

RESEARCH ARTICLE

Simulator Demonstration of Large Scale Variational Quantum Algorithm on HPC Cluster

MIKIO MORITA¹, YOSHINORI TOMITA, JUNPEI KOYAMA¹, AND KOICHI KIMURA

Quantum Laboratory, Fujitsu Ltd., Kawasaki 211-8588, Japan

Corresponding author: Mikio Morita (morita.mikio@fujitsu.com)

ABSTRACT Advances in quantum simulator technology are increasingly required because research on quantum algorithms is becoming more sophisticated and complex. State vector simulation utilizes CPU and memory resources in computing nodes exponentially with respect to the number of qubits; furthermore, in a variational quantum algorithm, the large number of repeated runs by classical optimization is also a heavy load. This problem has been addressed by preparing numerous computing nodes or simulation frameworks that operate effectively. This study aimed to accelerate quantum simulation using two newly proposed methods: to efficiently utilize limited computational resources by adjusting the ratio of the MPI and distributed processing parallelism corresponding to the target problem settings and to slim down the Hamiltonian by considering the effect of accuracy on the calculation result. Ground-state energy calculations of a fermionic model were performed using a variational quantum eigensolver (VQE) on an HPC cluster with up to 1024 FUJITSU Processor A64FX connected to each other by InfiniBand; the processor is also used on supercomputer Fugaku. We achieved 200 times higher speed over VQE simulations and demonstrated 32 qubits ground-state energy calculations in acceptable time. This result indicates that > 30 qubit state vector simulations can be realistically utilized for further research on variational quantum algorithms.

INDEX TERMS Distribute computing, quantum computing, quantum simulation, variational quantum algorithm, variational quantum eigensolver.

I. INTRODUCTION

Quantum computer has received much attention from both hardware and algorithmic perspectives in the last decades. This is because it was expected to perform faster than classical computation in certain calculation area [1]. Previous studies have reported several experiments of quantum computation conducted on a noisy intermediate scale quantum computer (NISQ [2]) [3], [4]. However, the limited capability of NISQ, derived from hardware noise and short coherence time, makes it challenging to validate sophisticated and complex quantum algorithms that contain many qubits or gate operations. Therefore, quantum simulations on classical computers, which simulate ideal noise less or manually

controllable noise environment, are increasingly important for the advancement of quantum algorithm study.

Quantum simulation requires large computational resources of classical computers. A state vector simulator, for example, Qulacs [5], must allocate 2^{n+4} bytes to store 2^n double-precision complex numbers for representing n qubit quantum state. Consequently, memory resource requirement increases exponentially. Accordingly, the gate operation must control exponentially increasing computational bases, hence computational costs also increase exponentially. In addition to the state vector, other quantum simulation methods have been proposed such as tensor networks [6] and decision diagrams [7], which also require large computational resources.

The variational quantum algorithm is one of practical quantum algorithms for NISQ. For each purpose, they are defined in the form of variational quantum eigensolver

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya¹.

(VQE) [8], [9], [10], [11], quantum approximate optimization algorithm (QAOA) [12], and quantum machine learning (QML) [13], [14]. Since they perform variational calculations by optimizing parameters embedded in the quantum circuit, the quantum circuit must be executed many times in the algorithm. VQE is an algorithm for quantum chemical calculations that has been studied by simulations and a few qubit real experiments [3], [4]. The VQE can calculate, in typical usage, ground-state energy or excitation energy [15], [16], [17] by representing molecular orbitals in the quantum circuit.

Several simulations were performed by previous studies. As a partial example of VQE, the ground-state calculations of BeH₂ of 14 qubits [18], N₂ of 16 qubits [19], and naphthalene of 20 qubits [20] have been reported. These problem targets are all fermionic model Hamiltonian. On the other hand, Heisenberg model Hamiltonian, a limited model that interacts with adjacent spins, has been reported to be up to 40 qubits [21]. Similarly, non-variational quantum computational simulations on HPC cluster have been reported. Quantum simulations up to 72 qubits supported by NASA HPC cluster *Pleiades* and *Electra* [22] and 121 qubits on the supercomputer *summit* [23] have been carried out. However, research on the large-scale implementation of variational quantum algorithms is still limited; thus discussion on the feasibility of this area is important. Especially, there is a lack of VQE for dealing with fermionic models, which require a large number of Hamiltonian terms.

This paper reports the demonstration of fermionic model VQE simulations up to 36 qubits using two newly proposed techniques. The first technique is to efficiently combine MPI parallel and distributed processing corresponding to the target problem. By optimizing the node usage ratio of two parallelization methods for the prepared computing nodes, the computing capability can be improved. The second technique is to slim down Hamiltonian to speed-up the expectation value calculation of quantum simulation; this accelerates VQE while maintaining nearly the same calculation accuracy. The simulation was carried out on an HPC cluster with up to 1024 FUJITSU Processor A64FX (A64FX) connected to each other by InfiniBand; the processor is also used on supercomputer *Fugaku*.

For the 28- and 32-qubit problems, we demonstrated a complete simulation until the obtained energy was converged. For the 36-qubit problem, only one iteration was performed; the characteristic was revealed.

II. METHODOLOGY

In this section, we introduce our VQE simulation methods for an HPC cluster system. The section is divided into the following:

- A. Computing node
- B. VQE algorithm
- C. MPI and distributed processing
- D. Hamiltonian terms cutoff

TABLE 1. HPC cluster system specifications.

Cluster node	FUJITSU PRIMEHPC FX700
# of nodes	1024
Theoretical peak FLOPS	3146 TFLOPS (double precision)
CPU	FUJITSU Processor A64FX
Instruction set architecture	Armv8.2-A + SVE
# of CPUs per node	1
# of cores per node	48 (computing core) 4 (assistant core)
# of threads per node	48
Base frequency	2.0 GHz
Boost frequency	2.0 GHz (no boost mode)
Theoretical peak FLOPS per CPU	3.1 TFLOPS (double precision)
Memory capacity per node	32 GiB
Memory band-width	1.024 GB/s
Interconnect	InfiniBand EDR

An A64FX is installed in an FX700 node. 32-GiB memory is stored in the A64FX. More than 1030 FX700 units are actually connected; however, only 1024 units are expected to be used simultaneously.

Subsections A and B describe known techniques, where A shows the hardware usage of the HPC cluster system and B shows the VQE algorithm. Subsections C and D describe the newly proposed methods; C explains a way to combine MPI and distributed processing parallelization, and D explains how to accelerate expectation value calculation by slimming down Hamiltonian. For another method related to coding, we used pySCF [24] for quantum chemical calculations, openfermion [25] for hamiltonian qubit mapping, and Qiskit SLSQP for parameter optimization. Qiskit SLSQP function is based on scipy [26].

A. COMPUTING NODE

Large-scale quantum simulations have been performed on a number of multi-node HPC cluster systems as in previous studies [22], [23]. Our system consists of FUJITSU PRIMEHPC FX700 (FX700), a computing node with A64FX and 32-GiB memory. A64FX, the Armv8.2-A instruction set architecture, is a processor also used in the *Fugaku* supercomputer. A total of 1024 nodes are connected by InfiniBand EDR. See Table. 1 for the detailed specifications of our HPC cluster system. Our system is also an HPC cluster as in previous studies; however it was designed for a state vector simulator.

Several quantum simulation frameworks such as Qiskit Aer, Intel-QS [27], and Qulacs [5] have been published. We used mpiQulacs [28], an extended Qulacs for distributed simulation, to run on this cluster system. MpiQulacs was selected because Qulacs is one of the fastest state vector simulation frameworks.

