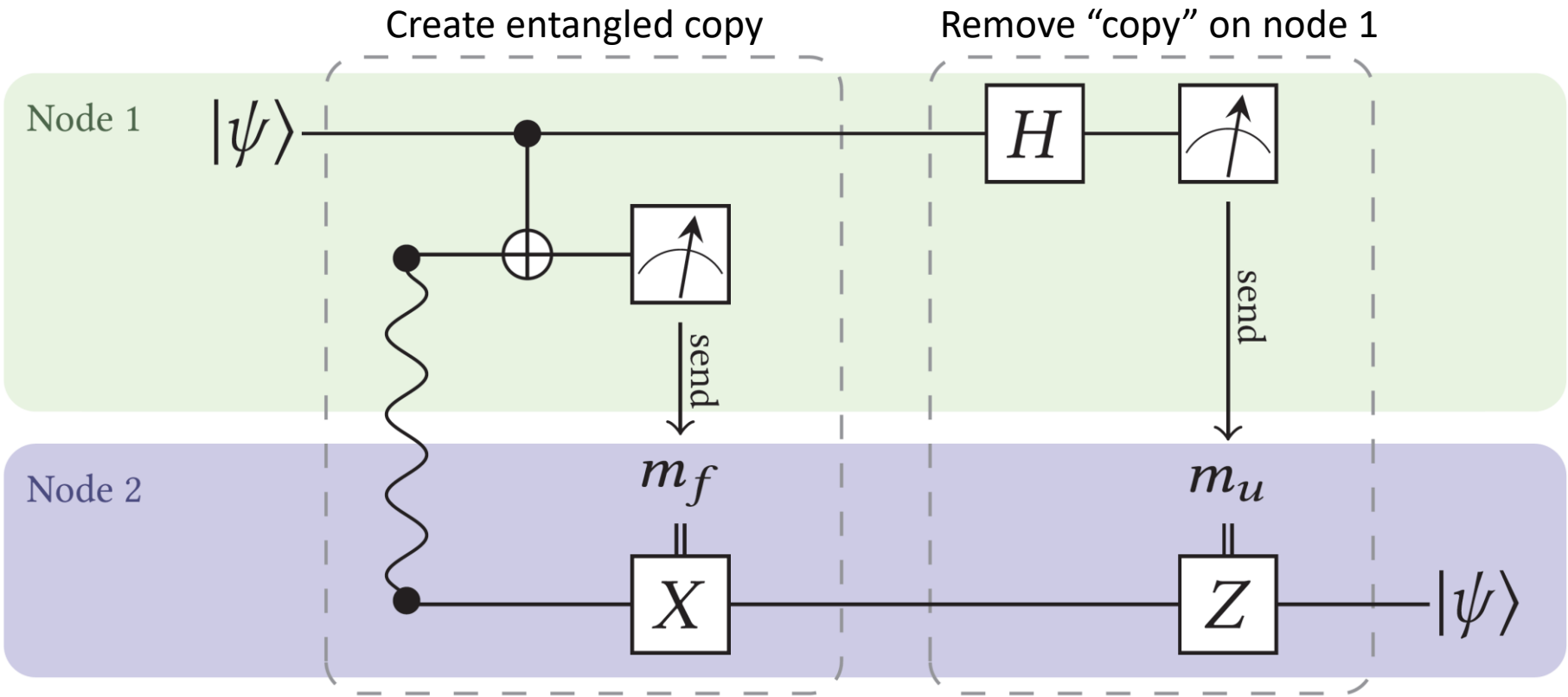


Step #2: Remove copy



Note: Removing a copy is cheaper than establishing it!
→ QMPI provides send & unsend

