



(12) 发明专利

(10) 授权公告号 CN 111177892 B

(45) 授权公告日 2023.05.02

(21) 申请号 201911269835.6

(22) 申请日 2019.12.11

(65) 同一申请的已公布的文献号
申请公布号 CN 111177892 A

(43) 申请公布日 2020.05.19

(73) 专利权人 中电普信(北京)科技发展有限公司
地址 100000 北京市海淀区东北旺村南1号楼6层609室

(72) 发明人 郑建国 叶少波 张晓丰 杨建池 马丽娜

(74) 专利代理机构 北京冠和权律师事务所
11399
专利代理师 张楠楠

(51) Int.Cl.
G06F 30/20 (2020.01)

(56) 对比文件
CN 101442555 A,2009.05.27
CN 101930463 A,2010.12.29

CN 102523104 A,2012.06.27

CN 102779199 A,2012.11.14

CN 106557611 A,2017.04.05

CN 107193639 A,2017.09.22

US 10484249 B1,2019.11.19

US 2010262559 A1,2010.10.14

US 2015377005 A1,2015.12.31

US 2018349528 A1,2018.12.06

US 2019093469 A1,2019.03.28

US 6038383 A,2000.03.14

US 8479126 B1,2013.07.02

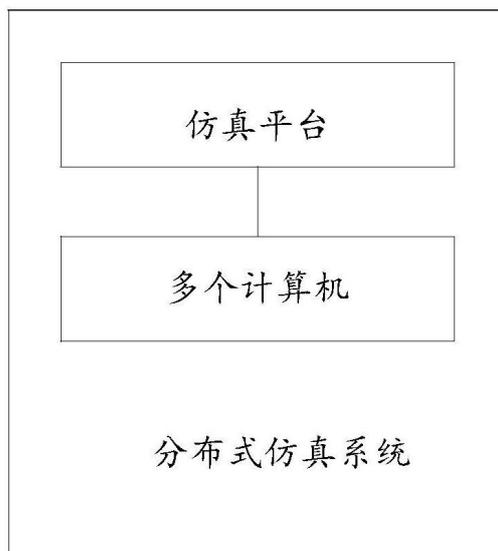
Piggott, MD.A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes.《INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS》.2008,第1-13页.
刘民岷.基于网格计算的分布式仿真关键问题研究.《中国优秀硕士学位论文全文数据库》.2011,第1-5页.

审查员 张曦

权利要求书4页 说明书16页 附图1页

(54) 发明名称
一种分布式仿真系统

(57) 摘要
本发明是关于一种分布式仿真系统。该系统包括：仿真平台，用于对多个实体模型进行仿真；以及用于并行运行所述仿真平台的多个计算机，所述多个计算机中各计算机上安装的仿真平台上分布有所述多个实体模型中的至少一个实体模型，用于并行仿真所述多个实体模型，其中：所述各计算机，用于监控所述各计算机其自身上分布的实体模型的仿真状态，获取其自身的资源使用情况，并根据所述实体模型的仿真状态和/或所述其自身的资源使用情况，将实体模型在计算机之间进行迁移。通过本发明的技术方案，可在分布式仿真过程中实现动态负载均衡，从而显著提高仿真效率。



CN 111177892 B

1. 一种分布式仿真系统,其特征在于,包括:

仿真平台,用于对多个实体模型进行仿真;以及

用于并行运行所述仿真平台的多个计算机,所述多个计算机中各计算机上安装的仿真平台上分布有所述多个实体模型中的至少一个实体模型,用于并行仿真所述多个实体模型,其中:

所述各计算机,用于监控所述各计算机其自身上分布的实体模型的仿真状态,获取其自身的资源使用情况,并根据所述实体模型的仿真状态和/或所述其自身的资源使用情况,将实体模型在计算机之间进行迁移;

仿真管理进程,用于根据所述各计算机的仿真进程发送的所述各计算机其自身的资源使用率,确定所述各计算机中资源使用率最高的第一目标计算机;

所述仿真管理进程在所述第一目标计算机的资源使用率高于第一使用率时,将所述第一目标计算机上的实体模型迁移到所述各计算机中除所述第一目标计算机之外的第二目标计算机;

所述仿真管理进程还用于:

确定所述第一目标计算机上资源使用率排名第 i 的实体模型,并计算所述排名第 i 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率;

若所述第二目标计算机的预计资源使用率大于所述第一目标计算机迁移所述第 i 的实体模型之后的当前资源使用率,则计算所述第一目标计算机上资源使用率排名第 $i+1$ 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率,依次类推,直至所述第二目标计算机的预计资源使用率不大于所述第一目标计算机迁移第 n 实体模型之后的当前资源使用率时,确定将所述第一目标计算机上的第 n 实体模型迁移至所述第二目标计算机,所述第一目标计算机上共有 N 个实体模型, n 大于 $i+1$ 且小于 N ,资源使用率排名为资源使用率从高到低的排名, i 为不小于2的正整数, N 为正整数。

2. 根据权利要求1所述的系统,其特征在于,

所述各计算机上运行有至少一个仿真进程;

所述各计算机上运行的仿真进程,用于监控所述各计算机上分布的实体模型的仿真状态以及所述各计算机其自身的资源使用情况;

所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

所述各计算机中的仿真进程,用于监控所述各计算机其自身上分布的实体模型的仿真状态,并将监控的实体模型的仿真状态发送至所述仿真管理进程;

所述仿真管理进程,用于接收所述各计算机的仿真进程监控的实体模型的仿真状态,并进行记录。

3. 根据权利要求2所述的系统,其特征在于,

所述各计算机的仿真进程,还用于确定所述各计算机其自身上分布的所有实体模型分别对应的仿真步长中的当前最小步长,并将所述当前最小步长发送至所述仿真管理进程;

所述仿真管理进程,还用于根据所述各计算机的仿真进程发送的当前最小步长确定下一次仿真所述所有实体模型的最小时间步长,并将所述最小时间步长发送至所述各计算机。

4. 根据权利要求2所述的系统,其特征在于,

所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

所述各计算机的仿真进程,用于将所述各计算机其自身的资源使用情况发送至所述仿真管理进程;

所述仿真管理进程,用于接收所述各计算机的仿真进程发送的其自身的资源使用情况,并根据所述其自身的资源使用情况确定所述各计算机上分布的实体模型的迁移方向,所述资源使用情况包括资源使用率,所述资源包括CPU和/或内存。

5. 根据权利要求4所述的系统,其特征在于,

所述仿真管理进程,还用于在所述各计算机的资源使用率低于第二使用率时,确定所述多个仿真进程中第P_m个仿真进程上分布的各实体模型对所述多个仿真进程中第P_i个仿真进程上分布的实体模型的访问次数;

所述仿真管理进程基于所述第P_m个仿真进程上分布的各实体模型对所述第P_i个仿真进程上分布的实体模型的访问次数,计算所述第P_m个仿真进程上分布的各实体模型从所述第P_m个仿真进程迁移到对应访问次数最多的第P_{max}个仿真进程上的迁移收益;

所述仿真管理进程进行迁移收益排名,将所述第P_m个仿真进程上迁移收益最大的实体模型迁移至所述第P_{max}个仿真进程上,其中,所述多个仿真进程的编号为P₀~P_n,P_m、P_i、P_{max}为所述P₀~P_n个仿真进程中的仿真进程。