MpiQulacs takes an MPI approach similar to intel-QS [27], in which the vector representing the quantum state is divided by the number of MPI processes. The memory used per node is $\frac{2^{n+4}}{p} = 2^{n-\log_2 p+4}$ bytes, when the number of MPI processes is p and the number of qubits is n . This means storing a total of two to the n power of double complex type variables. Since the FX700 node has 32-GiB memory, considering the amount of memory required for operating systems, Python code, and so on, $2^{30+4} = 16$ GiB can

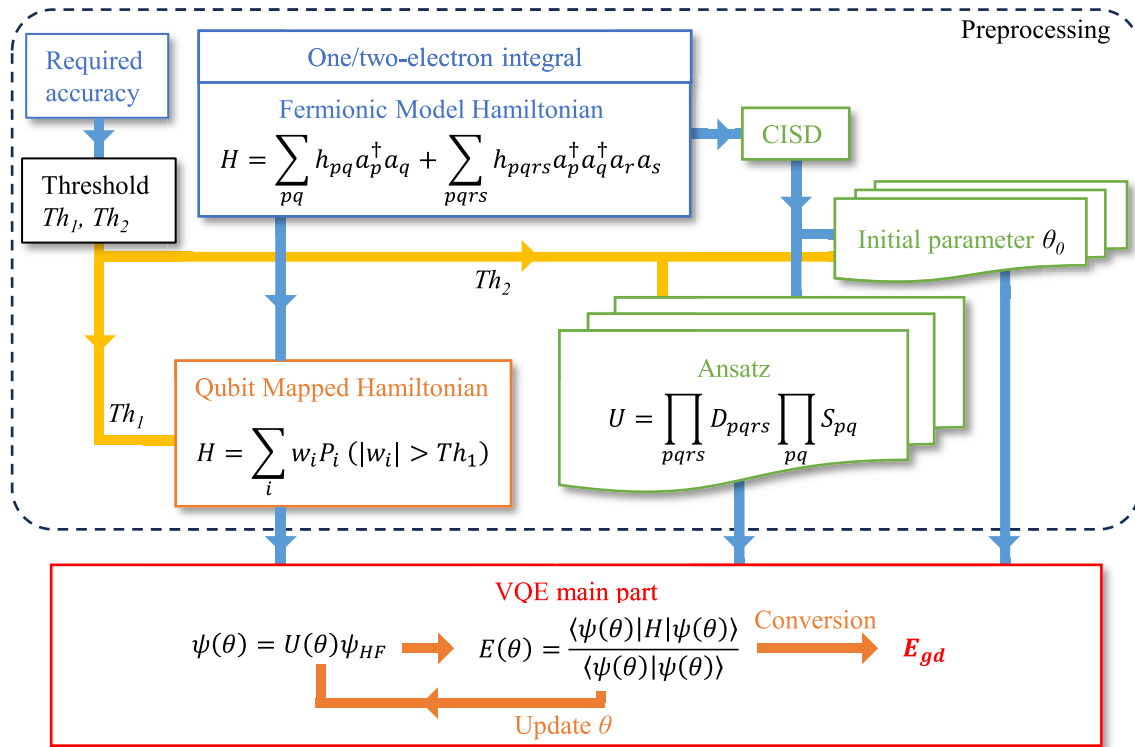


FIGURE 1. VQE overall algorithm flow. one/two-electron integral and CISD are calculated by pySCF. Threshold Th_1 is sent to Hamiltonian output process. Th_2 is sent to initial value and ansatz generation. In the VQE main part, the expected values of all observables are summed up to obtain the energy value $E(\theta)$. This is repeated until it converges to the minimum value E_{gd} .

be used to store quantum states. Therefore, the maximum number of qubits that can be executed by a single node is 30, and the minimum number of $2^{(n-30)}$ nodes are required when calculating > 30 qubits. For example, 36-qubit computation requires at least $2^{36-30} = 64$ nodes connected by MPI communication. Refer to the citation [28] for more information on MPI parallel techniques using mpiQulacs.

B. VQE ALGORITHM

The VQE algorithm has concerns to be determined carefully, particularly for the design of a quantum circuit ansatz and the overall structure of the algorithm including preprocessing.

To achieve better accuracy in shallower circuits, previous studies have proposed many VQE ansatz. Hardware efficient [4], symmetry preserving [29], and UCCSD [30] ansatz are representatives. In addition, their derivatives include gate fabric symmetry preserving [20], UpCCGSD [19], jastrow-factor [31] and qCC [32]. Regarding the overall structure of the VQE algorithm, for example, classical calculation for obtaining initial values of variational parameters [33], adaptive generation of quantum circuits [34], and post-processing to mitigate hardware noise [35], have been proposed as support feature.

The entire algorithm in this study is illustrated in Fig. 1. At the beginning of the preprocessing, values of the one- and two-electron integrals were calculated from the chemical problem of interest. Subsequently, the second quantized Hamiltonian H was generated; it is represented as (1) in the

fermionic model.

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s. \quad (1)$$

Values a^\dagger, a , and h mean generation operator, annihilation operator and coefficient, respectively.

These values were sent to two separate flows. One is to generate qubit-mapped Hamiltonians; because our ansatz has the quantum-number-preserving property, the Jordan-Wigner transform method [36] was chosen instead of Bravyi-Kitaev [37] or parity basis [38]. The Hamiltonian H is represented by the sum of Pauli strings, also known as observables P . In (2), (3), and (4), w represents the observable coefficient, and σ is the Pauli matrix of I, X, Y , or Z . See a citation [39] for the details on possible forms of P .

$$H = \sum w_i P_i. \quad (2)$$

$$P_i = \bigotimes_{k=1}^n \sigma_k^{(i)}. \quad (3)$$

$$\sigma_k^{(i)} \in \{I, X, Y, Z\}. \quad (4)$$

The another is to calculate the configuration interaction singles and doubles (CISD), one of the well-known computational methods of quantum chemistry, to determine the ansatz quantum circuit and initial values of variational parameters. The preprocessing approach to obtain the quantum circuit [32], [34] and the execution of the quantum chemical

calculation to generate the initial parameter value [33] have been studied. We used the modified version of the methods used in these previous studies by utilizing CISD because of its high accuracy and straightforward correspondence with quantum circuits. CI coefficients were extracted from the CISD calculation result. The CI coefficients-generated ansatz U was made up of the product of gate sets, where gate set S represents one-electron excitations on molecular orbitals and D represents two-electron excitations; see (5). Since one CI coefficient corresponds to one excitation operator, it generated one S or D . Among all the one- or two- electron excitations, the CI coefficient determined the excitation operator to be implemented in ansatz. Moreover, the initial parameters of the excitation operator were set, derived from the corresponding CI coefficients. For S and D , we referred to the gate set of the conventional method [20]. This ansatz form is like a cross between gate fabric symmetry preserving [20] and UCCSD [30]. The gate fabric symmetry preserving ansatz is a more advanced form of symmetry preserving, and it has good convergence. In the ansatz in this work, by using the initial value, it converges even better. Therefore, the simulation time should be shorter than that for typical ansatz.

After completing the preprocessing, the process moves to the VQE main part. The ansatz with embedded initial parameters $U(\theta_0)$ acted through various gate operations on the wave function ψ_{HF} representing the Hartree-Fock state. This created a wave function $\psi(\theta_0)$ which represents the superposition of molecular orbitals. The parameters were updated by a classical optimization process. This process is repeated until the obtained energy converges to the ground-state energy E_{gd} , as shown in (6),(7). Types of optimizers include BFGS, Powell, and COBYLA; SLSQP was selected from the viewpoint of the small number of circuit runs until convergence in a noise-less environment.

$$U(\theta) = \prod_{pqrs} D_{pqrs}(\theta_{pqrs}) \prod_{pq} S_{pq}(\theta_{pq}). \quad (5)$$

$$\psi(\theta) = U(\theta)\psi_{HF}. \quad (6)$$

$$E_{gd} = \min_{\theta} \frac{\langle \psi(\theta) | H | \psi(\theta) \rangle}{\langle \psi(\theta) | \psi(\theta) \rangle}. \quad (7)$$

The target molecules for ground-state energy calculation were CO_2 for 28 qubits and C_3H_6 for 32 and 36 qubits, respectively. The basis functions are STO-3G for all.