Move-semantics: Teleportation



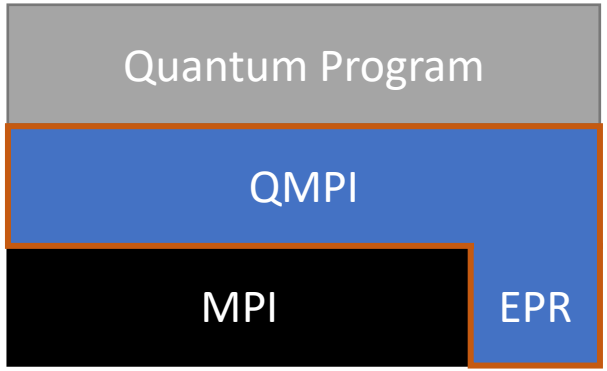
QMPI basics

> QMPI inherits classical communication from MPI

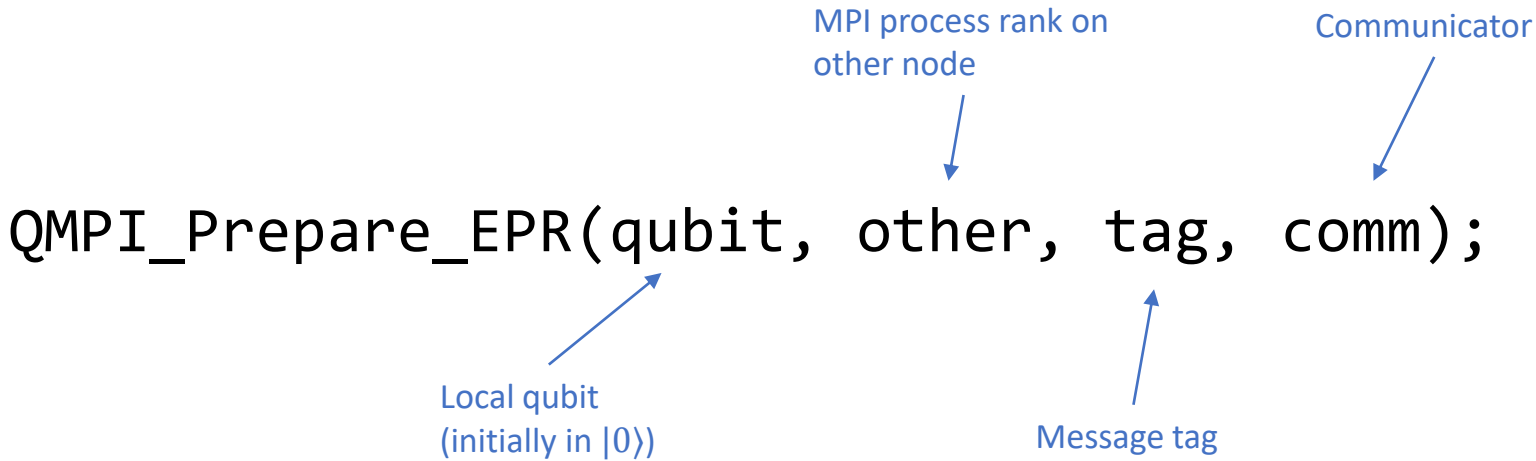
> Different QMPI “modes”: *Copy* vs *Move* semantics

> Reductions (*copy* semantics)

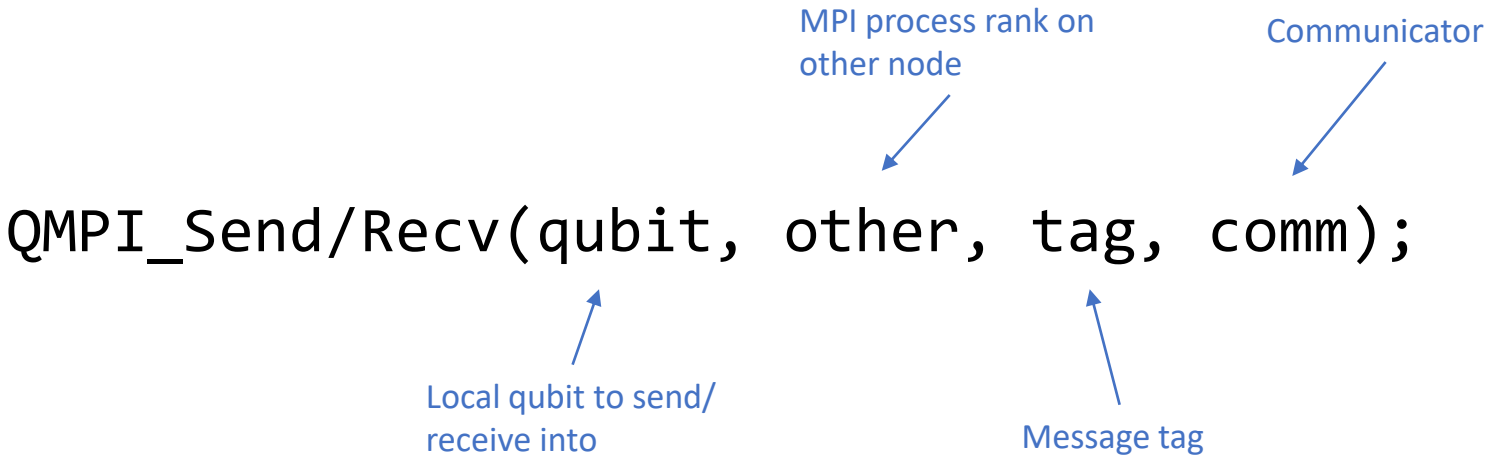
- operations must be reversible
- QMPI_Reduce *and* QMPI_Unreduce



