US 20250264922A1

(19) United States

(12) Patent Application Publication

KANG et al.

(10) Pub. No.: US 2025/0264922 A1

(43) Pub. Date: Aug. 21, 2025

(54) **APPARATUS AND METHOD FOR GPU POWER MANAGEMENT IN DISTRIBUTED DEEP LEARNING**

(71) Applicant: **Electronics and Telecommunications Research Institute**, Daejeon (KR)

(72) Inventors: **Ki-Dong KANG**, Daejeon (KR);
Hong-Yeon Kim, Daejeon (KR);
Baik-Song An, Seoul (KR);
Myung-Hoon Cha, Daejeon (KR)

(73) Assignee: **Electronics and Telecommunications Research Institute**, Daejeon (KR)

(21) Appl. No.: **18/933,007**

(22) Filed: **Oct. 31, 2024**

(30) **Foreign Application Priority Data**

Feb. 20, 2024 (KR) 10-2024-0024273

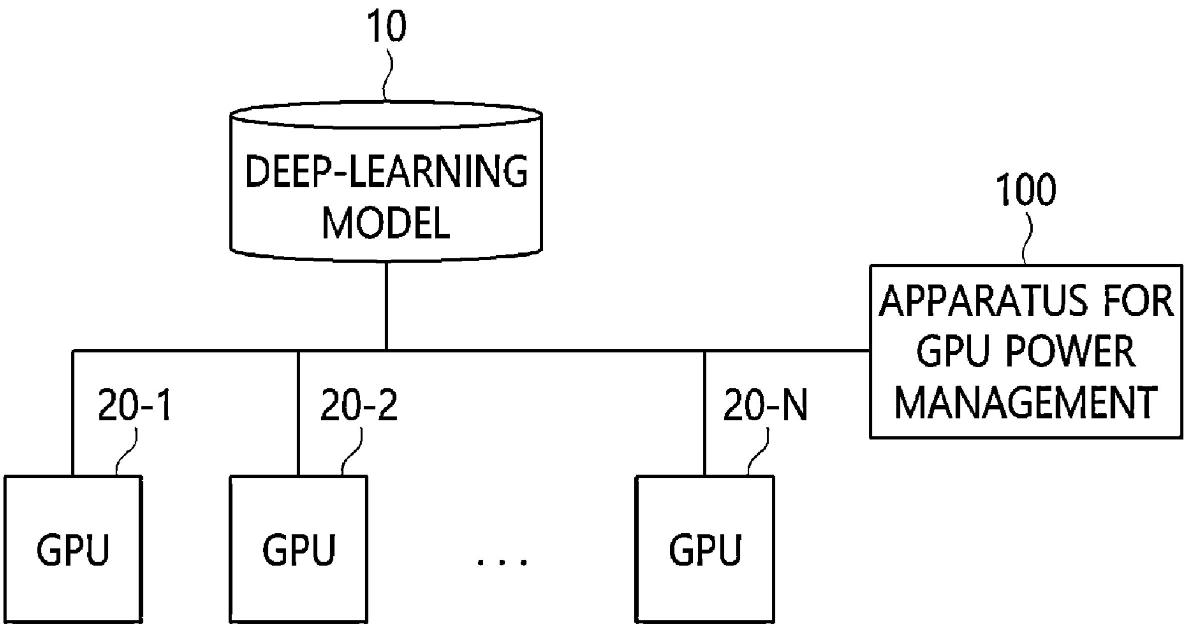
Publication Classification

(51) **Int. Cl.**
G06F 1/26 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 1/26* (2013.01)

(57) **ABSTRACT**

Disclosed herein is an apparatus and method for Graphics Processing Unit (GPU) power management in distributed deep learning. In the method, a training process or inference process of the distributed deep learning is performed through two or more GPUs, and the method may include identifying at least one communication section during the training process or inference process of the distributed deep learning and performing control such that the voltage and frequency of the GPU are optimized during the at least one communication section.