C. MPI AND DISTRIBUTED PROCESSING

In general, a method for computing plurality of nodes is not only MPI communication but also distributed processing using remote procedure call (RPC). RPC is the technology in the connected remote node; the technology can be utilized to speed-up the entire computation by concurrent execution of processes that need not be executed in series. Among the widely used RPC frameworks, we used gRPC [40] owing to its simplicity and high performance. As an operational form of gRPC, a gRPC client for overall management was assigned

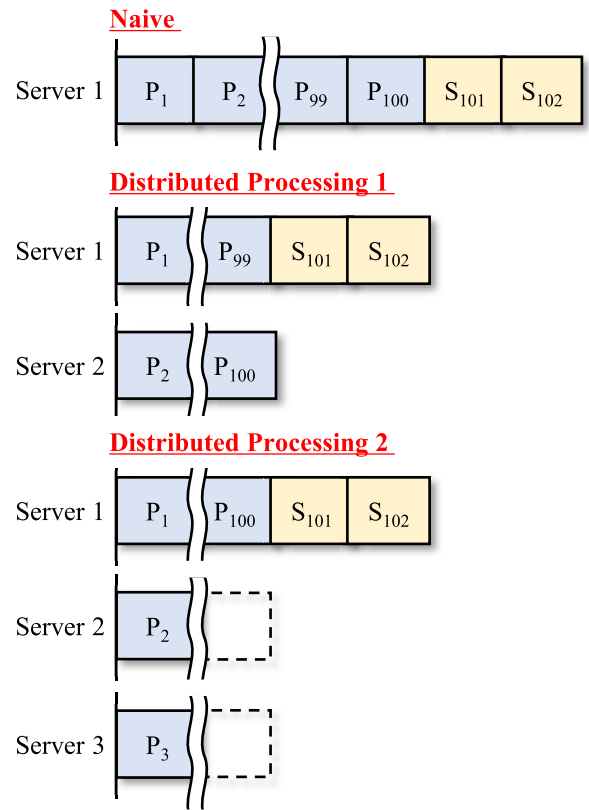


FIGURE 2. Conceptual diagram of distributed processing on optimization. P_1 - S_{102} indicates circuit run for one iteration. P_x means a process that can be executed in parallel and S_x means cannot. For example 1 in the middle column, a 1.96 times speedup is achieved and the parallelization efficiency is 98 %. In the case of example 2 in the lower column, the efficiency is 94 %.

to a single node, and numerous gRPC servers for computing were established.

Distributed processing by RPC can speed-up the VQE optimization process because the parameter vector θ consists of a plurality of scalar parameters θ_{pq} and θ_{pqrs} as shown in (5); thus, for each parameter, the value can be tuned independently on a certain optimizer. Fig. 2 shows a simple example of two and three gRPC servers with hundred parameters that achieves 1.96 and 2.83 times increase in speed. In this case, the parallelization efficiencies are $\frac{102}{2 \times 52} \approx 0.98$ and $\frac{102}{3 \times 36} \approx 0.94$, respectively. As a trend, the higher the number of parameters, the better the parallelization efficiency. Variational quantum algorithms are suitable because of their diverse parameters.

When dealing with sequential and parallel processing, the parallelization efficiency is generally derived by Amdahl's law, which is a famous theory of parallel computational acceleration. See Amdahl's law speed-up scaling factor (8) and efficiency (9).

$$S_{ideal} = \frac{N_p + N_s}{\frac{N_p}{n_{parallel}} + N_s}. \quad (8)$$

$$\epsilon_{ideal} = \frac{N_p + N_s}{N_p + N_s n_{parallel}}. \quad (9)$$

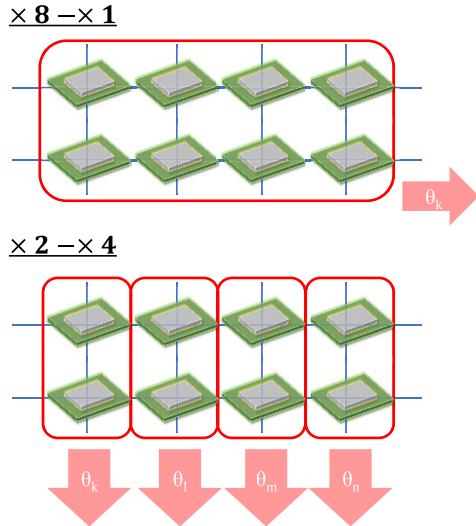


FIGURE 3. A schematic image of the combination of MPI parallel and distributed processing. The red square frame indicates MPI parallelism. In the image above, eight compute nodes are in MPI connection. In the image below, two nodes with MPI-connected configuration performs four distributed operations. Each configuration is assigned an individual parameter adjustment. The blue line indicates that the node is connected by InfiniBand. They were actually connected through switches, not directly.

$S_{ideal}, \epsilon_{ideal}, N_p, N_s,$ and $n_{parallel}$ indicate the speed-up scaling factor, parallelization efficiency, number of parallelizable process, sequential process, and parallelization. These formulae show that a sequential process becomes a bottleneck where $n_{parallel}$ gets larger. When distributed processing is performed by HPC cluster, creating a surplus of nodes is possible, as shown in Fig. 2. The efficiency in that case is shown as follows (10).

$$\epsilon_{DP} = \frac{N_p + N_s}{n_{server} \lceil \frac{N_p}{n_{server}} + N_s \rceil}. \quad (10)$$

$\lceil \cdot \rceil$ represents the ceiling function, which is the smallest integer greater than value in $\lceil \cdot \rceil$. ϵ_{DP} and n_{server} indicate the efficiency of distributed processing and the number of gRPC servers, respectively. N_p and N_s are derived from the number of variational parameters; the relational expression depends on the optimizer. For SLSQP we used, $N_p = (\# \text{ of parameter})$ and $N_s = 3$. When the parallelization efficiency of MPI parallel is ϵ_{MPI} , the efficiency of the whole system can be expressed by $\epsilon_{MPI}\epsilon_{DP}$. If no distributed processing is performed, the efficiency is equal to ϵ_{MPI} .

The proposed method aims to maximize $\epsilon_{MPI}\epsilon_{DP}$ by combining parallel execution by MPI and distributed processing. As an example of a small 8 nodes system, there are four patterns of node usage. $\times(\text{number of MPI parallelizations}) - \times(\text{number of gRPC servers})$ is expressed as $\times 8 - \times 1, \times 4 - \times 2, \times 2 - \times 4$ and $\times 1 - \times 8$. Fig. 3 shows an example.

In our validation, the best combination with a maximum of 1024 nodes was selected. Regarding MPI, ϵ_{MPI} is unpredictable because communication overhead is highly affected. ϵ_{MPI} was obtained by executing a quantum circuit once for all MPI patterns. In other words, the following

configurations were performed: $\times 1 - \times 1, \times 2 - \times 1, \times 4 - \times 1, \times 8 - \times 1, \times 16 - \times 1, \times 32 - \times 1, \times 64 - \times 1, \times 128 - \times 1, \times 256 - \times 1, \times 512 - \times 1, \times 1024 - \times 1$. Regarding distributed processing, ϵ_{DP} can be predicted by (10). Therefore, $\epsilon_{MPI}\epsilon_{DP}$ was figured out.

Distributed processing is also affected by communication overhead, whereas the effect should be negligible for large-scale calculations. A comparison between the theoretical values and the results obtained in the actual simulation is presented in the section III-B.

D. HAMILTONIAN TERMS CUTOFF

Calculating the expectation values of the Hamiltonian with the generated wave functions $\langle \psi | H | \psi \rangle$ must be a major bottleneck in the simulation elapsed time. The reason is obtaining an expectation value that requires an operations for exponentially increasing computational basis and the number of observable terms scale $O(n^4)$ in the quantum chemical calculation problem of fermionic model Hamiltonian [39].

The Hamiltonian terms are represented by (2), (3), and (4). Since the expectation value of the observables is limited to the range of $-1 \leq \langle \psi | P | \psi \rangle \leq +1$, it may be useful to cutoff terms with small coefficients to slim down Hamiltonian because that of the actual term wP moves in the range of $-|w| \leq \langle \psi | wP | \psi \rangle \leq +|w|$.

In large-scale simulations, knowing the relationship between cutoff threshold, accuracy, and simulation time is difficult because the obtained accuracy is not exactly known until the simulation is completed and energy converges to E_{gd} . Our approach determines the Hamiltonian cutoff threshold Th_1 by estimating the accuracy from a complete run of a relatively small qubit problem. Algorithm 1 shows pseudocode for obtaining the Hamiltonian cutoff ratio.