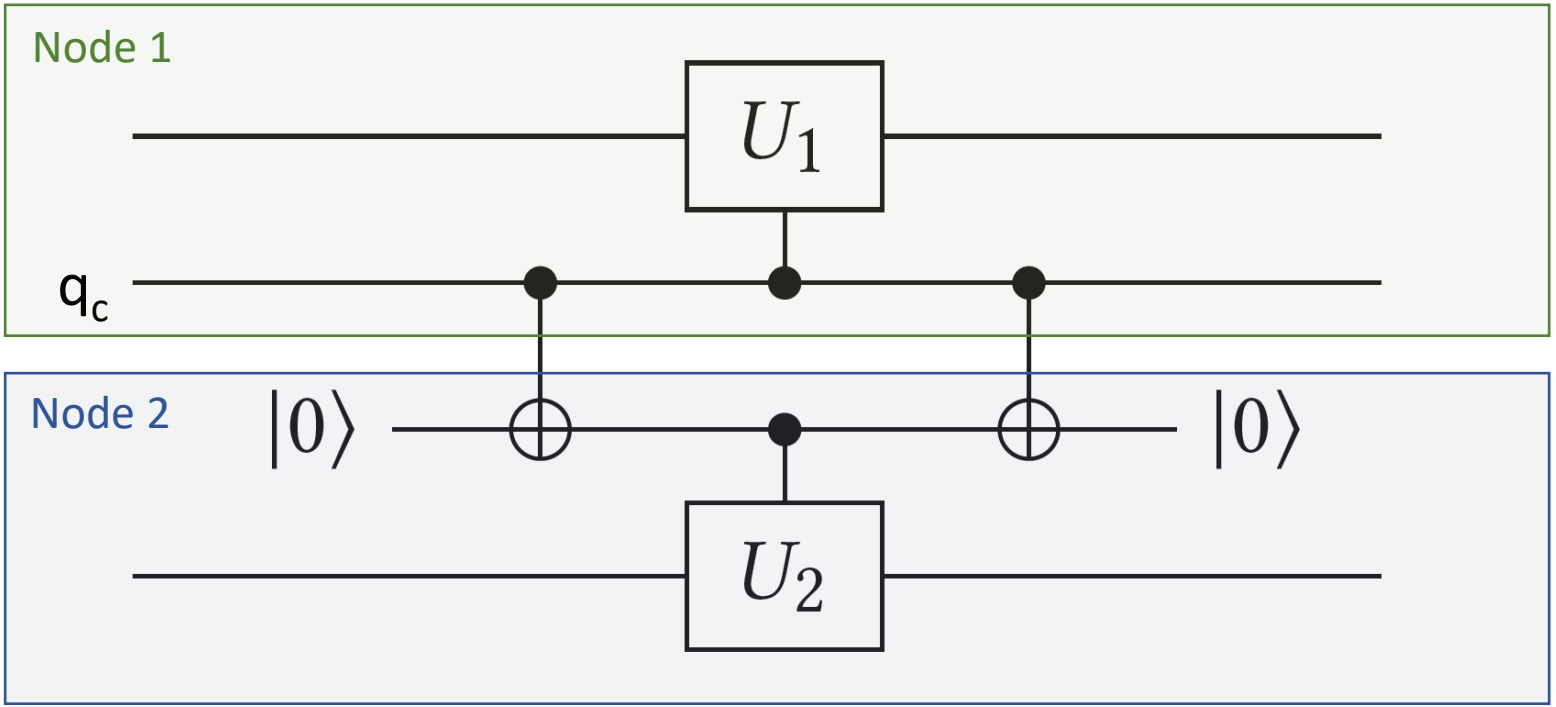
QMPI basics: EPR pairs



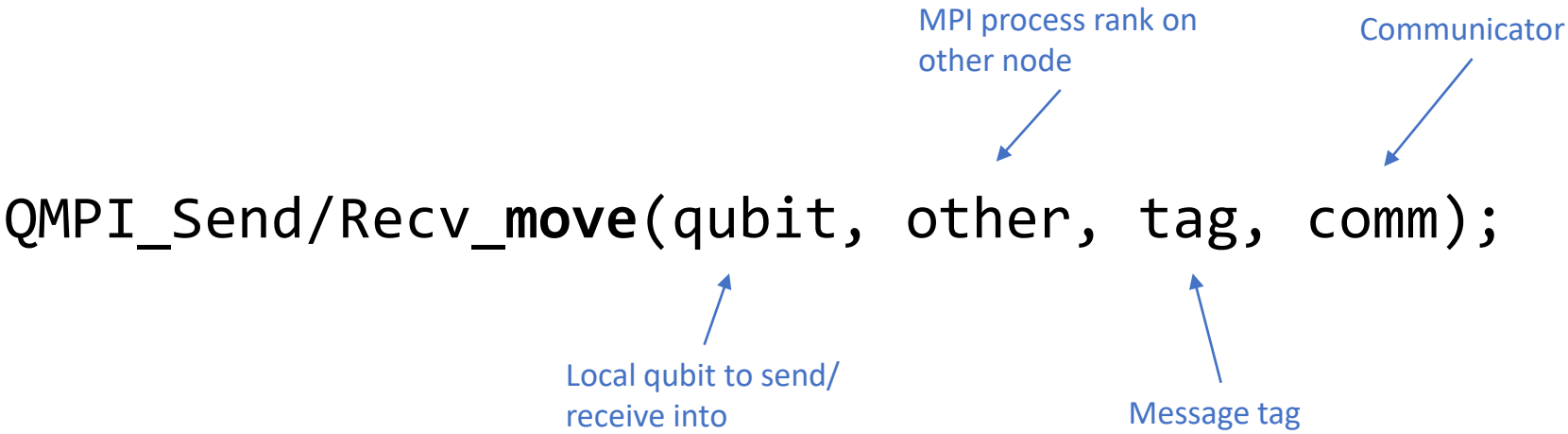
QMPI basics: Copying send/recv



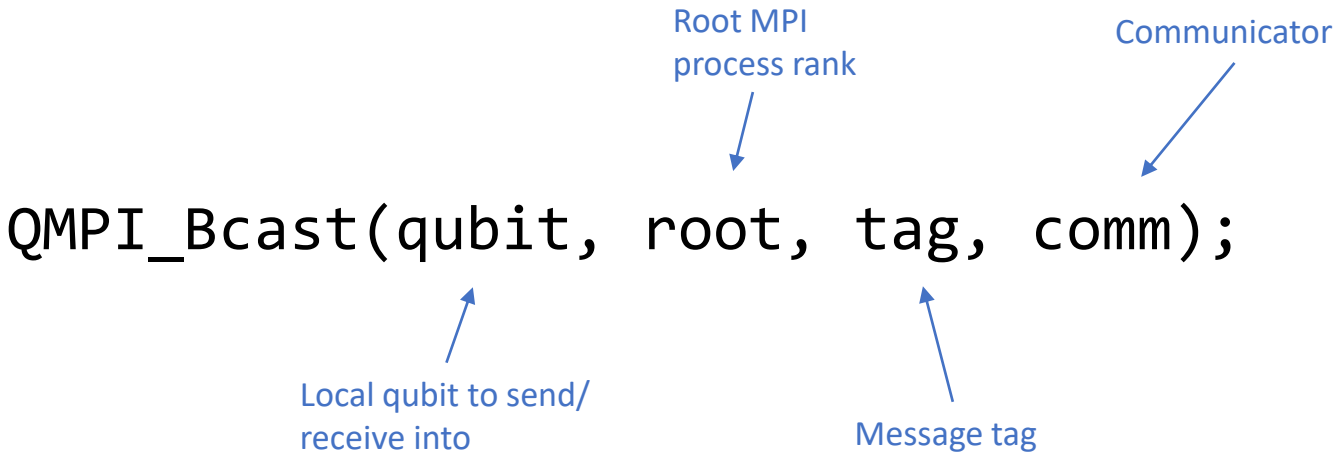
QMPI basics: Copying send/recv



QMPI basics: Moving send/recv



QMPI basics: Bcast



Operation	Reverse operation
QMPI_Send, QMPI_Bsend, QMPI_Ssend, QMPI_Rsend	QMPI_Unsend, QMPI_Bunsend QMPI_Sunsend, QMPI_Runsend
QMPI_Recv, QMPI_Mrecv	QMPI_Unrecv, QMPI_Munrecv
QMPI_Sendrecv	QMPI_Unsendrecv
QMPI_Sendrecv_replace	QMPI_Unsendrecv_replace
QMPI_Cancel	—
QMPI_Send_move, QMPI_Bsend_move, QMPI_Ssend_move, QMPI_Rsend_move	QMPI_Unsend_move, QMPI_Bunsend_move, QMPI_Sunsend_move, QMPI_Runsend_move
QMPI_Recv_move, QMPI_Mrecv_move	QMPI_Unrecv_move, QMPI_Munrecv_move

Operation	Reverse operation
QMPI_Bcast	QMPI_Unbcast