6. 根据权利要求4所述的系统,其特征在于,

所述仿真管理进程基于以下步骤计算将所述多个计算机中一个计算机上的实体模型迁移到另一个计算机上获得的迁移总收益;

设W_i为仿真进程P_i的负载比重,其计算公式如下:

$$W_i = \frac{K_c * U_c(i) + K_m * U_m(i) + K_d * U_d(i) + K_b * U_b(i)}{K_c + K_m + K_d + K_b}$$

其中U_c(i)、U_m(i)、U_d(i)、U_b(i)分别表示单位时间T内仿真进程P_i占用的CPU、内存、磁盘IO和网络带宽占用率,K_c、K_m、K_d、K_b分别表示CPU、内存、磁盘IO和网络带宽对计算机的预设影响因子,且K_c+K_m+K_d+K_b=1;

设e为被迁移进程的计算机编号,f为迁移的目标计算机编号,e=1,2,3,⋯,E,f=1,2,3,⋯,F,M_{ie}表示将仿真进程P_i上分布的实体模型从计算机e迁移到计算机f后计算机e的迁移收益:

$$M_{ie} = \frac{\max \left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}} \right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}}} - \frac{\max \left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}} \right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max \left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}} \right) - e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}} \max \left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}} \right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n-1) \sum_{i=1}^n W_i}{n(n-1)}}}$$

M_{if} 表示将仿真进程 P_i 迁移到计算机 f 后计算机 f 的迁移收益:

$$M_{if} = \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n+1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right) - e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n+1) \sum_{i=1}^n W_i}{n(n+1)}}}$$

则仿真进程 P_i 上的实体模型从计算机 e 迁移到计算机 f 的总收益为 M_i :

$$M_i = M_{ie} + M_{if}$$

根据 M_i 的大小进行排序, M_i 越大,迁移收益越高;

将总收益 M_i 最高的计算机 e 的仿真进程上分布的实体模型迁移至计算机 f 上。

7. 根据权利要求2所述的系统,其特征在於,

所述仿真管理进程还用于:

根据所述各计算机中的仿真进程发送的所述各计算机其自身上分布的实体模型的仿真状态,生成所述各计算机上的实体模型的当前仿真记录;

调用预先获取的实体模型在实际运行环境下的预设状态记录;

根据所述当前仿真记录和所述预设状态记录,生成至少一个实体模型仿真运行指令;

将所述至少一个实体模型仿真运行指令广播至所述各计算机中的仿真进程,以使所述各计算机中的仿真进程将所述至少一个实体模型仿真运行指令发送至所述各计算机其自身上分布的实体模型中对应的实体模型,从而控制所述对应的实体模型的仿真过程或者调节所述对应的实体模型的模型参数。

8. 根据权利要求1所述的系统,其特征在於,所述各计算机还用于:

采用通用建模工具对所述各计算机其自身上分布的实体模型对应的被仿真对象进行数学模型搭建,得到简易仿真模型;

采用专业建模工具对所述各计算机其自身上分布的实体模型对应的被仿真对象进行数学模型搭建,得到精细仿真模型;

以所述精细仿真模型为衡量标准,对所述简易仿真模型的模型结构和模型参数做初步验证,并对不符合第一预设要求的模型结构和模型参数进行修正;

利用所述被仿真对象的第一组实验数据对初步验证修正后的简易仿真模型做再次验证,并对所述修正后的简易仿真模型的模型参数做再次修正,直至在相同的输入下修正后的简易仿真模型的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标满足预设接受值,获得根据所述第一组实验数据得到的简易仿真模型;

利用所述被仿真对象的第二组实验数据作为所述第一组实验数据得到的简易仿真模型的输入,经过所述简易仿真模型的仿真计算,得到所述第二组实验数据对应的输出数据;

计算所述第二组实验数据对应的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标;

判断所述第二组实验数据的一致性衡量指标是否满足所述预设接受值;

如果是,则将所述第二组实验数据得到的简易仿真模型作为所述被仿真对象的实体模型。

一种分布式仿真系统

技术领域

[0001] 本发明涉及仿真技术领域,尤其涉及分布式仿真系统。

背景技术

[0002] 目前,随机计算机仿真技术自诞生,仿真技术受到了世界各国的广泛关注和高度重视,然而现在的仿真性能并不佳,尤其是当仿真模型较多或者实体模型过于复杂、计算量过大时,单台计算机性能有限,仿真速度不能满足要求,因而,如何提高仿真效率以有效提升仿真性能成为亟待解决的问题。

发明内容

[0003] 本发明实施例提供了一种分布式仿真系统。所述技术方案如下:

[0004] 仿真平台,用于对多个实体模型进行仿真;以及

[0005] 用于并行运行所述仿真平台的多个计算机,所述多个计算机中各计算机上安装的仿真平台上分布有所述多个实体模型中的至少一个实体模型,用于并行仿真所述多个实体模型,其中:

[0006] 所述各计算机,用于监控所述各计算机其自身上分布的实体模型的仿真状态,获取其自身的资源使用情况,并根据所述实体模型的仿真状态和/或所述其自身的资源使用情况,将实体模型在计算机之间进行迁移。

[0007] 在一个实施例中,所述各计算机上运行有至少一个仿真进程;

[0008] 所述各计算机上运行的仿真进程,用于监控所述各计算机上分布的实体模型的仿真状态以及所述各计算机其自身的资源使用情况。

[0009] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0010] 所述各计算机中的仿真进程,用于监控所述各计算机其自身上分布的实体模型的仿真状态,并将监控的实体模型的仿真状态发送至所述仿真管理进程;

[0011] 所述仿真管理进程,用于接收所述各计算机的仿真进程监控的实体模型的仿真状态,并进行记录。

[0012] 在一个实施例中,所述各计算机的仿真进程,还用于确定所述各计算机其自身上分布的所有实体模型分别对应的仿真步长中的当前最小步长,并将所述当前最小步长发送至所述仿真管理进程;

[0013] 所述仿真管理进程,还用于根据所述各计算机的仿真进程发送的当前最小步长确定下一次仿真所述所有实体模型的最小时间步长,并将所述最小时间步长发送至所述各计算机。

[0014] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0015] 所述各计算机的仿真进程,用于将所述各计算机其自身的资源使用情况发送至所

述仿真管理进程；

[0016] 所述仿真管理进程，用于接收所述各计算机的仿真进程发送的其自身的资源使用情况，并根据所述其自身的资源使用情况确定所述各计算机上分布的实体模型的迁移方向，所述资源使用情况包括资源使用率，所述资源包括CPU和/或内存。

[0017] 在一个实施例中，所述仿真管理进程，还用于根据所述各计算机的仿真进程发送的所述各计算机其自身的资源使用率，确定所述各计算机中资源使用率最高的第一目标计算机；

[0018] 所述仿真管理进程在所述第一目标计算机的资源使用率高于第一使用率时，将所述第一目标计算机上的实体模型迁移到所述各计算机中除所述第一目标计算机之外的第二目标计算机。

[0019] 在一个实施例中，所述仿真管理进程还用于：

[0020] 确定所述第一目标计算机上资源使用率排名第*i*的实体模型，并计算所述排名第*i*的实体模型迁移至所述第二目标计算机后，所述第二目标计算机的预计资源使用率；