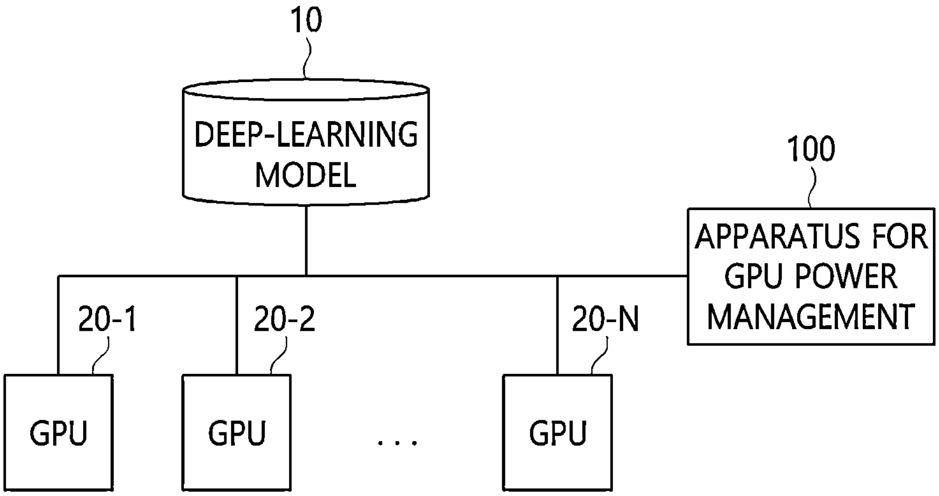

10
DEEP-LEARNING MODEL
100
APPARATUS FOR GPU POWER MANAGEMENT
20-1
20-2
20-N
GPU
GPU
. . .
GPU

FIG. 1

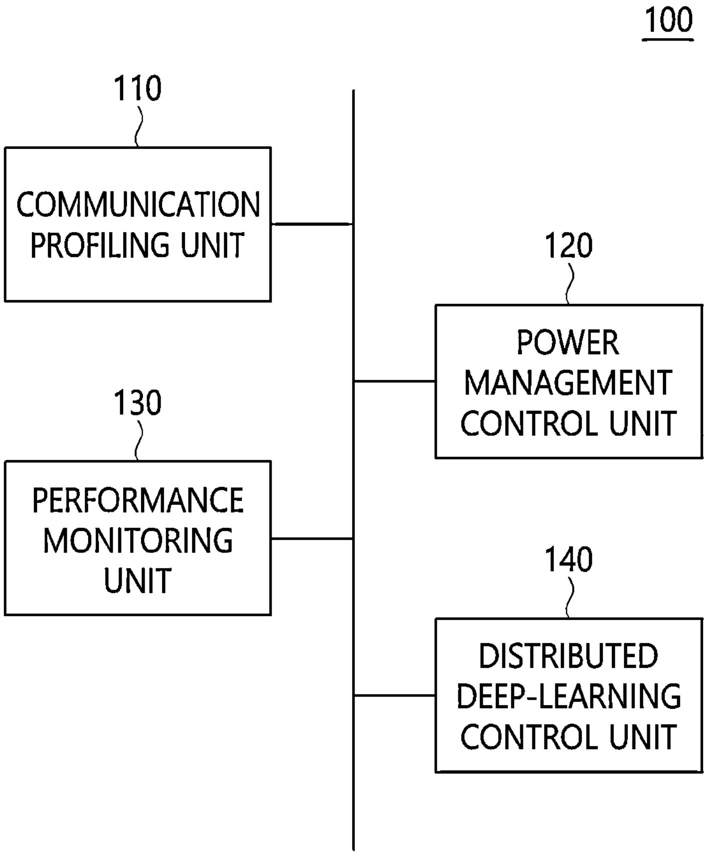

100
110
COMMUNICATION PROFILING UNIT
120
POWER MANAGEMENT CONTROL UNIT
130
PERFORMANCE MONITORING UNIT
140
DISTRIBUTED DEEP-LEARNING CONTROL UNIT