QMPI_Gather, QMPI_Gatherv	QMPI_Ungather, QMPI_Ungatherv
QMPI_Scatter, QMPI_Scatterv	QMPI_Unscatter, QMPI_Unscatterv
QMPI_Allgather, QMPI_Allgatherv	QMPI_Unallgather, QMPI_Unallgatherv
QMPI_Alltoall, QMPI_Alltoallv, QMPI_Alltoallw	QMPI_Unalltoall, QMPI_Unalltoallv, QMPI_Unalltoallw
QMPI_Reduce	QMPI_Unreduce
QMPI_Allreduce	QMPI_Unallreduce
QMPI_Reduce_scatter, QMPI_Reduce_scatter_block	QMPI_Unreduce_scatter, QMPI_Unreduce_scatter_block
QMPI_Scan, QMPI_Exscan	QMPI_Unscan, QMPI_Unexscan
QMPI_Gather_move, QMPI_Gatherv_move	QMPI_Ungather_move, QMPI_Ungatherv_move
QMPI_Scatter_move, QMPI_Scatterv_move	QMPI_Unscatter_move, QMPI_Unscatterv_move
QMPI_Alltoall_move, QMPI_Alltoallv_move, QMPI_Alltoallw_move	QMPI_Unalltoall_move, QMPI_Unalltoallv_move, QMPI_Unalltoallw_move

How to optimize collectives?