[0021] 若所述第二目标计算机的预计资源使用率大于所述第一目标计算机迁移所述第*i*的实体模型之后的当前资源使用率，则计算所述第一目标计算机上资源使用率排名第*i+1*的实体模型迁移至所述第二目标计算机后，所述第二目标计算机的预计资源使用率，依次类推，直至所述第二目标计算机的预计资源使用率不大于所述第一目标计算机迁移第*n*实体模型之后的当前资源使用率时，确定将所述第一目标计算机上的第*n*实体模型迁移至所述第二目标计算机，所述第一目标计算机上共有*N*个实体模型，*n*大于*i+1*且小于*N*，资源使用率排名为资源使用率从高到低的排名，*i*为不小于2的正整数，*N*为正整数。

[0022] 在一个实施例中，所述仿真管理进程，还用于在所述各计算机的资源使用率低于第二使用率时，确定所述多个仿真进程中第*P_m*个仿真进程上分布的各实体模型对所述多个仿真进程中第*P_i*个仿真进程上分布的实体模型的访问次数；

[0023] 所述仿真管理进程基于所述第*P_m*个仿真进程上分布的各实体模型对所述第*P_i*个仿真进程上分布的实体模型的访问次数，计算所述第*P_m*个仿真进程上分布的各实体模型从所述第*P_m*个仿真进程迁移到对应访问次数最多的第*P_{max}*个仿真进程上的迁移收益；

[0024] 所述仿真管理进程进行迁移收益排名，将所述第*P_m*个仿真进程上迁移收益最大的实体模型迁移至所述第*P_{max}*个仿真进程上，其中，所述多个仿真进程的编号为*P₀~P_n*，*P_m*、*P_i*、*P_{max}*为所述*P₀~P_n*个仿真进程中的仿真进程。

[0025] 在一个实施例中，所述仿真管理进程基于以下步骤计算将所述多个计算机中一个计算机上的实体模型迁移到另一个计算机上获得的迁移总收益；

[0026] 设*W_i*为仿真进程*P_i*的负载比重，其计算公式如下：

$$[0027] \quad W_i = \frac{K_c * U_c(i) + K_m * U_m(i) + K_d * U_d(i) + K_b * U_b(i)}{K_c + K_m + K_d + K_b}$$

[0028] 其中*U_c(i)*、*U_m(i)*、*U_d(i)*、*U_b(i)*分别表示单位时间*T*内仿真进程*P_i*占用的CPU、内存、磁盘IO和网络带宽占用率，*K_c*、*K_m*、*K_d*、*K_b*分别表示CPU、内存、磁盘IO和网络带宽对计算机的预设影响因子，且*K_c+K_m+K_d+K_b=1*；

[0029] 设*e*为被迁移进程的计算机编号，*f*为迁移的目标计算机编号，*e=1,2,3,⋯,E*，*f=1,2,3,⋯,F*，*M_{ie}*表示将仿真进程*P_i*上分布的实体模型从计算机*e*迁移到计算机*f*后计算机*e*

的迁移收益：

$$\begin{aligned}
 M_{ie} &= \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}} \\
 [0030] \quad &= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right) - e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n-1) \sum_{i=1}^n W_i}{n(n-1)}}}
 \end{aligned}$$

[0031] M_{if} 表示将仿真进程 P_i 迁移到计算机 f 后计算机 f 的迁移收益：

$$\begin{aligned}
 M_{if} &= \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n+1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}} \\
 [0032] \quad &= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right) - e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n+1) \sum_{i=1}^n W_i}{n(n+1)}}}
 \end{aligned}$$

[0033] 则仿真进程 P_i 上的实体模型从计算机 e 迁移到计算机 f 的总收益为 M_i ：

$$[0034] \quad M_i = M_{ie} + M_{if}$$

[0035] 根据 M_i 的大小进行排序, M_i 越大,迁移收益越高;

[0036] 将总收益 M_i 最高的计算机 e 的仿真进程上分布的实体模型迁移至计算机 f 上。在一个实施例中,所述仿真管理进程还用于:

[0037] 根据所述各计算机中的仿真进程发送的所述各计算机其自身上分布的实体模型的仿真状态,生成所述各计算机上的实体模型的当前仿真记录;

[0038] 调用预先获取的实体模型在实际运行环境下的预设状态记录;

[0039] 根据所述当前仿真记录和所述预设状态记录,生成至少一个实体模型仿真运行指令;

[0040] 将所述至少一个实体模型仿真运行指令广播至所述各计算机中的仿真进程,以使所述各计算机中的仿真进程将所述至少一个实体模型仿真运行指令发送至所述各计算机其自身上分布的实体模型中对应的实体模型,从而控制所述对应的实体模型的仿真过程或者调节所述对应的实体模型的模型参数。

[0041] 在一个实施例中,所述各计算机还用于:

[0042] 采用通用建模工具对所述各计算机其自身上分布的实体模型(对应)的被仿真对象进行数学模型搭建,得到简易仿真模型;

[0043] 采用专业建模工具对所述各计算机其自身上分布的实体模型(对应)的被仿真对象进行数学模型搭建,得到精细仿真模型;

[0044] 以所述精细仿真模型为衡量标准,对所述简易仿真模型的模型结构和模型参数做

初步验证,并对不符合第一预设要求的模型结构和模型参数进行修正;

[0045] 利用所述被仿真对象的第一组实验数据对初步验证修正后的简易仿真模型做再次验证,并对所述修正后的简易仿真模型的模型参数做再次修正,直至相同的输入下修正后的简易仿真模型的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标满足预设接受值,获得根据所述第一组实验数据得到的简易仿真模型;

[0046] 利用所述被仿真对象的第二组实验数据作为所述第一组实验数据得到的简易仿真模型的输入,经过所述简易仿真模型的仿真计算,得到所述第二组实验数据对应的输出数据;

[0047] 计算所述第二组实验数据对应的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标;

[0048] 判断所述第二组实验数据的一致性衡量指标是否满足所述预设接受值;

[0049] 如果是,则将所述第二组实验数据得到的简易仿真模型作为所述被仿真对象的实体模型。

[0050] 本发明的实施例提供的技术方案可以包括以下有益效果:

[0051] 各计算机上均安装有一个仿真平台,通过使用多个计算机并行运行仿真平台,并将仿真平台中所有实体模型分布在不同的计算机上可利用并行仿真方式加快仿真进度,提高仿真效率,具体地,各计算机可分别监控所述各计算机其自身上分布的实体模型的仿真状态,并获取各计算机自身的资源使用情况,然后根据实体模型的仿真状态和/或所述其自身的资源使用情况,将实体模型在计算机之间进行迁移,以在分布式仿真过程中实现动态负载均衡,从而显著提高仿真效率。

[0052] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本发明。

附图说明

[0053] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本发明的实施例,并与说明书一起用于解释本发明的原理。

[0054] 图1是根据一示例性实施例示出的一种分布式仿真系统的框图。

具体实施方式

[0055] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本发明相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本发明的一些方面相一致的装置和方法的例子。

[0056] 为了解决上述技术问题,本发明实施例提供了一种分布式仿真系统如图1所示,该系统包括:

[0057] 仿真平台,所述仿真平台用于对多个实体模型进行仿真;以及

[0058] 用于并行运行所述仿真平台的多个计算机,所述多个计算机中各计算机上安装的仿真平台上分布有所述多个实体模型中的至少一个实体模型,用于并行仿真所述多个实体模型,其中:

[0059] 实体模型可以是实物在仿真平台中的虚拟对象表示,如飞机模型、卫星模型、车辆模型等。仿真平台可以是华如XSIM、神州普惠DWK等。

[0060] 所述各计算机,用于监控所述各计算机其自身上分布的实体模型的仿真状态,获取其自身的资源使用情况,并根据所述实体模型的仿真状态和/或所述其自身的资源使用情况,将实体模型在计算机之间进行迁移。

[0061] 各计算机上均安装有一个仿真平台,通过使用多个计算机并行运行仿真平台,并将仿真平台中所有实体模型分布在不同的计算机上(即各计算机分别负责所有需要仿真的实体模型中的一部分实体模型,也即上述多个实体仿真模型中的一部分实体模型)可利用并行仿真方式加快仿真进度,提高仿真效率,具体地,各计算机可分别监控所述各计算机其自身上分布的实体模型的仿真状态,并获取各计算机自身的资源使用情况,然后根据实体模型的仿真状态和/或所述其自身的资源使用情况,将实体模型在计算机之间进行迁移,以在分布式仿真过程中实现动态负载均衡,从而显著提高仿真效率。

[0062] 在一个实施例中,所述各计算机上运行有至少一个仿真进程;各计算机上可分别运行一个或多个仿真进程,专门用于负责该计算机自身上分布的若干个实体模型的仿真过程。

[0063] 所述各计算机上运行的仿真进程,用于监控所述各计算机上分布的实体模型的仿真状态以及所述各计算机其自身的资源使用情况。

[0064] 仿真状态可以是实体模型在某时刻的状态,如车辆模型在某时刻的位置、行驶方向等,飞机模型在某时刻的飞行姿态等。

[0065] 资源使用情况可以是CPU(central processing unit,中央处理器)、内存、硬盘等硬件或软件资源的当前使用率、剩余使用率等。

[0066] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0067] 所述各计算机中的仿真进程,用于监控所述各计算机其自身上分布的实体模型的仿真状态,并将监控的实体模型的仿真状态发送至所述仿真管理进程;其中,主控计算机自身的仿真进程也会采集主控计算机(安装的仿真平台上)分布的实体模型的仿真状态。

[0068] 所述仿真管理进程,用于接收所述各计算机的仿真进程监控的实体模型的仿真状态,并进行记录。

[0069] 仿真管理进程通过接收各计算机的仿真进程监控的实体模型的仿真状态,可对各实体模型的仿真状态进行详细统计,以便之后进行仿真评估,从而有利于准确、有效地进行仿真实验。

[0070] 在一个实施例中,所述各计算机的仿真进程,还用于确定所述各计算机其自身上分布的所有实体模型分别对应的仿真步长中的当前最小步长,并将所述当前最小步长发送至所述仿真管理进程;

[0071] 例如:各计算机的仿真进程的编号分别为 $P_0 \sim P_n$ 时,任意管理进程 P_m 上分布的所有实体模型为 $B_1 \sim B_p$,实体模型 $B_1 \sim B_p$ 分别对应的仿真步长为 $T^0 \sim T^p$,则若仿真步长 $T^0 \sim T^p$ 中仿真步长最小的值为 T^m ,则 T^m 为管理进程 P_m 的当前最小步长。

[0072] 计算机内部的仿真过程实际上是离散进行的并不是连续进行的,具体地是按照(仿真)步长进行的,且每个实体模型根据自身的仿真需求对应的仿真步长很可能不同,如

飞机模型的仿真步长与卫星模型的仿真步长就不同。

[0073] 所述仿真管理进程,还用于根据所述各计算机的仿真进程发送的当前最小步长确定下一次仿真所述所有实体模型的最小时间步长,并将所述最小时间步长发送至所述各计算机。

[0074] 为了便于进行仿真管理使所有实体模型可使用相同的仿真步长,各计算机的仿真进程需要确定各计算机其自身上分布的所有实体模型各自对应的仿真步长中的当前最小步长,然后各计算机将当前最小步长发送至仿真管理进程,从而使得仿真管理进程可利用根据所述各计算机的仿真进程发送的当前最小步长准确确定下一次仿真所述所有实体模型的最小时间步长,并将所述最小时间步长发送至所述各计算机,从而使得各计算机上的实体模型按照该最小时间步长进行仿真操作,这样,既可以综合考虑所有实体模型的实际仿真时间要求,又可以准确确定适合所有实体模型的仿真时长。例如:各计算机的仿真进程的编号分别为 $P_0 \sim P_n$ 时,若各计算机的访问进程返回的当前最小步长分别为 $t_0 \sim t_n$,且 $t_0 \sim t_n$ 中的最小值为 t_m ,则仿真管理进程可将 t_m 作为该最小时间步长。

[0075] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0076] 所述各计算机的仿真进程,用于将所述各计算机其自身的资源使用情况发送至所述仿真管理进程;

[0077] 所述仿真管理进程,用于接收所述各计算机的仿真进程发送的其自身的资源使用情况,并根据所述其自身的资源使用情况确定所述各计算机上分布的实体模型的迁移方向,所述资源使用情况包括资源使用率,所述资源包括CPU和/或内存,某计算机其自身的资源使用情况可以为该计算机的资源使用率。

[0078] 仿真管理进程通过接收各计算机的仿真进程发送的其自身的资源使用情况,可确定各计算机的动态负载情况,然后根据各计算机其自身的资源使用情况即可自动确定所述各计算机上(安装的仿真平台上)分布的实体模型的迁移方向,从而将负载负担重的计算机上的实体模型迁移到负载负担轻的计算机上,这样可在分布式并行仿真的基础上,通过负载动态均衡实现负载的平衡,从而有利于进一步缩短仿真时长、提高仿真效率。

[0079] 在一个实施例中,所述仿真管理进程,还用于根据所述各计算机的仿真进程发送的所述各计算机其自身的资源使用率,确定所述各计算机中资源使用率最高的第一目标计算机;

[0080] 所述仿真管理进程在所述第一目标计算机的资源使用率高于第一使用率时,将所述第一目标计算机上的实体模型迁移到所述各计算机中除所述第一目标计算机之外的第二目标计算机。

[0081] 仿真管理进程在所述第一目标计算机的资源使用率高于第一使用率(可以是80%或者90%等)时,可确认第一目标计算机的负载太重,因而,可将第一目标计算机上的实体模型迁移到所述各计算机中除所述第一目标计算机之外的第二目标计算机,即将负载特别高的计算机的若干实体向负载特别低的计算机迁移,从而实现负载的动态平衡,进而有利于进一步缩短仿真时长、提高仿真效率。

[0082] 在一个实施例中,所述仿真管理进程还用于:

[0083] 确定所述第一目标计算机上资源使用率排名第 i 的实体模型,并计算所述排名第 i

的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率;

[0084] 资源使用率排名为资源使用率从高到低的排名,通过确定第一目标计算机上所有实体模型各自所占用的资源使用率,即可进行资源使用率从高到低的排名。

[0085] 若所述第二目标计算机的预计资源使用率大于所述第一目标计算机迁移所述第 i 的实体模型之后的当前资源使用率,则计算所述第一目标计算机上资源使用率排名第 $i+1$ 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率,依次类推,直至所述第二目标计算机的预计资源使用率不大于所述第一目标计算机迁移第 n 实体模型之后的当前资源使用率时,确定将所述第一目标计算机上的第 n 实体模型迁移至所述第二目标计算机,所述第一目标计算机上共有 N 个实体模型, n 大于 $i+1$ 且小于 N , i 为不小于2的正整数, N 为正整数。

