



(12) 发明专利申请

(10) 申请公布号 CN 114138898 A

(43) 申请公布日 2022. 03. 04

(21) 申请号 202111375039.8

(22) 申请日 2021.11.20

(66) 本国优先权数据

202111337330.6 2021.11.15 CN

(71) 申请人 道和邦(广州)电子信息科技有限公司

地址 510440 广东省广州市白云区嘉禾街
鹤龙二路96号粤旺大厦C栋305室

(72) 发明人 不公告发明人

(51) Int. Cl.

G06F 16/27 (2019.01)

G06F 9/50 (2006.01)

G06F 9/448 (2018.01)

G06F 9/455 (2006.01)

G06F 21/60 (2013.01)

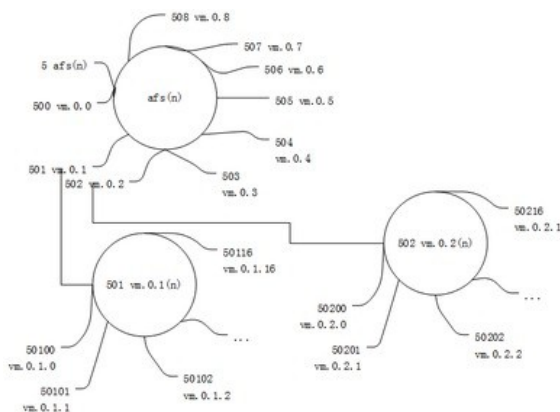
权利要求书4页 说明书9页 附图2页

(54) 发明名称

SMG-VME-AFS可迭代分布式存储系统

(57) 摘要

本发明:SMG-VME-AFS可迭代分布式存储系统。利用了斐波那契数列迭代变换在分布式操作系统,以及分布式存储系统上的具体实现。本发明可广泛用于超大集团国家和企业对核心数据的分布式存储,实现算力共享,分布式存储能力共享;对核心数据的调度共享;对大用户数据行为分析以及隐私保护;对用户智能合约调度;打通企业信息孤岛,对构建和谐可编程智慧世界实现智慧互联网有重要意义。



1. 一种可迭代算法的分布式存储系统;

1.1 基于数学可迭代函数 $f(n) = f(n-1) + f(n-2)$ 算法推论: 根据: $f(n) = f(n-1) + f(n-2)$, $n > 2$ 自然数, 斐波那契数列

变化 n

$\Rightarrow f(n) = f(n+1) + f(n+2)$

合并 $f(n+1)$, $f(n+2)$

$\Rightarrow f(n) = f(n+1)$

由大到小 把函数 f 抽象为 TSun 超恒星系统(太太阳), Sun 恒星系统(太阳) Planet 行星系统(地球), n 替换为 x 则由大到小推论如下

$\Rightarrow \text{TSun}(x) = \text{TSun}(x+1)$

$\Rightarrow \text{TSun}(x) = \text{Sun}(m) + \text{Sun}(n) + \dots$

$\Rightarrow \text{Sun}(x) = \text{Planet}(m) + \text{Planet}(n) + \dots$

$\Rightarrow \text{Planet}(x) = \text{Planet}(x+1)$

$\Rightarrow \text{vme}(x) = \text{vme}(x+1)$

变换 f

$\Rightarrow \text{afs}(x) = \text{afs}(x+1)$

afs 为可迭代分布式存储引擎

vme: 为可迭代分布式操作系统

1.2 一种分布式数据库系统, 能够存储基于 KV 键值对数据的非关系形数据库系统; 其特征: 能支持任意 $\$key=val\&\dots$ 数据格式化存储, 不限存储数据容量, 不限存储数据类型, 且数据可以自动扩容动态增长; 数据可以被更改

1.3 一种可容量超大数据的存储系统, 可以将大数据打散, 可以按制定 hash 算法散列在一种特殊的节点容器分布式存储, 节点与父节点以 2 平方的关系, 节点之间是遵循 1.1 算法可迭代;

1.4 基于 1.1 一种可以无限级路由扩展的分布式存储系统: 其特征节点扩容是按 2 的 n 次方进行父子级级联扩容。

2. 一种宇宙全息可迭代式的分布式系统;

其特征:

2.1 独立运作的操作系统; 每个子系统与父系统有联系, 即子系统可被父系统调度;

2.2 子系统与父系统共享库;

2.3 子系统通过父系统传递指令与结果;

2.4 每个 afs 集成在 vme (可迭代分布式操作系统) 下对数据进行分布式存储;