Specifically for 32-qubit problem, the effect on the accuracy was considered by executing the 28-qubit problem simulation until convergence for several different cutoff ratios. The cutoff ratio is determined to minimize the number of Hamiltonian terms while satisfying the required accuracy. Then, moving on to the 32-qubit problem, the cut-off threshold Th_1 is determined such that the Hamiltonian term is left by its ratio.

Algorithm 1 Get Hamiltonian Cutoff Ratio

Require: Smaller qubit Hamiltonian $H = \sum_1^N w_i P_i$

Require: $a < b \Rightarrow |w_a| \geq |w_b|$

Require: Required accuracy ΔE

$ratio \leftarrow 1$

while $ratio \geq 0.1$ **do**

$E_{gd}^{ratio} \leftarrow \text{executeVQE}(H = \sum_1^{(N \times ratio)} w_i P_i)$

if $|E_{ratio} - E_1| \geq \Delta E$ **then**

return $(ratio + 0.1)$

end if

end while

return $ratio$

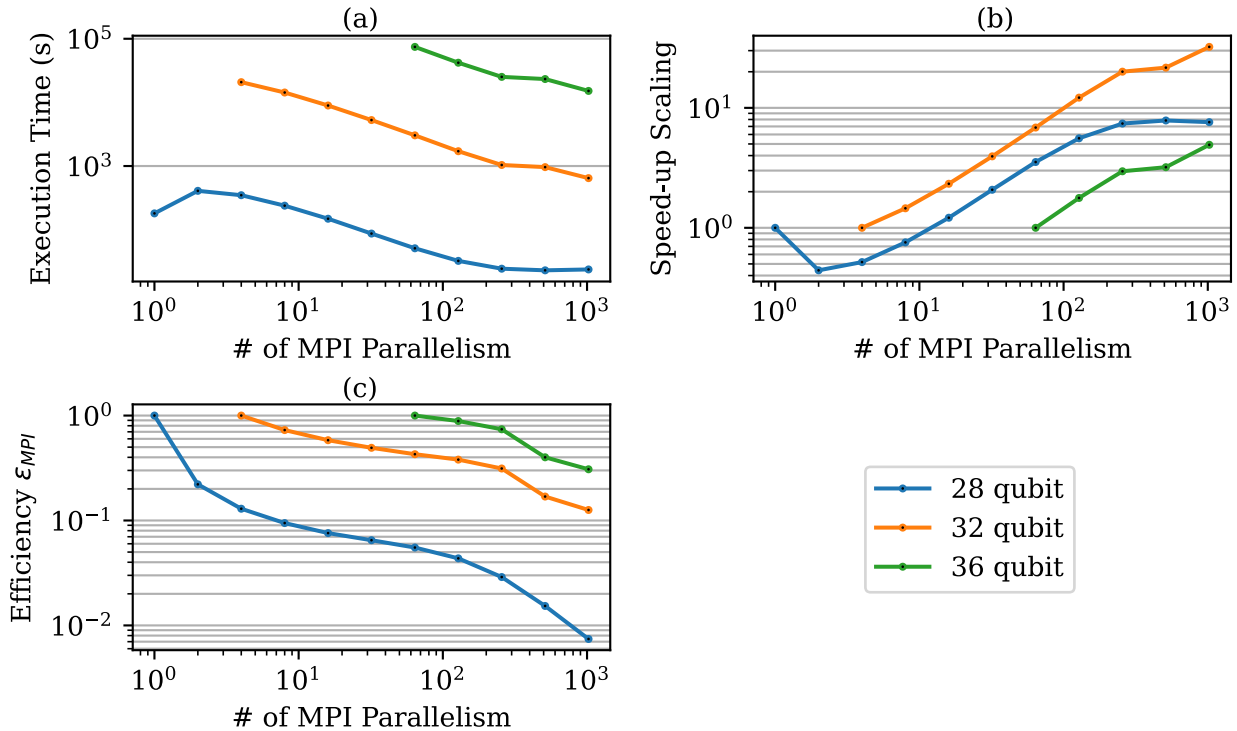


FIGURE 4. MPI parallel simulation results. The number of parallels is plotted as a power of 2, from 1 to 1024. The three subgraphs are derived from the same result. (a) Elapsed time per quantum circuit execution. (b) Speed-up scaling factor from the minimum configuration. (c) Parallelization efficiency ϵ_{MPI} .

III. RESULTS

Data obtained in previous studies reveal the properties of the fermionic model VQE simulation up to 20 qubits [18], [19], [20]. In this study, we present the results and characteristics of the VQE simulations up to 36 qubits run on an FX700 HPC cluster system. As new techniques, we yield parallel execution using a combination of MPI and distributed processing and a Hamiltonian terms cutoff.

This section is divided into the following subsections:

- A. MPI parallelism
- B. Distributed processing
- C. MPI and distributed processing
- D. Hamiltonian terms cutoff
- E. 32-qubit complete VQE simulation

The effects of parallel execution are discussed in terms of MPI parallelism, distributed processing, and their combination. Then Hamiltonian terms are discussed in subsection D. Finally, the complete simulation of the 32-qubit problem to convergence is shown.

A. MPI PARALLELISM

Up to 36-qubit calculation with mpiQulacs has been reported in a previous study [28]. The study showed the usefulness of quantum computational simulations using MPI for systems with up to 64 A64FX processors. In this study, we conducted VQE simulations of 28, 32, and 36 qubits with different degrees of MPI parallelism; furthermore, the elapsed time were obtained. To save the total running time, the durations

of single quantum circuit execution including an expectation value measurement were compared.

Fig. 4(a) shows the relationship between number of the MPI parallelisms and execution time. In the 28-qubit computation, the computation speed decreased when the number of MPI parallelisms was increased from 1 to 2. This implies that the communication overhead time exceeded the effect of increasing processing capability. We barely managed to overcome a single node with over 16 MPI parallelisms. In other words, single node computation was very efficient.

For the three datasets, the slopes of the trendlines with more than two parallelisms were almost identical. This indicates that the time reduction rate is less associated with the number of qubits in this MPI parallel region. The larger the number of MPI parallelisms, the shorter the execution time, with a negative slope. Since $\frac{\Delta \epsilon_{MPI}}{\Delta n_{parallel}} < 0$, doubling the number of MPI parallelisms does not reduce the time in half. Therefore, the slopes were a little gradual. This can also be observed from the slopes of more than 256 parallelisms.

In Fig. 4(b), a speed-up scaling factor from the minimum configuration can be observed. Here, the minimum configuration was $\times 1 - \times 1$ for 28 qubits, $\times 4 - \times 1$ for 32 qubits, and $\times 64 - \times 1$ for 36 qubits. Theoretically, if there is no communication overhead, the scaling in the $\times 1024 - \times 1$ configuration should be larger for those with smaller qubits. However, in reality, because of the communication overhead, single node computation can be faster instead of following the MPI parallel trend. Accordingly, the