[0086] 在进行实体迁移时,可计算将资源使用率排名第 i 的实体模型迁移至第二计算机后,第二目标计算机的预计资源使用率,进而若第二目标计算机的预计资源使用率的预计资源使用率大于所述第一目标计算机迁移所述第 i 的实体模型之后的当前资源使用率,则说明第一目标计算机上资源使用率排名第 i 的实体模型若迁移至第二目标计算机,第二目标计算机的负担会比较沉,不利于提高仿真效率,因而,可考虑第一目标计算机上资源使用率排名第 $i+1$ 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率,依次类推直至若第一目标计算机上的第 n 实体模型迁移至第二目标计算机后第二目标计算机的预计资源使用率不大于所述第一目标计算机迁移走该第 n 实体模型之后的当前资源使用率时,确定将所述第一目标计算机上的第 n 实体模型迁移至所述第二目标计算机,这样既减轻了第一目标计算机的负担,又避免给第二目标计算机增加过多负载负担,从而尽可能实现负载的合理均衡,有利于最大可能地提高仿真效率。

[0087] 另外,由于将资源使用率排名第一的实体模型所占用的资源使用率本来就最高,因而,通过考虑将资源使用率排名第二以及靠后的实体模型进行迁移,而不是将资源使用率排名第一的实体模型进行迁移,可避免直接给第二目标计算机造成较大负载负担。

[0088] 在一个实施例中,所述仿真管理进程,还用于在所述各计算机的资源使用率低于第二使用率(如可以是30%、50%等)时,确定所述多个仿真进程中第 P_m 个仿真进程上分布的各实体模型对所述多个仿真进程中第 P_i 个仿真进程上分布的实体模型的访问次数;

[0089] 所述仿真管理进程基于所述第 P_m 个仿真进程上分布的各实体模型对所述第 P_i 个仿真进程上分布的实体模型的访问次数,计算所述第 P_m 个仿真进程上分布的各实体模型从所述第 P_m 个仿真进程迁移到对应访问次数最多的第 P_{max} 个仿真进程上的迁移收益;

[0090] 所述仿真管理进程进行迁移收益排名,将所述第 P_m 个仿真进程上迁移收益最大的实体模型迁移至所述第 P_{max} 个仿真进程上,其中,所述多个仿真进程的编号为 $P_0 \sim P_n$, P_m 、 P_i 、 P_{max} 为所述 $P_0 \sim P_n$ 个仿真进程中的仿真进程。

[0091] 在各计算机的资源使用率均低于第二使用率时,说明实体模型跨计算机互相访问的次数太多,导致各计算机都在等待,则需要减少跨计算机的实体模型之间的交互,因而,通过计算所述第 P_m 个仿真进程上分布的各实体模型从所述第 P_m 个仿真进程迁移到对应访问次数最多的第 P_{max} 个仿真进程上的迁移收益并进行迁移收益排名,可自动地将所述第 P_m 个仿真进程上迁移收益最大的实体模型迁移至所述第 P_{max} 个仿真进程上,从而可将任意仿真进程上分布的访问某其他仿真进程上的实体模型次数最多的实体模型迁移至该其他仿

真进程上,如此可有效减少跨计算机的实体模型之间的交互,降低仿真交互成本。

[0092] 例如:如果所有的CPU使用率都比较低,说明跨计算机访问太多,导致各计算机都在等待,则需要减少跨计算机的实体模型之间的交互。设共有进程P0到Pn,某实体模型B位于进程Pm,访问进程Pi中的实体模型的次数为Ai,则 $A_{max} = \max(A_0, \dots, A_n)$,其中不包含Am。此时将此实体模型B从Pm迁移到Pmax (Pmax为Amax对应的进程)的迁移收益为 $A_{max} - A_m$,进而将进程Pm上的所有实体的迁移收益排序,然后选择Pm上迁移收益最大的一个实体模型迁移至Pmax上。

[0093] 在一个实施例中,所述仿真管理进程基于以下步骤计算将所述多个计算机中一个计算机上的实体模型迁移到另一个计算机上获得的迁移总收益;

[0094] 设Wi为仿真进程Pi的负载比重,其计算公式如下:

$$[0095] \quad W_i = \frac{K_c * U_c(i) + K_m * U_m(i) + K_d * U_d(i) + K_b * U_b(i)}{K_c + K_m + K_d + K_b}$$

[0096] 其中 $U_c(i)$ 、 $U_m(i)$ 、 $U_d(i)$ 、 $U_b(i)$ 分别表示单位时间T内仿真进程Pi占用的CPU、内存、磁盘IO和网络带宽占用率, K_c 、 K_m 、 K_d 、 K_b 分别表示CPU、内存、磁盘IO和网络带宽对计算机的预设影响因子,且 $K_c + K_m + K_d + K_b = 1$;

[0097] 设e为被迁移进程的计算机编号,f为迁移的目标计算机编号, $e = 1, 2, 3, \dots, E$, $f = 1, 2, 3, \dots, F$, M_{ie} 表示将仿真进程Pi上分布的实体模型从计算机e迁移到计算机f后计算机e的迁移收益:

$$[0098] \quad M_{ie} = \frac{\max\left(e^{W_i \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right) - e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n-1) \sum_{i=1}^n W_i}{n(n-1)}}}$$

[0099] M_{if} 表示将仿真进程Pi迁移到计算机f后计算机f的迁移收益:

$$[0100] \quad M_{if} = \frac{\max\left(e^{W_i \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right)}{e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right) - e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n+1} W_i - (n+1) \sum_{i=1}^n W_i}{n(n+1)}}}$$

[0101] 则仿真进程Pi上的实体模型从计算机e迁移到计算机f的总收益为 M_i :

$$[0102] \quad M_i = M_{ie} + M_{if}$$

[0103] 根据 M_i 的大小进行排序, M_i 越大,迁移收益越高;

[0104] 将总收益 M_i 最高的计算机 e 的仿真进程上分布的实体模型迁移至计算机 f 上。

[0105] 从计算机 e 迁移进程上分布的实体模型到计算机 f ,则计算机 e 的实体模型数会减少,其他模型或进程可占用的资源变多,完成所有进程的时间会相应的缩短,而计算机 f 的模型或进程数会增加,计算机 f 的其他模型或进程可占用的资源变少,完成所有模型仿真的时间会相应增加,综合计算机 ef 所有进程最终完成时间与迁移之前的完成时间的差值即为迁移收益,然后将迁移收益最高的计算机上的实体模型进行迁移,如此可有效减少跨计算机的实体模型之间的交互,降低仿真交互成本。

[0106] 在一个实施例中,所述仿真管理进程还用于:

[0107] 根据所述各计算机中的仿真进程发送的所述各计算机其自身上分布的实体模型的仿真状态,生成所述各计算机上的实体模型的当前仿真记录;

[0108] 调用预先获取的实体模型在实际运行环境下的预设状态记录;

[0109] 根据所述当前仿真记录和所述预设状态记录,生成至少一个实体模型仿真运行指令;