The SENDQ performance model

Should be simple, but capture important metrics

S: Size of EPR pair buffer (per node)

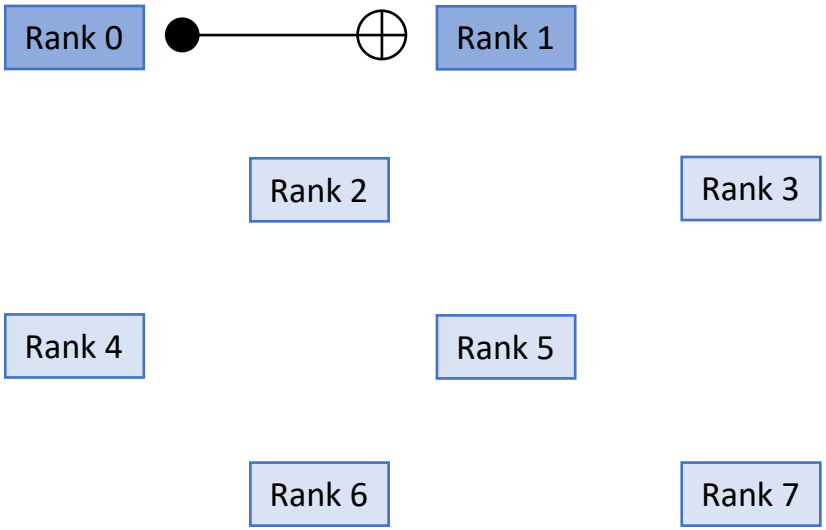
E: EPR pair generation time (with any other node, one at a time)

N: Number of nodes

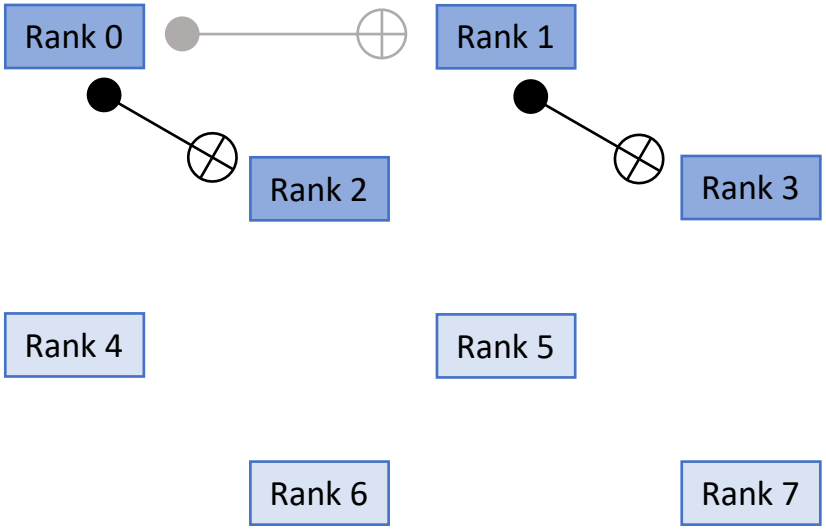
D: Delay due to local computation

Q: Qubits available for local computation (per node)

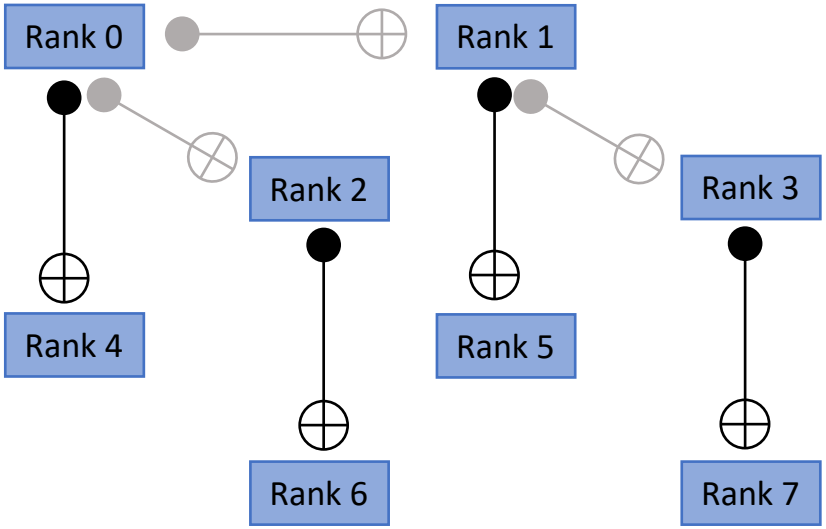
Example 1: QMPI_Bcast (S=1)



Example 1: QMPI_Bcast (S=1)

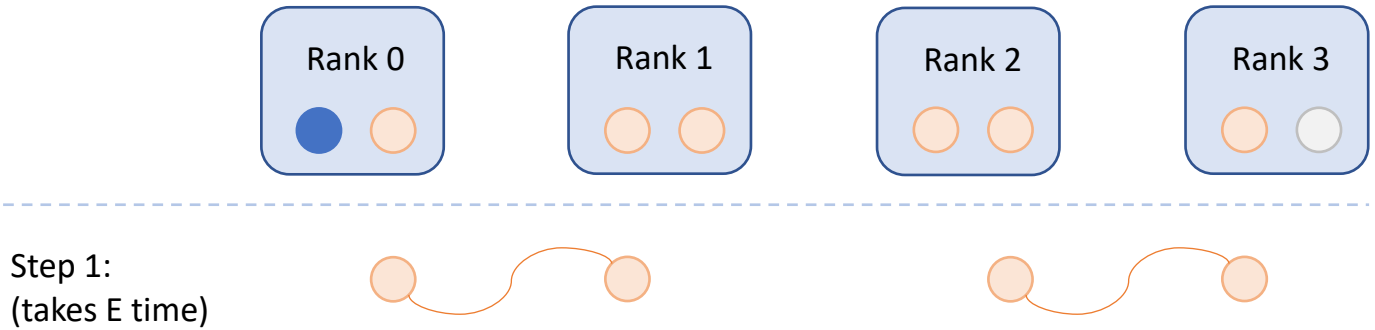


Example 1: QMPI_Bcast (S=1)