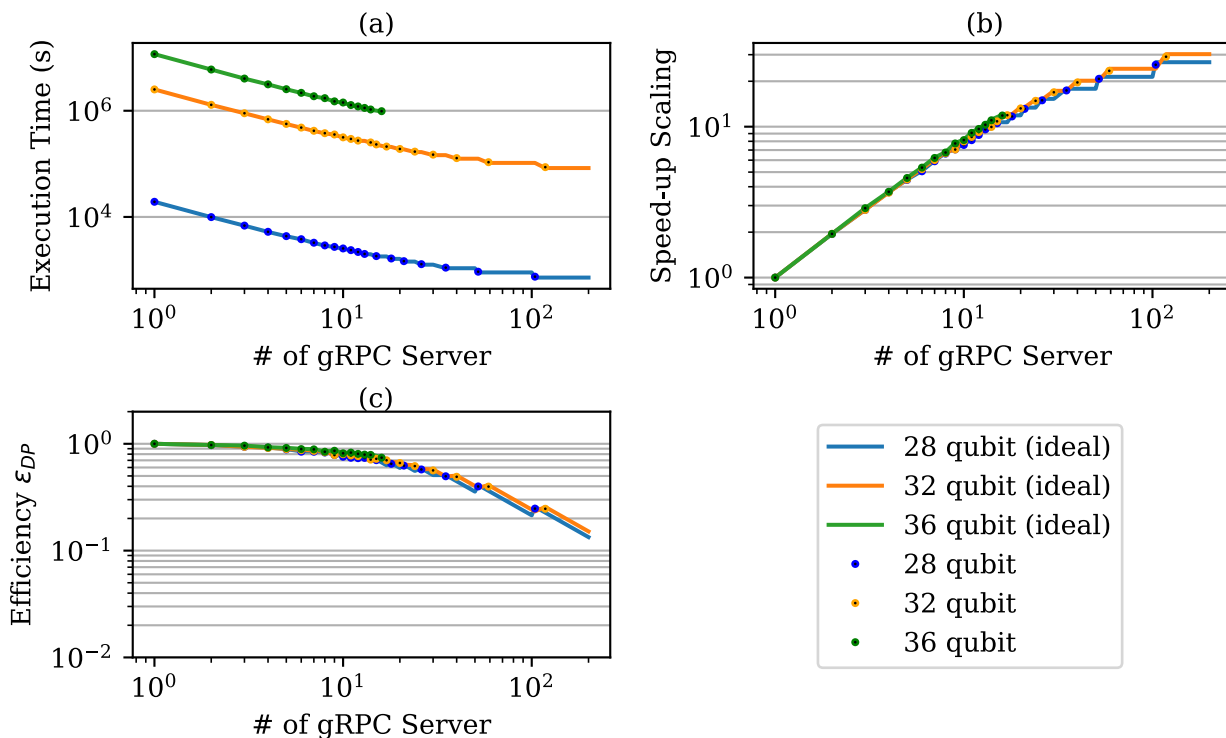


FIGURE 5. Distributed processing simulation results and estimated ideal values. (a),(b) and (c) represents execute time, speed-up scaling and efficiency, respectively as same as Fig. 4. Execute time is during 1 iteration starts to finishes. Plots mean actual obtained values from simulation. Line means ideal expected values driven from (10).

scaling of 28 qubits relative to a single node was worse than that of 32 qubits and better than that of 36 qubits.

Fig. 4(c) shows the efficiency ϵ_{MPI} . The trendline had a negative slope because $\frac{\Delta\epsilon_{mpi}}{\Delta n_{parallel}} < 0$. Since the efficiency was compared with the minimum configuration, larger values were naturally obtained in the order of 36, 32, and 28 qubits when comparing on $\times 1024 - \times 1$. For example, 32 qubits was 20 times faster in a $\times 512 - \times 1$ configuration but 128 times more with MPI parallelism, resulting in an efficiency of $\frac{20}{128} \approx 0.16$. A sharp drop in efficiency was commonly observed for more than 256 nodes, suggesting a limit when the number of nodes is increased indefinitely.

The results revealed that MPI parallelism reliably improves the speed by increasing the number of parallelisms, whereas larger parallelisms have a restricted efficiency.

B. DISTRIBUTED PROCESSING

When the gRPC is used for distributed processing, the degree of parallelism is the same as the number of gRPC servers. A single node gRPC client sent instructions to multiple gRPC servers with up to 1024 nodes. The gRPC server consisted of 1, 4, and 64 nodes for 28-, 32-, and 36-qubit problems, respectively. For example, 32 qubits measurements are run in a $\times 4 - \times N$ configuration, where N is the number of gRPC servers. The distributed processing method ideally has a parallelization efficiency shown in (10) however, a communication overhead should exist between the gRPC client and the gRPC server; the objective is to verify this effect. Furthermore, this subsection compares the speed-up

scaling and the efficiency to MPI parallelism. Unlike Fig. 4, the comparison was based on the execution time of one iteration; because of taking into account the negative effects of the surplus nodes.

Fig. 5(a) shows the relationship between the number of gRPC servers and execution time. The theoretical ideal values are shown as solid lines and the actual values obtained in the simulations are shown as plots. First of all, the theoretical and measured values were almost the same. This implies that the communication overhead and the impact of server operations for distributed processing are relatively small. Second, the execution time trends were similar for all three problem settings. A performance decrease was not observed when the number of nodes was increased from one to two, which was observed in the MPI parallel. We found that the degree of performance improvement achieved by the distributed processing was stable.

Fig. 5(b) and (c) display speed-up scaling factor and efficiency ϵ_{DP} as well as Fig. 4. The three problem settings exhibited approximately the same scaling and efficiency. The subtle difference arose from the number of parameters; each had 104, 118 and 153 variational parameters in the circuit. The efficiency ϵ_{DP} decreased as the number of gRPC servers increased. This is not primarily due to the communication overhead. In the parameter optimization process, parallelizability processes are executed simultaneously on the gRPC servers. As a result, the parallelization efficiency decreases because of the increase in the proportion of the sequential processing part. This is known as Amdal's law.