2.5 宇宙全息: 宇宙全息, 宇宙的个体与整体是统一继承关系, 个体无论再小都具有宇宙整体信息片段 DNA; 故此, 每个节点包含全息的架构体系, 最小的个体单元计算, 包含整体集群的计算组件;

2.6 每个节点数据是所有节点信息的一部分, 个体迭代计算输出给整体集群, 个体与集群是可迭代关系;

2.7 所有节点组合超大规模分布式存储, 但节点与节点是有统一的指令系统调度。

3. 一种可迭代宇宙全息最小单元 $\text{vme}(n)$ 程序集, 其构成如下:

(后小标引用之架构图) 3.0 4 vme(n)可迭代分布式操作系统 3.1 401 linuxOS 操作系统 3.2 402 java wrapper服务 3.3 403 jvm 3.4 404 ioserver通讯框架 3.5 405 wscomm. 公共模块 3.6 406 ppg 虚拟机引擎服务 3.7 407 ppgmt接口服务 3.8 408 application. 用户端应用 3.9 409 mbs可迭代商城。

4. 一种可执行智能合约程序的可迭代分布式存储系统

其特征如下:

afs(n) 简写afs

4.1 afs分布式存储对核心数据制定算法加密存储,同时支持制定算法解密,该算法快速有效,并支持高倍压缩算法生成密文;

4.2 afs 支持入存用户个体属性数据,数据应包含制定智能合约,包含制定智能函数,包含本节约定的数据加密、解密读写服务;

4.3 afs 支持制定智能合约解析器,并能够解析4.2 中约定的智能合约,智能函数,解密并运算,运算结果输出给afs,并支持写入前计算,并存储;

4.4智能合约:一种类java的计算机小程序,由afs自带的智能合约引擎解析并执行,支持简单数学运算,以及条件判断;智能合约是书面合同中涉及到双方利益部分的数学表达式,可以在限定的条件下由系统触发执行,以程序化电子化保持在afs节点用户数据块下,只能被afs智能合约解析引擎执行,存储智能合约需要双方的数字有效签名;

4.5 afs 支持4.3中数据计算过程不出afs存储底层,完成,智能合约约定的加密,解密,解析并运算执行,其结果写回存储器;

一种可支持超大容量且能秒存秒取的可迭代分布式存储系统

其特征如下:

5.0 其数据被制定hash算法,散列存储在集群下的分布式节点进行容量存储;

5.1每扩展一维 子节点按2几何倍数扩充一倍;

5.2系统扩容后,原数据节点自动升级为当前父节点,原节点数据自动重新按制定hash算法均衡的流入,当前父节点下所有子节点数据,原寻址算法同样适用制定hash算法调用(亦可寻址定位到老数据),制定hash算法要能包括父子集群维度因子;

5.3每层节点依次实现分布式forawrd,backward,router协议,实现与上级节点通讯与数据传输;

5.4各兄弟节点调动算法完全相同,即同时符合f可迭代函数规则;

5.5 各数据节点满足制定hash算法,数据分散性特点,同时兼顾集群升维因子,故在子集群升级节点过程原数据确保不会被散列到其他集群;散列算法原则,那里来回那里去;

5.6 各数据节点负载满足均衡性特点;

5.7数据一致性,幂等性写运算: $f(f(x))=f(x)$,意义在于可重复运算n次,只是第一次成功写入为准;

5.8 afs 支持用户空间数据独立性,私密性,以及用户数据保护;

5.9 afs 支持用户区块属性数据的动态改写,并支持存储空间的自动扩容,自动扩容旧数据空间被自动回收;

5.10 afs支持用户区块数据读取秒存秒取,且不受用户规模的限制,因集群扩展与节点扩展遵循2的n次方规则;

5.11 afs 支持定量用户区块数据的块写,闪存;

5.12 afs支持多用户的并发写,并保证数据的唯一性;

5.13 afs支持异地集群的远程读写,以及服务调用,mapsend 数据代理发送,mapsum 代理汇总(类google GFS mapreduce) ,数据每提升一层数据规模缩减一半;

5.14 安全性:afs 支持自动部署并aDDoS(反DDoS攻击)入侵检测的能力,非法入侵自动进入黑名单绑定;

5.15 最小单元集群,单物理机集群,支持1x(8+1)x16:1个物理机,下配置8个虚拟机,1个虚拟机root,每个虚拟机下配置16个vme,root不存数据只配置3个vme负责监控调度 ;