FIG. 2

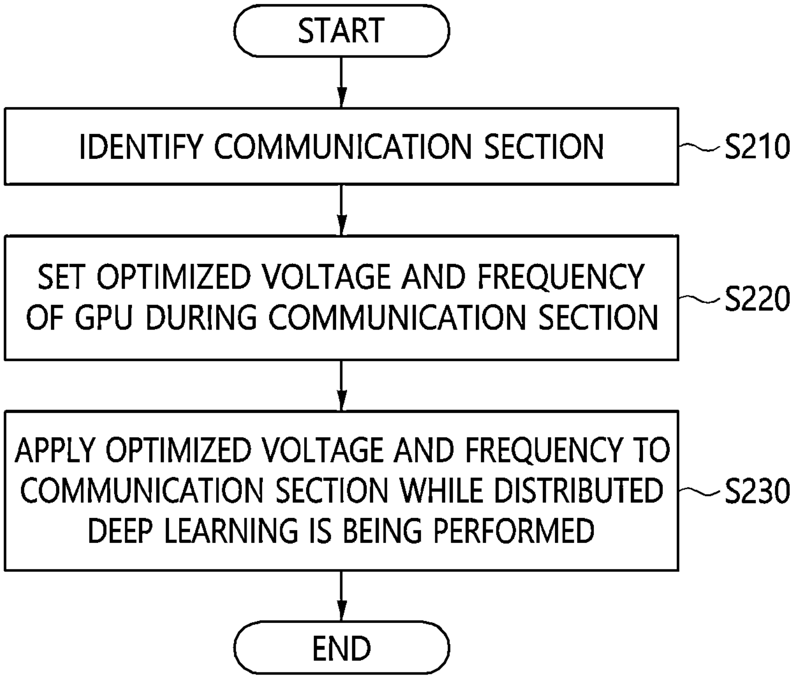

START
IDENTIFY COMMUNICATION SECTION
S210
SET OPTIMIZED VOLTAGE AND FREQUENCY OF GPU DURING COMMUNICATION SECTION
S220
APPLY OPTIMIZED VOLTAGE AND FREQUENCY TO COMMUNICATION SECTION WHILE DISTRIBUTED DEEP LEARNING IS BEING PERFORMED
S230
END

FIG. 3

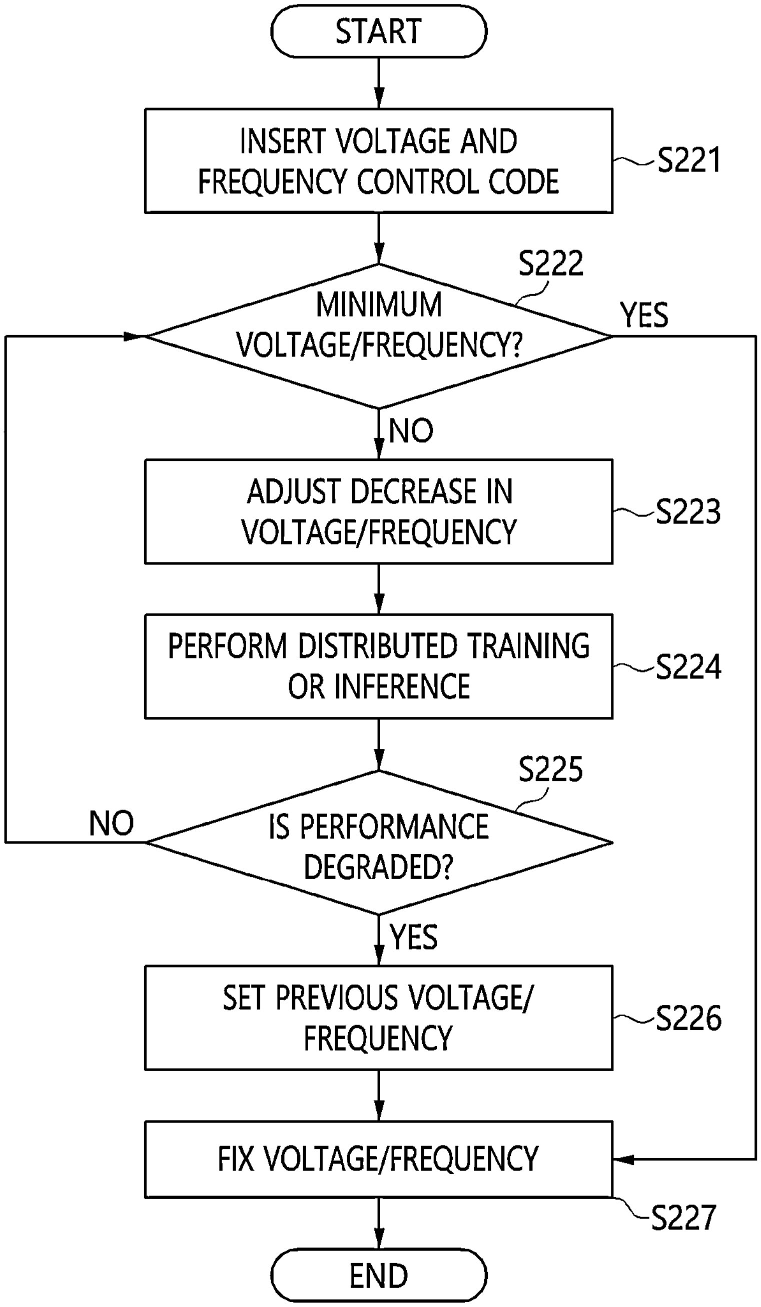

START
INSERT VOLTAGE AND FREQUENCY CONTROL CODE
S221
S222
MINIMUM VOLTAGE/FREQUENCY?
YES
NO
ADJUST DECREASE IN VOLTAGE/FREQUENCY
S223
PERFORM DISTRIBUTED TRAINING OR INFERENCE
S224
S225
NO
IS PERFORMANCE DEGRADED?
YES
SET PREVIOUS VOLTAGE/ FREQUENCY
S226
FIX VOLTAGE/FREQUENCY
S227
END