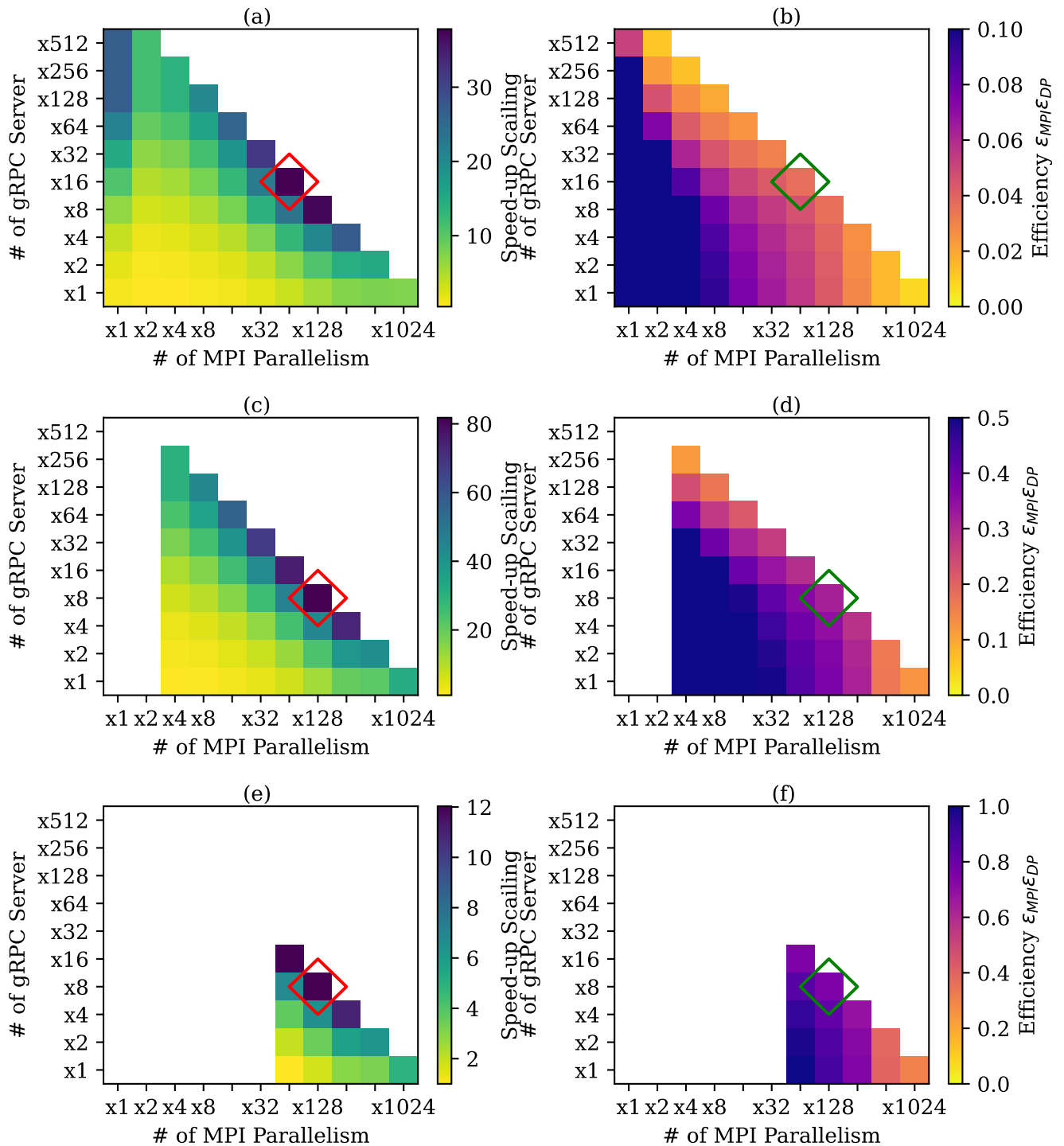


FIGURE 6. Combination of MPI and distributed processing. (a), (c) and (e) represent speed up scaling of 28, 32 and 36 qubits simulation. (b), (d) and (f) represent parallelization efficiency $\epsilon_{MPI\epsilon DP}$ of 28, 32 and 36 qubits.. The red and green frames in the figure indicate the combination with the highest speed up scaling factor and the most efficient combination when using 1024 nodes. White color areas are not feasible combinations. The speed up scaling and efficiency of the distributed processing are estimated from (10).

C. MPI AND DISTRIBUTED PROCESSING

Accelerating the simulation by combining MPI and distributed processing in a suitable ratio is one of the new attempts of this study; the effect is verified here. The effects of the independent method were discussed

in the previous subsections. The question is whether the combination provides improvement. Since the total number of available nodes is 1024, a combination such as $\times 256 - \times 16$ cannot be configured. Similarly, because a 32 qubits simulation requires at least four nodes of

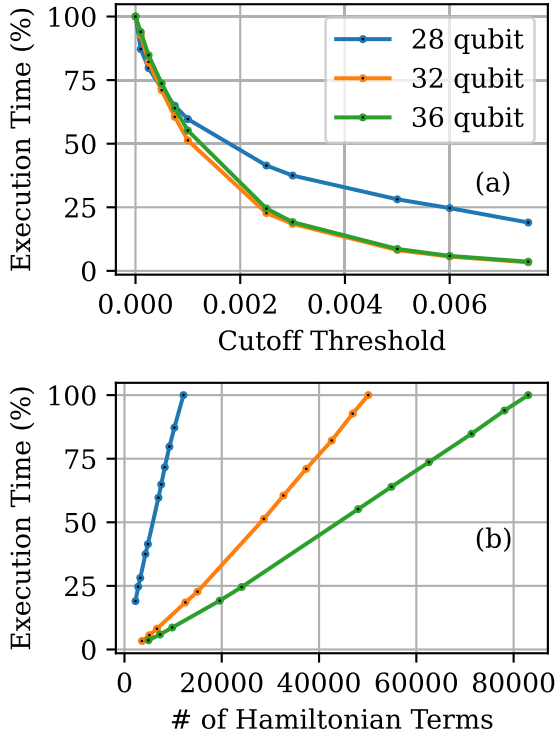


FIGURE 7. Effect of Hamiltonian term reduction on execution time. The two subgraphs derive from the same result. The time of one quantum circuit execution when the cutoff threshold set to 0, is normalized to 100%.

MPI parallel, a configuration such as $\times 1 - \times 64$ is not possible.

Fig. 6 shows heatmaps of the improvement in speed and efficiency. The white areas are not feasible combinations. The values on the left and bottom edges of the graph were equal to those obtained in Figs. 4 and 5, respectively. The left edge corresponds to independent distributed processing and the bottom edge corresponds to MPI parallel; as the degree of parallelism increases, the degree of speeding-up increases, and the parallelization efficiency deteriorates. The new results obtained here is provided at the center of the graph.

From (a), (c) and (e), the maximum speed-up scaling factors of 28, 32 and 36 qubits were obtained for $\times 64 - \times 16$, $\times 128 - \times 8$ and $\times 128 - \times 8$, respectively. While these three results agree that using all 1024 nodes is the fastest, and even faster when used in combination rather than independently. In particular, the 32 qubits results showed a 4-fold speed-up compared with using only the individual methods.

Whole parallelization efficiency $\epsilon_{MPI \in DP}$ is shown in (b), (d), and (f). Efficiency has the best value $\epsilon_{MPI \in DP} = 1$ for the minimum configuration, and tends to worsen as the degree of parallelism increases. It is the upper right outermost region, which means the line using 1024 nodes, which is important, where the most efficient combination corresponds to the one with the highest speed improvement represented by (a), (c) and (e).

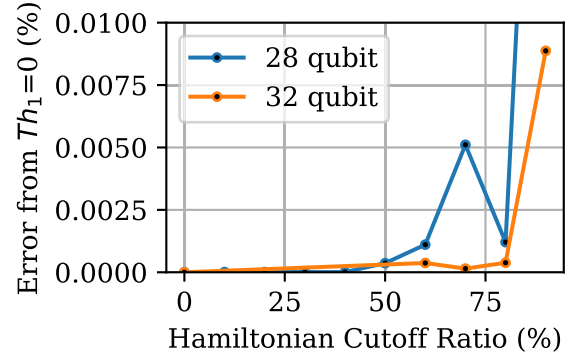


FIGURE 8. Accuracy and Hamiltonian cutoff ratio. The error indicates the difference from Egd at the threshold $Th_1=0$. The Hamiltonian cutoff ratio indicates the percentage of terms reduced.

These results suggest that MPI and distributed processing should be combined for the fastest and most efficient use of prepared nodes. The results are highly dependent on the communication overhead; different hardware communication configurations can lead to different ratio.

D. HAMILTONIAN TERMS CUTOFF

Here, we have seen how reducing the number of terms in the Hamiltonian improves the simulation speed and decreases the calculation accuracy. The Jordan-Wigner transformed Hamiltonian terms were sorted according to the absolute values of the coefficients. Then the values below the set threshold Th_1 were cutoff to slim down and reconstruct the Hamiltonian.

Fig. 7(a) displays the relationship between the Th_1 and execution time; the time corresponds to single quantum circuit execution with a minimum configuration. The time is normalized to 100% when the $Th_1 = 0$. The three colors of the plots show similar trends.

Fig. 7(b) displays the relationship between the number of Hamiltonian terms and execution time. A proportional relationship was observed, which can be imagined from the steps of the computation process, by calculating the terms individually.

Fig. 7 accounts that adjusting Th_1 in the range of 10^{-3} can dramatically reduce the time required to obtain the expected value $\langle \psi | H | \psi \rangle$.

Fig. 8 shows the relationship between the cutoff ratio and accuracy of the ground-state energy. The ratio means that the Th_1 was set to cutoff the terms in a particular percentage. For the 28-qubit problem, the accuracy was obtained with cutoff ratios in increments of 10%. For the 32-qubit problem, it was obtained at 0, 60, 70, 80, and 90% were used as references. The two problems show the same trend; the accuracy deteriorates as the number of terms to be cutoff increases. In both problem settings, the accuracy was significantly worse when the cutoff ratio was increased to 90%.

The objective of these measurements was to determine the Th_1 value for the complete simulation of 32 qubits. Based on these results, cutoff ratio was set to 70%.

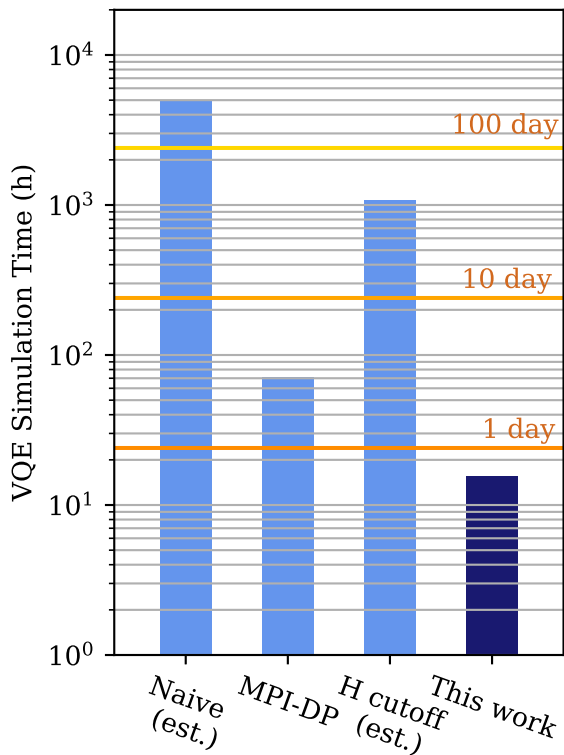


FIGURE 9. Overall VQE simulation time. The measurement time is only for the VQE main part; preprocessing is excluded. The preprocessing time is relatively small hence negligible. First and third from the left are estimates. The naive estimation done by multiplying the number of executions by the time required for each quantum circuit execution.