[0110] 将所述至少一个实体模型仿真运行指令广播至所述各计算机中的仿真进程,以使所述各计算机中的仿真进程将所述至少一个实体模型仿真运行指令发送至所述各计算机其自身上分布的实体模型中对应的实体模型,从而控制所述对应的实体模型的仿真过程或者调节所述对应的实体模型的模型参数。

[0111] 通过生成所述各计算机上的实体模型的当前仿真记录并调用预先获取的实体模型在实际运行环境下的预设状态记录,可根据当前仿真记录和所述预设状态记录(如进行记录比较),生成一个或多个实体模型仿真运行指令,然后将这些指令广播至各计算机中的仿真进程,从而使得各计算机中的仿真进程能够将这些指令分发至对应的实体模型,以便于控制该对应的实体模型的仿真过程或者调节所述对应的实体模型的模型参数,以提高仿真效率和准确率,使得仿真效果尽可能贴近仿真模型对应的实物在真实的运行环境运行情况,或者使得实体模型更加准确。

[0112] 在一个实施例中,所述各计算机还用于:

[0113] 采用通用建模工具对所述各计算机其自身上分布的实体模型(对应)的被仿真对象进行数学模型搭建,得到简易仿真模型;

[0114] 通用建模工具的作用为:既能够方便快捷地实现对被仿真对象的原理性建模,又能够支持模型自动代码的生成,从而可以将简易仿真模型下载到通用实时仿真计算机进行实时仿真,通过实时仿真计算机运行解算简易仿真模型,实现与外界相关实物设备的数据交互,发挥简易仿真模型的实际作用。

[0115] 其中,通过对所述各计算机其自身上分布的实体模型行工作原理分析实现对通用建模工具的合理选择,从而简易仿真模型到实时仿真模型的快速过渡问题,使模型的验证过程都集中在对简易仿真模型的验证过程。

[0116] 采用专业建模工具对所述各计算机其自身上分布的实体模型(对应)的被仿真对象进行数学模型搭建,得到精细仿真模型;实体模型(对应)的被仿真对象即实体模型对应的实物。

[0117] 随着数字仿真对仿真模型的精确度要求越来越高,借助各专业建模工具,可以实现较为准确的物理对象建模,例如,利用Solidworks来实现机械系统的建模、利用AMESim来

实现液压系统的建模等。

[0118] 专业建模工具的特点是对被仿真对象进行精细建模,通过有限元计算、高阶微分方程组解算等,实现对被仿真对象的复杂精细计算。专业建模工具能够对被仿真对象进行全面的、细微特性的模拟解算,因此可以选择专业建模工具进行精细仿真模型。

[0119] 以所述精细仿真模型为衡量标准,对所述简易仿真模型的模型结构和模型参数做初步验证,并对不符合第一预设要求的模型结构和模型参数进行修正;

[0120] 第一预设要求为两个数字仿真模型一致的判定准则,可以包括:变化范围一致、变化趋势一致、在线性段斜率一致、若有非线性(如弧线)则拐点位置一致等。

[0121] 利用所述被仿真对象的第一组实验数据对初步验证修正后的简易仿真模型做再次验证,并(通过预设验证方法)对所述修正后的简易仿真模型的模型参数做再次修正,直至相同的输入下修正后的简易仿真模型的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标满足预设接受值,获得根据所述第一组实验数据得到的简易仿真模型;

[0122] 预设验证方法可以是秩和检验法、动态关联分析法、数理统计法等。

[0123] 预设接受值指的是两个数据结果一致程度量化指标,例如若1为完全一致,0为完全不一致,则预设接受值可以为0.75。

[0124] 利用所述被仿真对象的第二组实验数据作为所述第一组实验数据得到的简易仿真模型的输入,经过所述简易仿真模型的仿真计算,得到所述第二组实验数据对应的输出数据;

[0125] 计算所述第二组实验数据对应的输出数据与所述被仿真对象对应的输出数据的一致性衡量指标;

[0126] 判断所述第二组实验数据的一致性衡量指标是否满足所述预设接受值;

[0127] 如果是,则将所述第二组实验数据得到的简易仿真模型作为被仿真对象的实体模型(即最终的实体模型)。

[0128] 通过将模型验证方法和模型参数优化方法集成以实现对数字仿真模型(即简易仿真模型)的自动修正,相比通过建模人员对数字仿真模型修正而言,可减少人工修改模型的工作量,提高了对数字仿真模型的修正效率,以及获得的被仿真对象的实体模型的准确性。

[0129] 下面将进一步详细说明本发明的技术方案:

[0130] 多台计算机中的每台计算机负责仿真平台中的若干个实体模型。分散实体模型的代价主要来自实体模型之间的交互需要通过网络进行。

[0131] 为减少网络交互,可以采取以下策略:

[0132] (1) 把不需要交互的实体模型分离出去。

[0133] (2) 把交互多的实体模型尽量放在同一台机器。

[0134] 为解决计算量过大的问题,可以采取以下策略:

[0135] (1) 把计算量大的实体模型分散到不同的机器上。

[0136] (2) 整个仿真过程中不会受干扰的参数可以提前计算出来,不必按仿真时间进行计算。例如大多数卫星不会变轨,其轨迹可以提前算好。

[0137] (3) 如果有计算特别大的模型,可以根据具体问题的特点,把一个模型的部分计算再分解到多台计算机上。

[0138] 运行开始时按照一定的规则将实体模型分配到多台机器,运行过程中监视每台机

器的CPU使用率和每个实体模型的运行成本和交互成本,必要时将实体模型从一台机器移动到另一台机器。

[0139] 交互成本分成与本机实体模型交互和与其他计算机实体模型交互,将实体模型从机器M1移动到M2,可以减少与M2的交互成本,但是会增加与M1的交互成本。

[0140] 交互成本最高的应该是碰撞检测一类的问题,每一个步长都要查询大量其他实体模型的位置。

[0141] 初始分配规则

[0142] 通常想定只有一个中心,离中心近的实体模型与其他实体模型交互比较多,离中心远的实体模型与其他实体模型交互比较少。因此可以按离中心的距离将实体模型分配到不同的计算机。

[0143] 假设计算机数量为 $\#M$,实体模型数量为 $\#E$ 。从所有实体模型位置的质心开始,按离质心的距离排序,将实体模型队列分成 $\#M$ 份,分别分配到一台计算机上,每台机器负责 $\#E/\#M$ 个实体模型。

[0144] 实体模型迁移规则

[0145] 按CPU使用率迁移

[0146] 如果某计算机CPU使用率高,则需将部分实体模型迁移到CPU使用率低的机器。

[0147] 按对外访问量迁移

[0148] 把访问其他机器过多的实体模型迁移到访问最多的机器上。同时注意避免影响CPU使用率。如果目标机CPU使用率过高,本实体模型的CPU使用率也较高,则考虑迁移下一个实体模型。

[0149] 实体模型迁移的成本

[0150] 必须保证实体模型迁移后能继续正确运行。有可能实体模型的状态比较多,网络传输比较慢,迁移完成之前,仿真时间无法推进,所有机器被迫停下来等待。

[0151] 实体模型分配

[0152] 首先启动的进程ID为0,负责启动其他进程 P_x ($x=1,2,\dots$)。P0和 P_x 都加载完整想定,初始化所有实体模型。P0将实体模型分配到各个进程 P_x ,每个实体模型有一个所属引擎的ID,由此确定当前引擎负责哪些实体模型的计算,去哪个引擎访问其他实体模型。