FIG. 4

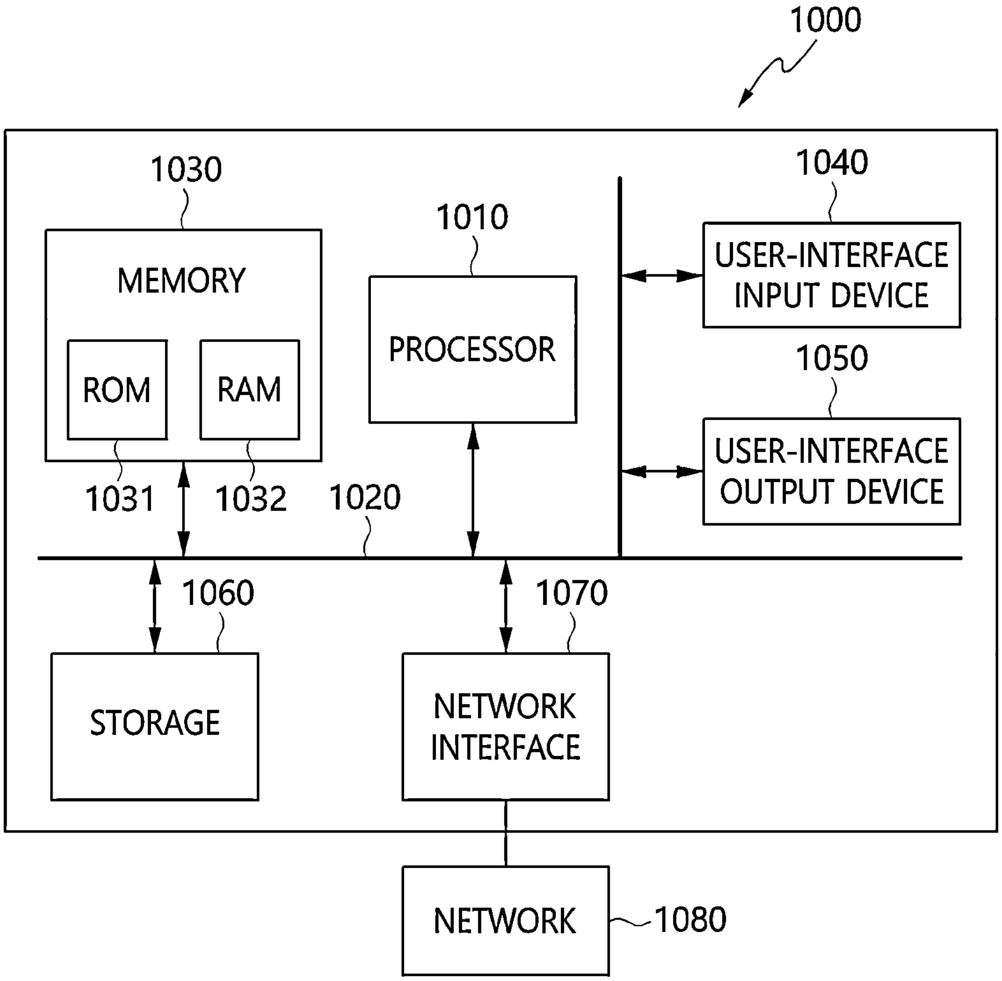

1000
1030
MEMORY
ROM
RAM
1031
1032
1010
PROCESSOR
1020
1040
USER-INTERFACE INPUT DEVICE
1050
USER-INTERFACE OUTPUT DEVICE
1060
STORAGE
1070
NETWORK INTERFACE
NETWORK
1080

FIG. 5

APPARATUS AND METHOD FOR GPU POWER MANAGEMENT IN DISTRIBUTED DEEP LEARNING

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of Korean Patent Application No. 10-2024-0024273, filed Feb. 20, 2024, which is hereby incorporated by reference in its entirety into this application.

BACKGROUND OF THE INVENTION

1. Technical Field

[0002] The disclosed embodiment relates to technology for Graphics Processing Unit (GPU) power management in distributed training or inference of deep learning.

2. Description of the Related Art

[0003] As deep-learning models have recently become larger, distributed deep learning, which requires multiple Graphics Processing Units (GPUs) for training and inference of deep-learning models, has become more common. At the same time, as multiple GPUs are used for such large-scale distributed deep learning, energy consumption emerges as a new problem.

[0004] As a solution to this problem, a GPU power management method that improves energy efficiency without learning performance degradation by utilizing pipeline bubble time caused in distributed deep-learning training based on pipeline parallelism, which is known as inter-operator parallelism, has been proposed.

[0005] However, because the proposed method uses pipeline bubbles for GPU power management, it is difficult to apply the method to distributed deep learning based on data parallelism and tensor parallelism, which are known as intra-operator parallelism.