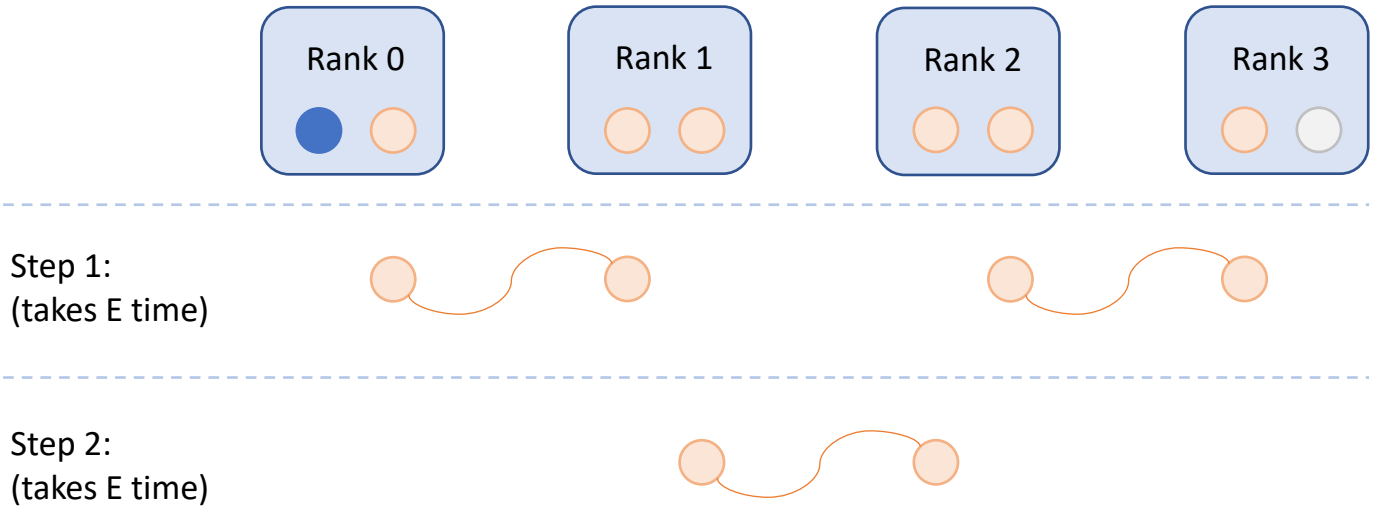


Time for Bcast on N nodes: $\sim E[\log_2(N)]$

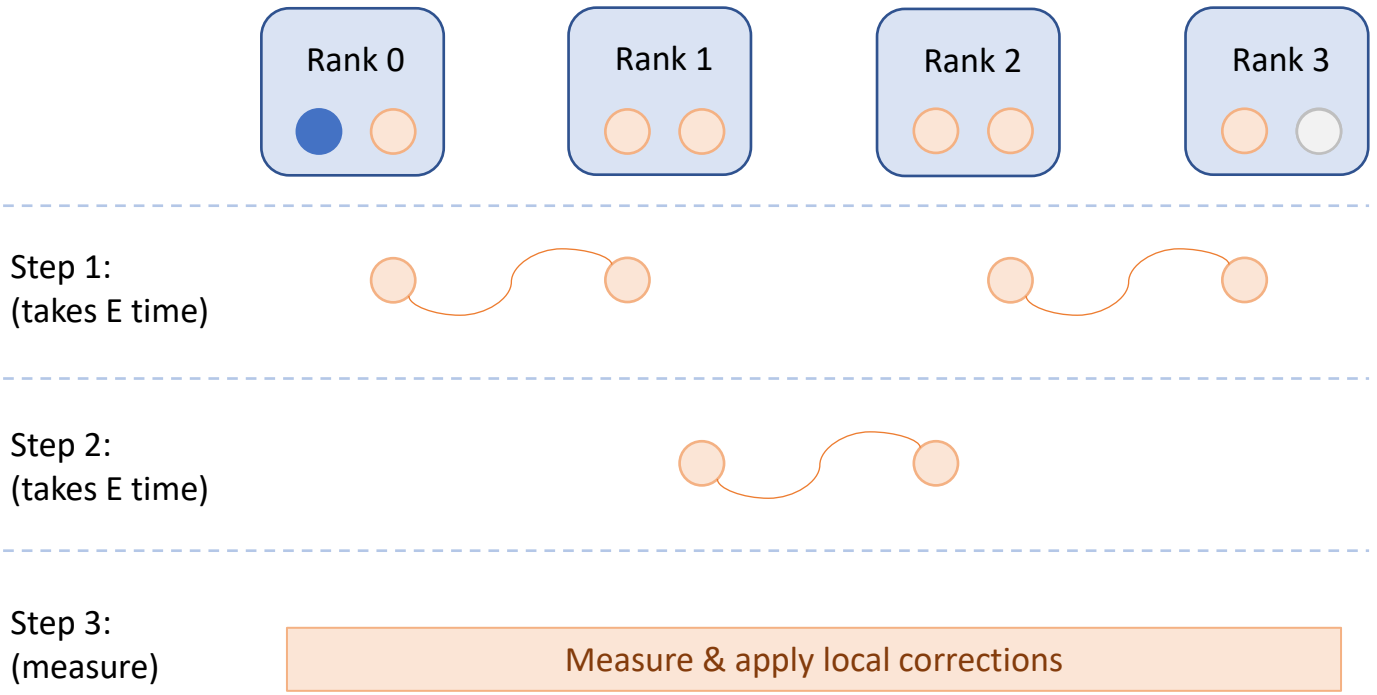
Example 1: QMPI_Bcast (S=2)