[0153] 时间推进和时间同步

[0154] 每台计算机上运行一个引擎进程,进程P0负责时间进度,将当前时间和步长发送给所有其他引擎进程 P_x ,所有引擎进程 P_x 完成当前步长的计算之后,回复P0,P0再将下一个时间和步长发送给所有 P_x 。

[0155] 如果实体模型不需要与其他实体模型进行交互,则可以提前进行预计算。

[0156] P0将当前时间T和步长S发送到 P_x , P_x 遍历本进程负责的所有实体模型,如果 $T+S$ 大于实体模型当前时间 T_e 加当前步长 S_e ,则计算此实体模型 T_e+S_e 时刻的状态。计算完成后,遍历本进程负责的所有实体模型,将其中的最小步长返回P0。P0取所有 P_x 返回的最小步长为下一个步长。

[0157] 每个实体模型的当前状态对应的的时间都比引擎时间超前一点点,引擎或其他实体模型访问实体模型的属性时,实体模型必须返回引擎时间(T_e+S_e)的状态。

[0158] 性能监视

[0159] 运算过程中需要监视每台机器的CPU使用率,每个实体模型的CPU使用量,每个实体模型访问其他实体模型的次数。必须优化监视算法,降低监视成本。

[0160] 实体模型迁移

[0161] P_x 将所在机器的CPU使用率发送给 P_0 。 P_0 将CPU使用率平滑以后,确定迁移方向。如果有的CPU使用率特别高,有的CPU使用率特别低,则从特别高的机器选择实体模型向特别低的机器迁移。首先考虑将CPU使用率最高的机器上计算成本第二的实体模型移到CPU使用率最低的机器,如果预计迁移后,目标机器的CPU使用率大于原机器的当前CPU使用率,则考虑排名第三的实体模型,依此类推。

[0162] 如果所有的CPU使用率都比较低,说明跨机器访问太多,导致大家都在等待,则需要减少跨机器的实体模型交互。设共有进程 P_0 到 P_n ,某实体模型位于进程 P_m ,访问进程 P_i 中的实体模型的次数为 A_i ,则 $A_{max} = \max(A_0, \dots, A_n)$,其中不包含 A_m 。此时将此实体模型从 P_m 迁移到 P_{max} (P_{max} 为 P_m 上的所有实体模型各自对应的 A_{max} 中的最大值所对应的被访问进程)的迁移收益为 $A_{max} - A_m$ 。将所有实体模型的迁移收益排序,选择迁移收益最大的一个实体模型进行迁移。

[0163] 实体模型之间的访问

[0164] 如果目标实体模型与自己在同一个进程,则直接访问。如果不在一个进程,则通过网络访问。各进程之间一直保持TCP连接。

```

string Engine::getEntityProperty(int nEntityId, string strPropertyName)
{
    Entity entity = m_vecEntity[nEntityId];
    If(entity.nEngineId == m_ID) //位于本机
        return m_vecEntity[nEntity].getProperty(strPropertyName);
[0165]     else
        return
m_vecRemoteEngine[entity.nEngineId].getEntityProperty(nEntityId,
strPropertyName);//其他机器
}

```

```
string RemoteEngine::getEntityProperty(int nEntityId, string
strPropertyName)
{
    m_sock.write("GetProperty:%d %s", nEntityId, strPropertyName);
    return m_sock.read();
}

bool RemoteEngine::sendEntity(Entity entity)
{
[0166]     m_sock.wirte("PutProperty:");
        m_sock.write(entity.serialize());
}

bool RemoteEngine::recvEntity()
{
    Entity entity;
    entity.unserialize(m_sock.read());
    m_localEngine.m_vecEntity[entity.nId] = entity;
}
```

[0167] 最后,需要明确的是:本领域技术人员可根据实际需求,将上述多个实施例进行自由组合。

[0168] 对应本发明实施例提供的分布式仿真系统,本发明实施例还提供一种分布式仿真方法,该方法包括:

[0169] 通过各计算机分别监控所述各计算机其自身上分布的实体模型的仿真状态;

[0170] 获取所述各计算机其自身的资源使用情况;

[0171] 根据所述实体模型的仿真状态和/或所述各计算机其自身的资源使用情况,将实体模型在计算机之间进行迁移,以实现分布式仿真,所述各计算机上安装有仿真平台,分别分布有至少一个实体模型,用于并行仿真所述至少一个实体模型。

[0172] 在一个实施例中,所述各计算机上运行有至少一个仿真进程;

[0173] 通过所述各计算机上运行的仿真进程监控所述各计算机上分布的实体模型的仿真状态以及所述各计算机其自身的资源使用情况。

[0174] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0175] 通过所述各计算机中的仿真进程监控所述各计算机其自身上分布的实体模型的仿真状态,并将监控的实体模型的仿真状态发送至所述仿真管理进程;

[0176] 通过所述仿真管理进程接收所述各计算机的仿真进程监控的实体模型的仿真状

态,并进行记录。

[0177] 在一个实施例中,通过所述各计算机的仿真进程确定所述各计算机其自身上分布的所有实体模型分别对应的仿真步长中的当前最小步长,并将所述当前最小步长发送至所述仿真管理进程;

[0178] 控制所述仿真管理进程根据所述各计算机的仿真进程发送的当前最小步长确定下一次仿真所述所有实体模型的最小时间步长,并将所述最小时间步长发送至所述各计算机。

[0179] 在一个实施例中,所述各计算机中存在一个主控计算机,所述主控计算机上运行的仿真进程为仿真管理进程;

[0180] 通过所述各计算机的仿真进程将所述各计算机其自身的资源使用情况发送至所述仿真管理进程;

[0181] 通过所述仿真管理进程接收所述各计算机的仿真进程发送的其自身的资源使用情况,并根据所述其自身的资源使用情况确定所述各计算机上分布的实体模型的迁移方向,所述资源使用情况包括资源使用率,所述资源包括CPU和/或内存。

[0182] 在一个实施例中,控制所述仿真管理进程根据所述各计算机的仿真进程发送的所述各计算机其自身的资源使用率,确定所述各计算机中资源使用率最高的第一目标计算机;

[0183] 控制所述仿真管理进程在所述第一目标计算机的资源使用率高于第一使用率时,将所述第一目标计算机上的实体模型迁移到所述各计算机中除所述第一目标计算机之外的第二目标计算机。

[0184] 在一个实施例中,控制所述仿真管理进程确定所述第一目标计算机上资源使用率排名第 i 的实体模型,并计算所述排名第 i 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率;以及

[0185] 若所述第二目标计算机的预计资源使用率大于所述第一目标计算机迁移所述第 i 的实体模型之后的当前资源使用率,则计算所述第一目标计算机上资源使用率排名第 $i+1$ 的实体模型迁移至所述第二目标计算机后,所述第二目标计算机的预计资源使用率,依次类推,直至所述第二目标计算机的预计资源使用率不大于所述第一目标计算机迁移第 n 实体模型之后的当前资源使用率时,确定将所述第一目标计算机上的第 n 实体模型迁移至所述第二目标计算机,所述第一目标计算机上共有 N 个实体模型, n 大于 $i+1$ 且小于 N ,资源使用率排名为资源使用率从高到低的排名, i 为不小于2的正整数, N 为正整数。