SUMMARY OF THE INVENTION

[0006] An object of the disclosed embodiment is to manage GPU power so as to mitigate energy inefficiency caused in a distributed training or inference process of deep learning.

[0007] Another object of the disclosed embodiment is to enable GPU power management to be applied not only to distributed deep learning based on pipeline parallelism but also to distributed deep learning based on data parallelism and tensor parallelism.

[0008] A method for Graphics Processing Unit (GPU) power management in distributed deep learning, the training process or inference process of which is performed through two or more GPUs, according to an embodiment may include identifying at least one communication section during the training process or inference process of the distributed deep learning and performing control such that the voltage and frequency of the GPU are optimized during the at least one communication section.

[0009] Here, performing the control may comprise setting the voltage and frequency of the GPU during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

[0010] Here, performing the control may include inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

[0011] Here, performing the control may include checking whether the voltage and frequency of the GPU are minimum values; and setting the voltage and frequency as the optimal voltage and frequency of the GPU when the voltage and frequency are the minimum values.

[0012] Here, performing the control may include checking whether the voltage and frequency of the GPU are minimum values, decreasing the voltage and frequency of the GPU by a predetermined value when the voltage and frequency are not the minimum values, monitoring performance degradation after performing distributed training or inference in the GPU, and setting a previous voltage and frequency as the optimal voltage and frequency of the GPU when performance degradation occurs.

[0013] Here, performing the control may proceed to decreasing the voltage and frequency of the GPU by the predetermined value when performance degradation does not occur.

[0014] Here, the method for GPU power management in distributed deep learning according to an embodiment may further include applying the optimized voltage and frequency of the GPU to the communication section while the distributed deep learning is being performed.

[0015] Here, identifying the at least one communication section, performing the control, and applying the optimized voltage and frequency may be performed for each of the two or more GPUs.

[0016] An apparatus for GPU power management in distributed deep learning according to an embodiment includes memory in which at least one program is recorded and a processor for executing the program. The program may perform identifying at least one communication section during the training process or inference process of the distributed deep learning in two or more GPUs and performing control such that the voltage and frequency of the GPU are optimized during the at least one communication section.

[0017] Here, when performing the control, the program may set the voltage and frequency of the GPU during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

[0018] Here, when performing the control, the program may perform inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

[0019] Here, when performing the control, the program may perform checking whether the voltage and frequency of the GPU are minimum values; and setting the voltage and frequency as the optimal voltage and frequency of the GPU when the voltage and frequency are the minimum values.

[0020] Here, when performing the control, the program may perform checking whether the voltage and frequency of the GPU are minimum values, decreasing the voltage and frequency of the GPU by a predetermined value when the voltage and frequency are not the minimum values, monitoring performance degradation after performing distributed training or inference in the GPU, and setting a previous voltage and frequency as the optimal voltage and frequency of the GPU when performance degradation occurs.

[0021] Here, when performing the control, the program may proceed to decreasing the voltage and frequency of the GPU by the predetermined value when performance degradation does not occur.

[0022] Here, the program may further perform applying the optimized voltage and frequency of the GPU to the communication section while the distributed deep learning is being performed.

[0023] Here, the program may perform identifying the at least one communication section, performing the control, and applying the optimized voltage and frequency for each of the two or more GPUs.

[0024] A method for GPU power management in distributed deep learning, the training process or inference process of which is performed through two or more GPUs, according to an embodiment may include identifying at least one communication section during the training process or inference process of the distributed deep learning and setting the voltage and frequency of each of the two or more GPUs during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

[0025] Here, setting the voltage and frequency may include inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

[0026] Here, setting the voltage and frequency may include checking whether the voltage and frequency of the GPU are the minimum values; and setting the voltage and frequency as the optimal voltage and frequency of the GPU when the voltage and frequency are the minimum values.

[0027] Here, setting the voltage and frequency may include checking whether the voltage and frequency of the GPU are the minimum values, decreasing the voltage and frequency of the GPU by a predetermined value when the voltage and frequency are not the minimum values, monitoring performance degradation after performing distributed training or inference in the GPU, and setting a previous voltage and frequency as the optimal voltage and frequency of the GPU when performance degradation occurs. When performance degradation does not occur, setting the voltage and frequency may procced to decreasing the voltage and frequency of the GPU by the predetermined value.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] The above and other objects, features, and advantages of the present disclosure will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings, in which:

[0029] FIG. **1** is a schematic block diagram of a distributed deep-learning system to which an embodiment is applied;

[0030] FIG. **2** is a schematic internal configuration diagram of an apparatus for GPU power management in distributed deep learning according to an embodiment;

[0031] FIG. **3** is a flowchart for explaining a method for GPU power management in distributed deep learning according to an embodiment;

[0032] FIG. **4** is a flowchart for explaining a step of performing control such that the voltage and frequency of a GPU are optimized according to an embodiment; and

[0033] FIG. **5** is a view illustrating a computer system configuration according to an embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0034] The advantages and features of the present disclosure and methods of achieving them will be apparent from the following exemplary embodiments to be described in more detail with reference to the accompanying drawings. However, it should be noted that the present disclosure is not limited to the following exemplary embodiments, and may be implemented in various forms. Accordingly, the exemplary embodiments are provided only to disclose the present disclosure and to let those skilled in the art know the category of the present disclosure, and the present disclosure is to be defined based only on the claims. The same reference numerals or the same reference designators denote the same elements throughout the specification.