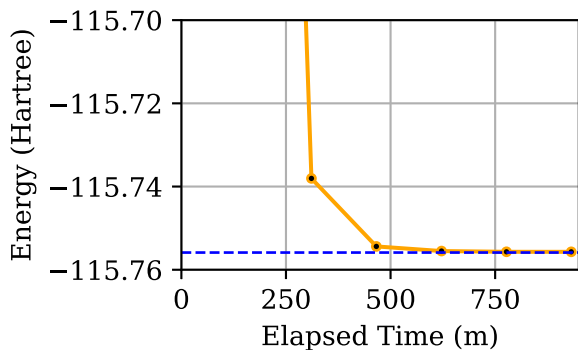


FIGURE 10. Energy convergence process of complete VQE. The plots mean the completion point of each iteration. The blue dashed line is the ground state energy obtained without the Hamiltonian cutoff.

E. 32-QUBIT COMPLETE VQE SIMULATION

Large-scale VQE simulations with fermionic Hamiltonian beyond 30 qubits have not been performed; however, we demonstrated VQE simulations at 32 qubits using the introduced techniques. The demonstration was conducted using a combination of MPI and distributed processing at an optimal ratio and using a Hamiltonian cutoff threshold of 0.0025. The threshold was set to reduce the Hamiltonian terms by 70%. The accuracy of the ground-state energy is targeted at 0.01 Hartree from that with $Th_1 = 0$, hence an error of 0.01% is acceptable. Moreover, the execution time for the comparison configuration is also indicated; however, this

TABLE 2. Obtained ground-state energy.

28-qubit problem: CO_2	
Method	Obtained ground-state energy (Hartree)
VQE cutoff ratio=90%	-185.2966
VQE cutoff ratio=80%	-185.2337
VQE cutoff ratio=70%	-185.2454
VQE cutoff ratio=60%	-185.2380
VQE cutoff ratio=50%	-185.2353
VQE cutoff ratio=40%	-185.2359
VQE cutoff ratio=30%	-185.2359
VQE cutoff ratio=20%	-185.2359
VQE cutoff ratio=10%	-185.2359
VQE cutoff ratio=0%	-185.2360
HF	-185.0678
CCSD	-185.2698
CCSD(T)	-185.2939
32-qubit problem: C_3H_6	
Method	Obtained ground-state energy (Hartree)
VQE cutoff ratio=90%	-115.7456
VQE cutoff ratio=70%	-115.7557
VQE cutoff ratio=0%	-115.7559
HF	-115.6603
CCSD	-115.8835
CCSD(T)	-115.8848

HF and CCSD mean Hartree-Fock and coupled cluster singles and doubles, respectively. Both are typical computational methods used for quantum chemical calculations. CCSD(T) is known as a gold standard method.

takes a very long time so that the estimation value is shown. The estimation value was obtained by measuring the time required for one quantum circuit execution and multiplying by the expected number of executions.

Fig. 9 shows the overall execution time of VQE except for preprocessing. This is because the time for preprocessing is very short and does not affect the overall time. The time required by this work, implementing newly two techniques, was approximately 15 hours, which is a realistic acceptable time for conducting algorithm research. The energy convergence process is shown in Fig. 10 and the ground-state energies are listed in Table 2. Typical conventional calculation methods were also compared. This means that the proper ground-state energy could be obtained by our VQE.

When none or only one of the techniques was applied, the estimated times were 200 days for naive simulation, 3 days for the MPI-DP combination, and 40 days for Hamiltonian cutoff. The improvement ratio from the naive estimation to this work was approximately 200. This is somewhat consistent with the approximate percentage improvement $\frac{82}{(1-0.7)}$ derived from the discussion in subsections III-C and III-D. The effect of the two techniques can be represented approximately by simple multiplication because they are independent and do not affect each other.

This result indicates that by implementing the two techniques, the fermionic model VQE simulation until energy convergence can be completed within an acceptable time to proceed with the study of quantum algorithms. On the other hand, it was estimated that using only a single technique would take more than three days. Increasing the number

of nodes from 1024 is another way to solve this problem; however, it incurs additional costs.

IV. CONCLUSION

Previous research has attempted to further develop quantum algorithms by simulating quantum calculations using a large number of qubits. Despite these significant steps, simulations of variational quantum algorithms, which require multiple quantum circuit executions, had have problem of not completing them in an acceptable time. This is because, as the number of qubits and variational parameters increases, the number of quantum circuit executions for optimization is high. Therefore, to the best of the authors' knowledge, VQE for solving the Hamiltonian of fermionic model has only been reported up to 20 qubits [20]. In our method, we added newly techniques using the characteristics of the algorithm with parameter optimization and expectation value calculation. One is to run the 1024 FX700 nodes not simply in parallel with MPI, but concurrently through distributed processing. The another is to slim down the Hamiltonian used to calculate the expectation value without losing too much accuracy.

We demonstrated that 32-qubit VQE simulation could be completed in 15 hours. This is an acceptable time to study the algorithm while changing the gates implemented in ansatz or the problem settings. Large-scale VQE simulations can be performed by the efficient use of HPC cluster systems. In addition, two newly techniques were verified. The computational speed-up owing to distributed processing was not significantly affected by the communication overhead. This behavior followed the modified Amdahl's law. The combination of MPI and distributed processing was found to be faster than the use of only one method. In particular, for the 32-qubit problem, the efficient combination got four times faster than only MPI parallelism or distributed processing. For the Hamiltonian slimming down, the error was 0.05% even when the terms were reduced by 70% for the 28-qubit problem. In verifying VQE by simulation, the result implies that the calculation time can be reduced by several dozen percent by this method if high calculation accuracy is not required.

The 32 qubits simulation demonstration updated the previously reported maximum number of qubits for VQE simulations of fermionic models. The two newly techniques should also provide improvements not only for VQE but also for other variational algorithms such as QAOA and QML. Since this work takes the form of a statevector simulation, the effect of noise can also be simulated. Running simulations and experiments on real quantum hardware make it possible to separate the effects of noise from the shortcomings of the algorithm and extract them from the experimental results.

Several factors should be considered. For both MPI and distributed processing, communication processing has an effect; therefore, if the communication environment is not well developed, execution time may increase significantly and become unstable. We chose SLSQP as the optimizer. Other optimization methods may not be as effective as those

in this study if the ratio of sequential processing is large. Conversely, a more parallelism-dominated optimizer could be further improved.

This study allowed us to increase the number of qubits that can be simulated in research of variational quantum algorithm. These techniques will contribute to a practical algorithm search for NISQ.

In future work, we would like to challenge the completion of the VQE simulation with more than 32 qubits in realistic time. We would also like to expand the scope and work on variational quantum algorithms other than VQE. Leveraging many nodes simultaneously creates a variety of operational problems such as communication. It is necessary to solve this problem using general multi-node know-how and unique use of quantum simulation.