[0186] 在一个实施例中,控制所述仿真管理进程在所述各计算机的资源使用率低于第二使用率时,确定所述多个仿真进程中第 P_m 个仿真进程上分布的各实体模型对所述多个仿真进程中第 P_i 个仿真进程上分布的实体模型的访问次数;

[0187] 控制所述仿真管理进程基于所述第 P_m 个仿真进程上分布的各实体模型对所述第 P_i 个仿真进程上分布的实体模型的访问次数,计算所述第 P_m 个仿真进程上分布的各实体模型从所述第 P_m 个仿真进程迁移到对应访问次数最多的第 P_{max} 个仿真进程上的迁移收益;

[0188] 控制所述仿真管理进程进行迁移收益排名,并将所述第 P_m 个仿真进程上迁移收益最大的实体模型迁移至所述第 P_{max} 个仿真进程上,其中,所述多个仿真进程的编号为 $P_0 \sim P_n$, P_m 、 P_i 、 P_{max} 为所述 $P_0 \sim P_n$ 个仿真进程中的仿真进程。

[0189] 在一个实施例中,所述仿真管理进程基于以下步骤计算将所述多个计算机中一个计算机上的实体模型迁移到另一个计算机上获得的迁移总收益;

[0190] 设 W_i 为仿真进程 P_i 的负载比重,其计算公式如下:

$$[0191] \quad W_i = \frac{K_c * U_c(i) + K_m * U_m(i) + K_d * U_d(i) + K_b * U_b(i)}{K_c + K_m + K_d + K_b}$$

[0192] 其中 $U_c(i)$ 、 $U_m(i)$ 、 $U_d(i)$ 、 $U_b(i)$ 分别表示单位时间 T 内仿真进程 P_i 占用的CPU、内存、磁盘IO和网络带宽占用率, K_c 、 K_m 、 K_d 、 K_b 分别表示CPU、内存、磁盘IO和网络带宽对计算机的预设影响因子,且 $K_c + K_m + K_d + K_b = 1$;

[0193] 设 e 为被迁移进程的计算机编号, f 为迁移的目标计算机编号, $e=1,2,3,\dots,E$, $f=1,2,3,\dots,F$, M_{ie} 表示将仿真进程 P_i 上分布的实体模型从计算机 e 迁移到计算机 f 后计算机 e 的迁移收益:

$$[0194] \quad M_{ie} = \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right)}{e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n-1} W_i}{n-1}}\right) - e^{\frac{\sum_{i=1}^{n-1} W_i}{n-1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n-1} W_i - (n-1) \sum_{i=1}^n W_i}{n(n-1)}}$$

[0195] M_{if} 表示将仿真进程 P_i 迁移到计算机 f 后计算机 f 的迁移收益:

$$[0196] \quad M_{if} = \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right)}{e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}}} - \frac{\max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{\sum_{i=1}^n W_i}{n}}}$$

$$= \frac{e^{\frac{\sum_{i=1}^n W_i}{n}} \max\left(e^{W_i - \frac{\sum_{i=1}^{n+1} W_i}{n+1}}\right) - e^{\frac{\sum_{i=1}^{n+1} W_i}{n+1}} \max\left(e^{W_i - \frac{\sum_{i=1}^n W_i}{n}}\right)}{e^{\frac{n \sum_{i=1}^{n+1} W_i - (n+1) \sum_{i=1}^n W_i}{n(n+1)}}$$

[0197] 则仿真进程 P_i 上的实体模型从计算机 e 迁移到计算机 f 的总收益为 M_i :

$$[0198] \quad M_i = M_{ie} + M_{if}$$

[0199] 根据 M_i 的大小进行排序, M_i 越大,迁移收益越高;

[0200] 将总收益 M_i 最高的计算机 e 上的仿真进程上分布的实体模型迁移至计算机 f 上。

[0201] 在一个实施例中,所述仿真管理进程还用于:

[0202] 控制根据所述各计算机中的仿真进程发送的所述各计算机其自身上分布的实体模型的仿真状态,生成所述各计算机上的实体模型的当前仿真记录;

[0203] 调用预先获取的实体模型在实际运行环境下的预设状态记录;

[0204] 根据所述当前仿真记录和所述预设状态记录,生成至少一个实体模型仿真运行指令;

[0205] 将所述至少一个实体模型仿真运行指令广播至所述各计算机中的仿真进程,以使所述各计算机中的仿真进程将所述至少一个实体模型仿真运行指令发送至所述各计算机其自身上分布的实体模型中对应的实体模型,从而控制所述对应的实体模型的仿真过程或者调节所述对应的实体模型的模型参数。

[0206] 在一个实施例中,所述方法还包括:

[0207] 采用通用建模工具对所述各计算机其自身上分布的实体模型进行数学模型搭建,得到简易仿真模型;

[0208] 采用专业建模工具对所述各计算机其自身上分布的实体模型进行数学模型搭建,得到精细仿真模型;

[0209] 以所述精细仿真模型为衡量标准,对所述简易仿真模型的模型结构和模型参数做初步验证,并对不符合第一预设要求的模型结构和模型参数进行修正;

[0210] 利用所述各计算机其自身上分布的实体模型的第一组实验数据对初步验证修正后的简易仿真模型做再次验证,并对所述修正后的简易仿真模型的模型参数做再次修正,直至相同的输入下修正后的简易仿真模型的输出数据与所述各计算机其自身上分布的实体模型对应的输出数据的一致性衡量指标满足预设接受值,获得根据所述第一组实验数据得到的简易仿真模型;

[0211] 利用所述各计算机其自身上分布的实体模型的第二组实验数据作为所述第一组实验数据得到的简易仿真模型的输入,经过所述简易仿真模型的仿真计算,得到所述第二组实验数据对应的输出数据;

[0212] 计算所述第二组实验数据对应的输出数据与所述各计算机其自身上分布的实体模型对应的输出数据的一致性衡量指标;

[0213] 判断所述第二组实验数据的一致性衡量指标是否满足所述预设接受值;

[0214] 如果是,则将所述第二组实验数据得到的简易仿真模型作为最终确定简易仿真模型。

[0215] 本领域技术人员在考虑说明书及实践这里发明的发明后,将容易想到本发明的其它实施方案。本申请旨在涵盖本发明的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本发明的一般性原理并包括本发明未发明的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本发明的真正范围和精神由下面的权利要求指出。

[0216] 应当理解的是,本发明并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本发明的范围仅由所附的权利要求来限制。

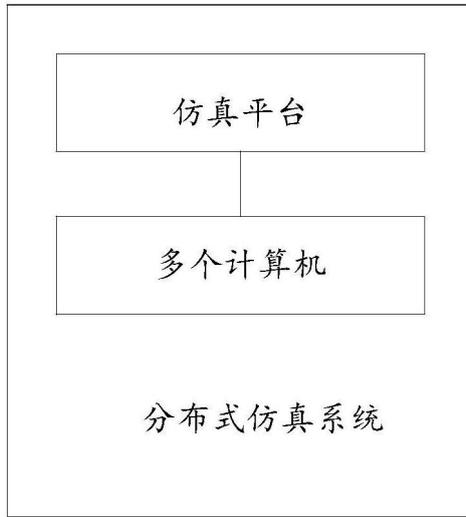


图1