[0035] It will be understood that, although the terms “first,” “second,” etc. may be used herein to describe various elements, these elements are not intended to be limited by these terms. These terms are only used to distinguish one element from another element. For example, a first element discussed below could be referred to as a second element without departing from the technical spirit of the present disclosure.

[0036] The terms used herein are for the purpose of describing particular embodiments only and are not intended to limit the present disclosure. As used herein, the singular forms are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises,” “comprising,”, “includes” and/or “including,” when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0037] Unless differently defined, all terms used herein, including technical or scientific terms, have the same meanings as terms generally understood by those skilled in the art to which the present disclosure pertains. Terms identical to those defined in generally used dictionaries should be interpreted as having meanings identical to contextual meanings of the related art, and are not to be interpreted as having ideal or excessively formal meanings unless they are definitively defined in the present specification.

[0038] FIG. **1** is a schematic block diagram of a distributed deep-learning system to which an embodiment is applied.

[0039] Referring to FIG. **1**, the distributed deep-learning system according to an embodiment may include multiple GPUs **20-1**, **20-2**, . . . , **20**-N for performing distributed deep learning by being assigned a deep-learning model **10** and an apparatus **100** for GPU power management in distributed deep learning, which manages distributed deep learning and power of the multiple GPUs **20-1**, **20-2**, . . . , **20**-N.

[0040] Here, the deep-learning model **10** is assigned to the multiple GPUs **20-1**, **20-2**, . . . , **20**-N using at least one of data parallelism, or tensor parallelism, or a combination thereof, whereby a distributed deep-learning process may be performed.

[0041] The apparatus **100** for GPU power management in distributed deep learning according to an embodiment performs management such that the power of each of the multiple GPUs **20-1**, **20-2**, . . . , **20**-N is optimized while the training process or inference process of the deep-learning model **10** is being performed through the multiple GPUs **20-1**, **20-2**, . . . , **20**-N.

[0042] FIG. **2** is a schematic internal configuration diagram of an apparatus for GPU power management in distributed deep learning according to an embodiment.

[0043] Referring to FIG. **2**, the apparatus **100** for GPU power management in distributed deep learning (referred to as the 'management apparatus' hereinbelow) according to an embodiment may include a communication profiling unit **110**, a power management control unit **120**, and a performance monitoring unit **130**.

[0044] The management apparatus **100** according to an embodiment may further include a distributed deep-learning control unit **140**.

[0045] The communication profiling unit **110** identifies at least one communication section occurring during training and inference of distributed deep learning.

[0046] The power management control unit **120** applies power management technology in the identified at least one communication section. That is, the power management control unit **120** performs control such that the voltage and frequency of a GPU are optimized during the at least one communication section.

[0047] Here, the power management control unit **120** may set the voltage and frequency of the GPU during the at least one communication section to minimum values at which the performance of the training process or inference process of distributed deep learning is not degraded.

[0048] Here, the power management control unit **120** may insert voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

[0049] Here, the power management control unit **120** may check whether the voltage and frequency of the GPU are minimum values, and when the corresponding values are the minimum values, the power management control unit **120** may set the corresponding values as the optimal voltage and frequency of the GPU.

[0050] Here, when the voltage and frequency of the GPU are not the minimum values, the power management control unit **120** may decrease the voltage and frequency of the GPU by a predetermined value.

[0051] Here, the voltage and frequency of the GPU may be adjusted to decrease by the predetermined value in a stepwise manner.

[0052] In other words, when distributed training or inference is performed in the GPU in the state in which the voltage and frequency of the GPU are decreased, if performance is not degraded, the power management control unit **120** may repeatedly decrease the voltage and frequency of the GPU in a stepwise manner.

[0053] However, when performance degradation occurs in a performance degradation monitoring result, the power management control unit **120** may set the previous voltage and frequency as the optimal voltage and frequency of the GPU.

[0054] Here, performance degradation monitoring may be performed by the performance monitoring unit **130**.

[0055] Meanwhile, the distributed deep-learning control unit **140** may apply the determined power management state to the communication section while distributed deep learning is being performed.