REFERENCES

- [1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982.
- [2] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
- [3] P. J. J. O'Malley et al., "Scalable quantum simulation of molecular energies," *Phys. Rev. X*, vol. 6, Jul. 2016, Art. no. 031007.
- [4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, Sep. 2017.
- [5] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya, T. Yamamoto, T. Yan, T. Kawakubo, Y. O. Nakagawa, Y. Ibe, Y. Zhang, H. Yamashita, H. Yoshimura, A. Hayashi, and K. Fujii, "Qulacs: A fast and versatile quantum circuit simulator for research purpose," *Quantum*, vol. 5, p. 559, Oct. 2021.
- [6] R. Orús, "Tensor networks for complex quantum systems," *Nature Rev. Phys.*, vol. 1, no. 9, pp. 538–550, Aug. 2019.
- [7] A. Abdollahi and M. Pedram, "Analysis and synthesis of quantum circuits by using quantum decision diagrams," in *Proc. Design Autom. Test Eur. Conf.*, Mar. 2006, pp. 1–6.
- [8] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014.
- [9] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, Feb. 2016, Art. no. 023023.
- [10] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, "The variational quantum eigensolver: A review of methods and best practices," *Phys. Rep.*, vol. 986, pp. 1–128, Nov. 2022.
- [11] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, "VQE method: A short survey and recent developments," *Mater. Theory*, vol. 6, no. 1, p. 2, Jan. 2022.
- [12] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.
- [13] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp. Phys.*, vol. 56, no. 2, pp. 172–185, 2015.
- [14] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, Sep. 2017.
- [15] K. M. Nakanishi, K. Mitarai, and K. Fujii, "Subspace-search variational quantum eigensolver for excited states," *Phys. Rev. Res.*, vol. 1, no. 3, Oct. 2019, Art. no. 033062.
- [16] R. M. Parrish, E. G. Hohenstein, P. L. McMahon, and T. J. Martínez, "Quantum computation of electronic transitions using a variational quantum eigensolver," *Phys. Rev. Lett.*, vol. 122, no. 23, Jun. 2019, Art. no. 033062.
- [17] O. Higgott, D. Wang, and S. Brierley, "Variational quantum computation of excited states," *Quantum*, vol. 3, p. 156, Jul. 2019.

- [18] H. R. Grimsley, G. S. Barron, E. Barnes, S. E. Economou, and N. J. Mayhall, "Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus," *NPJ Quantum Inf.*, vol. 9, no. 1, p. 19, Mar. 2023.
- [19] J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, "Generalized unitary coupled cluster wave functions for quantum computation," *J. Chem. Theory Comput.*, vol. 15, no. 1, pp. 311–324, Jan. 2019.
- [20] G.-L.-R. Anselmetti, D. Wierichs, C. Gogolin, and R. M. Parrish, "Local, expressive, quantum-number-preserving VQE nsätze for fermionic systems," *New J. Phys.*, vol. 23, no. 11, Nov. 2021, Art. no. 113010.
- [21] M. S. Jattana, F. Jin, H. De Raedt, and K. Michielsen, "Improved variational quantum eigensolver via quasidynamical evolution," *Phys. Rev. Appl.*, vol. 19, no. 2, Feb. 2023, Art. no. 024047.
- [22] B. Villalonga, S. Boixo, B. Nelson, C. Henze, E. Rieffel, R. Biswas, and S. Mandró, "A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware," *NPJ Quantum Inf.*, vol. 5, no. 1, p. 86, Oct. 2019.
- [23] B. Villalonga, D. Lyakh, S. Boixo, H. Neven, T. S. Humble, R. Biswas, E. G. Rieffel, A. Ho, and S. Mandró, "Establishing the quantum supremacy frontier with a 281 Pflop/s simulation," *Quantum Sci. Technol.*, vol. 5, no. 3, Apr. 2020, Art. no. 034003.
- [24] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, "PySCF: The Python-based simulations of chemistry framework," *Rev. Comput. Mol. Sci.*, vol. 8, no. 1, p. e1340, 2018.
- [25] J. R. McClean et al., "OpenFermion: The electronic structure package for quantum computers," *Quantum Sci. Technol.*, vol. 5, no. 3, Jun. 2020, Art. no. 034014.
- [26] P. Virtanen, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, Feb. 2020.
- [27] G. G. Guerreschi, J. Hogaboam, F. Baruffa, and N. P. D. Sawaya, "Intel quantum simulator: A cloud-ready high-performance simulator of quantum circuits," *Quantum Sci. Technol.*, vol. 5, no. 3, May 2020, Art. no. 034007.
- [28] S. Imamura, M. Yamazaki, T. Honda, A. Kasagi, A. Tabuchi, H. Nakao, N. Fukumoto, and K. Nakashima, "MpiQulacs: A distributed quantum computer simulator for A64FX-based cluster systems," 2022, *arXiv:2203.16044*.
- [29] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm," *NPJ Quantum Inf.*, vol. 6, no. 1, p. 10, Jan. 2020.
- [30] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," *Quantum Sci. Technol.*, vol. 4, Oct. 2019, Art. no. 014008.
- [31] Y. Matsuzawa and Y. Kurashige, "Jastrow-type decomposition in quantum chemistry for low-depth quantum circuits," *J. Chem. Theory Comput.*, vol. 16, no. 2, pp. 944–952, Feb. 2020.
- [32] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Izmaylov, "Qubit coupled cluster method: A systematic approach to quantum chemistry on a quantum computer," *J. Chem. Theory Comput.*, vol. 14, no. 12, pp. 6317–6326, Dec. 2018.
- [33] M. Kühn, S. Zanker, P. Deglmann, M. Marthaler, and H. Weiß, "Accuracy and resource estimations for quantum chemistry on a near-term quantum computer," *J. Chem. Theory Comput.*, vol. 15, no. 9, pp. 4764–4780, Sep. 2019.
- [34] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Commun.*, vol. 10, no. 1, p. 3007, Jul. 2019.
- [35] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 6, Dec. 2018, Art. no. 062339.
- [36] P. Jordan and E. Wigner, "Über das Paulische Äquivalenzverbot," *Zeitschrift für Physik*, vol. 47, pp. 631–651, Sep. 1928.
- [37] S. B. Bravyi and A. Y. Kitaev, "Fermionic quantum computation," *Ann. Phys.*, vol. 298, no. 1, pp. 210–226, May 2002.
- [38] J. T. Seeley, M. J. Richard, and P. J. Love, "The Bravyi–Kitaev transformation for quantum computation of electronic structure," *J. Chem. Phys.*, vol. 137, no. 22, Dec. 2012, Art. no. 224109.
- [39] P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong, " $O(N^3)$ measurement cost for variational quantum eigensolver on molecular Hamiltonians," *IEEE Trans. Quantum Eng.*, vol. 1, pp. 1–24, 2020.
- [40] *gRPC A High Performance, Open Source Universal RPC Framework*. Accessed: Feb. 10, 2024. [Online]. Available: <https://grpc.io/>



MIKIO MORITA received the B.S. and M.S. degrees in astrophysics from Waseda University, Tokyo, Japan, in 2011.

He joined Fujitsu Ltd., in 2011. He was a Hardware Engineer of server and belonged to the SPEC Organization and the Standard Performance Committee, for benchmarking performance of various computers. He has been a Researcher of quantum computing, since 2020. His research interests include the performance benchmarking of overall computing, including classical and quantum principles.

Mr. Morita was a recipient of the SPEC Contributor Award, in 2019.



YOSHINORI TOMITA received the B.E. and M.E. degrees from Tokyo Institute of Technology, in 1993 and 1995, respectively.

He has been with Fujitsu Quantum-Inspired Computing Digital Annealer Unit. He is currently working on practical use cases of quantum simulator with Fujitsu Ltd. His research interests include optimization problems and simulation.



JUNPEI KOYAMA received the B.E., M.E., and Ph.D. degrees from The University of Tokyo, Tokyo, Japan, in 2003, 2005, and 2008, respectively.

He joined Fujitsu Laboratories Ltd., in 2008. He was engaged in the development of video coding technology, until 2018. He is currently working on the development of quantum/quantum-inspired technology. His research interests include quantum computing applications and multimedia signal processing.



KOICHI KIMURA received the B.E. degree from Shinshu University, Nagano, in 1990.

After that, he joined Fujitsu Ltd., and has been developing lightweight material and bioplastic for PC housing. Next, he explored new applications for new computers (digital annealers) developed by Fujitsu Ltd., inspired by quantum computing. Currently, he has begun researching applications for quantum computing. He is a member of Japan Society of Mechanical Engineers.

...