Example 1: QMPI_Bcast (S=2)



Example 1: QMPI_Bcast (S=2)



Example 2: Chemistry

> Chemistry Hamiltonian

$$H = \sum_{ij,\sigma} h_{ij} a^{\dagger}_{(i,\sigma)} a_{(j,\sigma)} + \frac{1}{2} \sum_{ijkl,\sigma\rho} h_{ijkl} a^{\dagger}_{(i,\sigma)} a^{\dagger}_{(k,\rho)} a_{(l,\rho)} a_{(j,\sigma)}$$

> $a_i^{\dagger}, a_i$ are **fermionic** creation/annihilation operators

Example 2: Chemistry

$$P = X \otimes Z \otimes \cdots \otimes Y$$

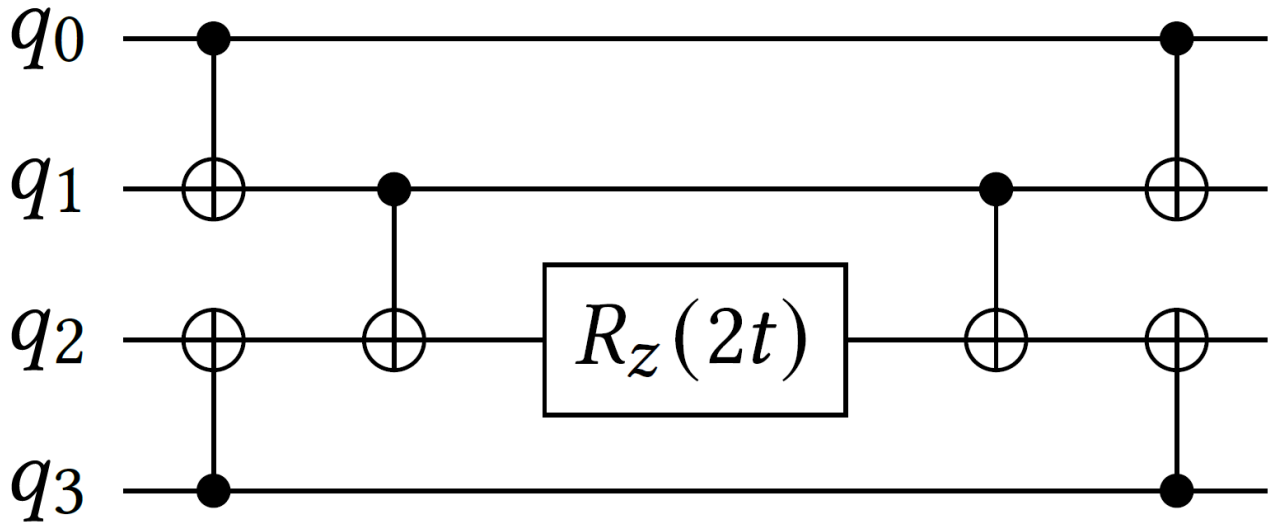
$$e^{i\theta P}$$

Example 2: Chemistry

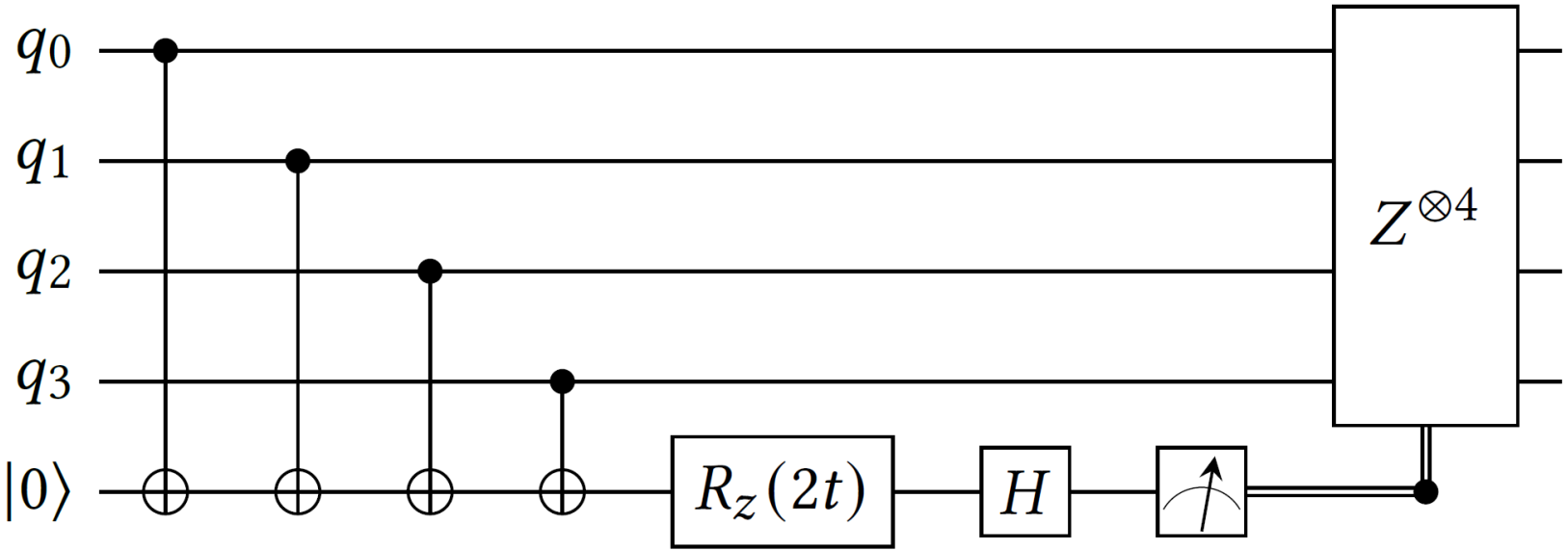
$$P' = Z \otimes Z \otimes \cdots \otimes Z$$

$$e^{i\theta P} = U^{\dagger} e^{i\theta P'} U$$

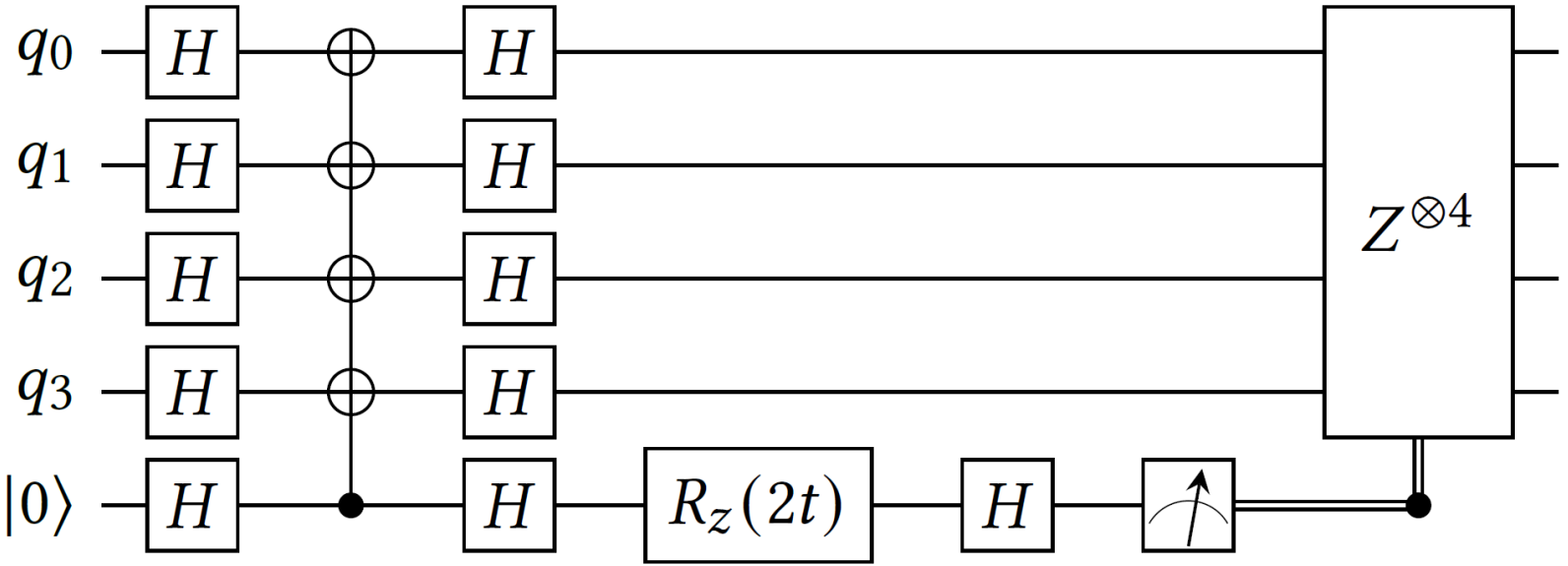
Example 2: Chemistry



Example 2: Chemistry



Example 2: Chemistry



Distributed Quantum Computing with QMPI

T. Häner, D. Steiger, T. Hoefler, M. Troyer