[0056] FIG. **3** is a flowchart for explaining a method for GPU power management in distributed deep learning according to an embodiment.

[0057] Referring to FIG. **3**, in the method for GPU power management in distributed deep learning according to an embodiment, the training process or inference process of distributed deep learning is performed through two or more Graphics Processing Units (GPUs), and the method may include identifying at least one communication section during the training process or inference process of distributed deep learning at step S**210** and performing control such that the voltage and frequency of the GPU are optimized during the at least one communication section at step S**220**.

[0058] The method for GPU power management in distributed deep learning according to an embodiment may further include applying the optimized voltage and frequency of the GPU to the communication section while distributed deep learning is being performed.

[0059] Here, steps from S**210** to S**230** may be performed for each of the two or more GPUs.

[0060] FIG. **4** is a flowchart for explaining a step of performing control such that the voltage and frequency of a GPU are optimized according to an embodiment.

[0061] Referring to FIG. **4**, performing the control at step S**220** may comprise setting the voltage and frequency of a GPU during at least one communication section to minimum values at which the performance of the training process or inference process of distributed deep learning is not degraded.

[0062] Here, performing the control at step S**220** may include inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof at step S**221**.

[0063] That is, the voltage and frequency control code is inserted before and after the communication section, whereby a low frequency may be used during the communication section.

[0064] Here, performing the control at step S**220** may include checking whether the voltage and frequency of the GPU are the minimum values at step S**222** and setting, when the voltage and frequency of the GPU are the minimum values, the corresponding values as the optimal voltage and frequency of the GPU at step S**227**.

[0065] That is, whether the current voltage and frequency of the GPU are the minimum values is checked in order to set the optimal voltage and frequency in terms of energy, and when the current voltage and frequency are the minimum values, the corresponding values are consistently used.

[0066] Here, performing the control at step S**220** may include checking whether the voltage and frequency of the GPU are the minimum values at step S**222**, decreasing the voltage and frequency of the GPU by a predetermined value at step S**223** when the voltage and frequency are not the minimum values, monitoring performance degradation at step S**225** after performing distributed training or inference in the GPU at step S**224**, and setting a previous voltage and frequency as the optimal voltage and frequency of the GPU at step S**226** when performance degradation occurs. That is, in order to prevent performance degradation, the previous voltage and frequency values are set, and the corresponding values are consistently used at step S**227**.

[0067] Here, performing the control at step S**220** may proceed to checking whether the voltage and frequency of the GPU are the minimum values at step S**222** when performance degradation does not occur at step S**225**.

[0068] In other words, if performance degradation does not occur, it is determined that energy efficiency can be more improved, so the process of performing control to further decrease the voltage and frequency is repeated, whereby the minimum voltage and frequency at which performance degradation does not occur are searched for and applied.

[0069] Here, the voltage and frequency of the GPU may be adjusted to decrease by a predetermined value in a stepwise manner.

[0070] In other words, the power management control unit **120** may repeatedly decrease the voltage and frequency of the GPU in a stepwise manner when performance is not degraded in a result of distributed training or inference performed in the GPU in the state in which the voltage and frequency of the GPU are decreased.

[0071] According to the embodiment described above, the energy consumption by GPUs may be significantly reduced without performance degradation by utilizing GPU power management during the communication time that occurs during training or inference of distributed deep learning.

[0072] FIG. **5** is a view illustrating a computer system configuration according to an embodiment.

[0073] The apparatus **100** for GPU power management in distributed deep learning according to an embodiment may be implemented in a computer system **1000** including a computer-readable recording medium.

[0074] The computer system **1000** may include one or more processors **1010**, memory **1030**, a user-interface input device **1040**, a user-interface output device **1050**, and storage **1060**, which communicate with each other via a bus **1020**. Also, the computer system **1000** may further include a network interface **1070** connected with a network **1080**. The processor **1010** may be a central processing unit or a semiconductor device for executing a program or processing instructions stored in the memory **1030** or the storage **1060**. The memory **1030** and the storage **1060** may be storage media including at least one of a volatile medium, a non-volatile medium, a detachable medium, a non-detachable medium, a communication medium, or an information delivery medium, or a combination thereof. For example, the memory **1030** may include ROM **1031** or RAM **1032**.

[0075] According to the disclosed embodiment, energy inefficiency caused in the distributed training or inference process of deep learning may be mitigated.

[0076] The disclosed embodiment may be easily applied not only to distributed deep learning based on pipeline parallelism but also to distributed deep learning based on data parallelism and tensor parallelism.

[0077] Although embodiments of the present disclosure have been described with reference to the accompanying drawings, those skilled in the art will appreciate that the present disclosure may be practiced in other specific forms without changing the technical spirit or essential features of the present disclosure. Therefore, the embodiments described above are illustrative in all aspects and should not be understood as limiting the present disclosure.

What is claimed is:

1. A method for Graphics Processing Unit (GPU) power management in distributed deep learning, a training process or inference process of which is performed through two or more GPUs, the method comprising:

identifying at least one communication section during the training process or inference process of the distributed deep learning; and

performing control such that a voltage and frequency of the GPU are optimized during the at least one communication section.

2. The method of claim **1**, wherein performing the control comprises setting the voltage and frequency of the GPU during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

3. The method of claim **1**, wherein performing the control includes inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

4. The method of claim **1**, wherein performing the control includes

checking whether the voltage and frequency of the GPU are minimum values; and

when the voltage and frequency are the minimum values, setting the voltage and frequency as an optimal voltage and frequency of the GPU.

5. The method of claim **1**, wherein performing the control includes

checking whether the voltage and frequency of the GPU are minimum values;

when the voltage and frequency are not the minimum values, decreasing the voltage and frequency of the GPU by a predetermined value;

monitoring performance degradation after performing distributed training or inference in the GPU; and

when performance degradation occurs, setting a previous voltage and frequency as an optimal voltage and frequency of the GPU.

6. The method of claim **5**, wherein performing the control proceeds to decreasing the voltage and frequency of the GPU by the predetermined value when performance degradation does not occur.

7. The method of claim **1**, further comprising:

applying an optimized voltage and frequency of the GPU to the communication section while the distributed deep learning is being performed.

8. The method of claim **7**, wherein identifying the at least one communication section, performing the control, and applying the optimized voltage and frequency are performed for each of the two or more GPUs.

9. An apparatus for Graphics Processing Unit (GPU) power management in distributed deep learning, comprising:

memory in which at least one program is recorded; and

a processor for executing the program,

wherein the program performs

identifying at least one communication section during a training process or inference process of the distributed deep learning in two or more GPUs and

performing control such that a voltage and frequency of the GPU are optimized during the at least one communication section.

10. The apparatus of claim **9**, wherein, when performing the control, the program sets the voltage and frequency of the GPU during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

11. The apparatus of claim **9**, wherein, when performing the control, the program performs inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

12. The apparatus of claim **9**, wherein, when performing the control, the program performs

checking whether the voltage and frequency of the GPU are minimum values; and

when the voltage and frequency are the minimum values, setting the voltage and frequency as an optimal voltage and frequency of the GPU.

13. The apparatus of claim **9**, wherein, when performing the control, the program performs

checking whether the voltage and frequency of the GPU are minimum values;

decreasing the voltage and frequency of the GPU by a predetermined value when the voltage and frequency are not the minimum values;

monitoring performance degradation after performing distributed training or inference in the GPU; and

setting a previous voltage and frequency as an optimal voltage and frequency of the GPU when performance degradation occurs.

14. The apparatus of claim **13**, wherein, when performing the control, the program proceeds to decreasing the voltage and frequency of the GPU by the predetermined value when performance degradation does not occur.

15. The apparatus of claim **9**, wherein the program further performs applying an optimized voltage and frequency of the GPU to the communication section while the distributed deep learning is being performed.

16. The apparatus of claim **15**, wherein the program performs identifying the at least one communication section, performing the control, and applying the optimized voltage and frequency for each of the two or more GPUs.

17. A method for Graphics Processing Unit (GPU) power management in distributed deep learning, a training process or inference process of which is performed through two or more GPUs, the method comprising:

identifying at least one communication section during the training process or inference process of the distributed deep learning; and

setting a voltage and frequency of each of the two or more GPUs during the at least one communication section to minimum values at which performance of the training process or inference process of the distributed deep learning is not degraded.

18. The method of claim **17**, wherein setting the voltage and frequency includes inserting voltage and frequency control code into the GPU at at least one of a time point before the at least one communication section, or a time point after the at least one communication section, or a combination thereof.

19. The method of claim **17**, wherein setting the voltage and frequency includes

checking whether the voltage and frequency of the GPU are the minimum values; and

when the voltage and frequency are the minimum values, setting the voltage and frequency as an optimal voltage and frequency of the GPU.

20. The method of claim **17**, wherein

setting the voltage and frequency includes

checking whether the voltage and frequency of the GPU are the minimum values,

decreasing the voltage and frequency of the GPU by a predetermined value when the voltage and frequency are not the minimum values,

monitoring performance degradation after performing distributed training or inference in the GPU, and

setting a previous voltage and frequency as an optimal voltage and frequency of the GPU when performance degradation occurs, and

when performance degradation does not occur, setting the voltage and frequency proceeds to decreasing the voltage and frequency of the GPU by the predetermined value.

* * * * *