5.16 支持5.16单集群克隆扩展多多物理机集群的能力;

5.17 最小单元集群5.16中虚拟机网络规划规则如下:

#名称	地址	进程名	功能
vm.0	192.168.1.179:9000	ppg0	root
vm.0.1	192.168.1.180:9100		vm
vm.0.2	192.168.1.181:9200		vm
...			
vm.0.8	192.168.1.187:9800		vm

从虚拟机1-8 依次分配vme

虚拟机1:

#名称	端口	进程名	功能
vm.0.1.0	9100	ppg0	root
vm.0.1.1	9101	ppg1	vme
vm.0.1.2	9102	ppg2	vme
...			
vm.0.1.16	9116	ppg16	vme

虚拟机2:

#名称	端口	进程名	功能
vm.0.2.0	9200	ppg0	root
vm.0.2.1	9201	ppg1	vme
vm.0.2.2	9202	ppg2	vme
...			
vm.0.2.16	9216	ppg16	vme

...

虚拟机8:

#名称	端口	进程名	功能
vm.0.8.0	9800	ppg0	root
vm.0.8.1	9801	ppg1	vme
vm.0.8.2	9802	ppg2	vme
...			
vm.0.8.16	9816	ppg16	vme

注:

5.17.1 1-16 为数据节点;

5.17.2 倒数第2位点分隔符元素为虚拟机编号;

5.17.3 端口号9xxx 第2位即虚拟机编号;

5.17.4将上述5.17.3 端口号第一位 9换成8,即8xxx的端口号,其他规则如上,则表示给vme shell管理用的,9xxx是给程序接口afs调用

5.17.5 一种快速网络寻址算法,依据权利5.17.1,5.17.2,5.17.3,5.17.4 可以通过虚拟机编号,快速算得ip,亦可通过ip轻易算得编号;亦可通过端口号算得节点编号,ip末尾号,通过动态映射算法可轻易算得寻址;

5.18 mapport 端口映射服务

物理机宿主操作系统:Windows Server 2012(Fix)

通过portproxy 制定mapport.bat 进行端口路由映射 物理机如图网络适配器;要将9xxx,8xxx 端口 陆续映射给内网各个虚拟机使用

```
mapport $oip $vmeip $oport $vmeport
```

```
eg:mapport 192.168.1.170 192.168.1.179 9000 9000
```

\$oip 物理机ip \$vmeip 虚拟机vmeip \$oport 物理机端口 \$vmeport vme节点端口;

mapportbs.bat 可批量在本集群下实现所有虚拟机,vme端口9xxx ,shell 8xxx;

通过mapport 服务打通物理机与虚拟机,vme之间的ip通路;

5.19 基于5.18 联通各个节点与物理之间连接,物理机与外网绑定,虚拟机与物理机采用桥连,虚拟机与物理机共享外网ip,通过ip+端口即可访问任意虚拟机,以及vme节点,因端口唯一分配;

5.20 基于5.19 完成集群内联网,重复迭代复制物理机更换ip,即可复制集群,且集群内无需要再额外配置,只要更换ip,即可迭代到多集群实现升维概念,实现权利1,权利2中可迭代升级。

SMG-VME-AFS可迭代分布式存储系统

技术领域

[0001] 本发明是为了解决分布式大数据环境下,如何快存快取,分布式存储共享;以及并行计算,高效率的集群分布式存储。

[0002] 现有技术:分布式存储普遍采用的开源架构,apache组织的hadoop基于google GFS架构,现有技术体现开源,涉及组件背景复杂,调用api参差不齐,组件调用学习成本高,设计缺陷。

[0003] 本发明基于数学迭代函数斐波那契数列推论出的程序集。将CPU,内存,存储给解放出来更多集中于实体服务。afs可迭代分布式存储亦叫量子(可迭代)分布式存储,理论上存储数据无限制。数据存储规模与存储器非线性增长关系。传统数据存储器呈线性增长。本发明存储器扩容,以2的n次方,指数级扩容。

[0004] 最小单元集群建立分布式存储,实测10亿+用户在单集群下做到秒存秒取;理论性能可达100万级/秒数通信调度。

[0005] 本项发明,依赖于SMG-VME可迭代分布式操作系统,具体参国家专利申请号:2021113373306,电子申请案卷编号:364636644,得以继承实现。亦可单独作为核心产品服务模块单独定制算力输出;分布式存储输出。

[0006] 专利审查:

截至2021-11-20网络搜索暂无同案例

背景技术

[0007] 随着工业互联网,万物智能互联,分布式计算,大数据推衍发展,超大规模数据的计算与存储摆在面前。本发明主要是研究并行计算引擎,超大规模数据存储与计算而针对性设计出一套可行的框架。对外进行算力输出构建 IAAS,SAAS系统。超集团企业共享算力,共享存储。提升CPU使用效率,内存存储效率,提升分布式数据存储能力。

图标说明

图1是可迭代分布式操作系统抽象类比图;

1超恒星系统Tsun (x)

2太阳系sun (m)

3行星系统planet (n)

4可迭代分布式操作系统vme (n)

推衍:TSun (x) 1->2->3->4vme (n) 由大到小;反推则为宇宙裂变原理

图2是vme (n) 系统组件功能图;

4 vme (n) 可迭代分布式操作系统

401 linuxOS操作系统,centos6.8 (FX-SMG)

402 java wrapper服务

403 jvm JDK1.78

404 ioserver通讯框架,轻量级nio通讯框架

405 wscomm公共模块
 406 ppg系统模式引擎服务
 407 ppgmt业务模式引擎接口服务
 408 application用户服务
 409 mbs (n) 可迭代商城

5 afs (n) 可迭代分布式存储引擎 (量子存储)

图3为afs (n) 可迭代分布式存储引擎 (量子存储) 分区节点图;

5 afs (n) 可迭代分布式存储引擎

500 vm.0.0根用顶级根vm.0表示

此外每级节点末号为0的为当前节点根,根不负责具体数据存储只负责指令传递,与子节点管理

501 vm.0.1 1号虚拟机数据节点

502 vm.0.2 2号虚拟机数据节点

...

508 vm.0.8 8号虚拟机数据节点

50100为1号根 (尾号逢00为父根)

50101 vm.0.1.1 1号1数据节点

50102 vm.0.1.2 1号2数据节点

...

50116 vm.0.1.16 1号16数据节点

每级子根与父节点的数量为2的倍数关系

发明内容

[0008] 可迭代算法的分布式存储系统

1.1算法依据:根据斐波那契数列: $f(n) = f(n-1) + f(n-2)$, $n > 2$ 自然数,斐波那契数列,为宇宙裂变函数。自然界普遍存在以此推衍案例,如海螺贝壳,宇宙星系,凡是符合斐波那契数列,即是符合万物裂变。

[0009] 变化n

$\Rightarrow f(n) = f(n+1) + f(n+2)$

合并 $f(n+1)$, $f(n+2)$

$\Rightarrow f(n) = f(n+1)$

由大到小

如图1

把函数f抽象为TSun超恒星系统 (太太阳), Sun恒星系统 (太阳) Planet行星系统 (地球), n替换为x则由大到小推论如下

$\Rightarrow TSun(x) = TSun(x+1)$

$\Rightarrow TSun(x) = Sun(m) + Sun(n) + \dots$

$\Rightarrow Sun(x) = Planet(m) + Planet(n) + \dots$

$\Rightarrow \text{Planet}(x) = \text{Planet}(x+1)$

$\Rightarrow \text{vme}(x) = \text{vme}(x+1)$

变换f

$\Rightarrow \text{afs}(x) = \text{afs}(x+1)$

afs为可迭代分布式存储引擎

vme:为可迭代分布式操作系统.

$F = \text{TSun} = \text{sun} = \text{planet}$ f由高维到低维后面子项可迭代不变

这里高维与低维有联系即F裂变联系.

高维可调度低维,进行信息通讯与指令传输操作,低维可向高维度传递指令执行结果数据.最终 $\text{vme}(n) = \text{vme}(n+1)$ 成立,即为可迭代分布式操作系统.

可迭代分布式操作系统应该由以下基础组件构成,这些基础组件与业务无关.所有场景通用;其中ppg,ppg-mt,ioserver组件必须实现可迭代模式 $f(n) = f(n+1)$.

1.2一种分布式数据库系统,能够存储基于KV键值对数据的非关系形数据库系统;其特征:能支持任意 $\$key = val \& \dots$ 数据格式化存储,不限存储数据容量,不限存储数据类型,且数据可以自动扩容动态增长;数据可以被更改.

[0010] 可存储任意kv数据的字符串,随着用户属性增长,存储空间增长;老的空间会被删除,回收.数据会迁移到新的地址;仅靠该用户其他数据寻址关系不变,这样只改动当前增长用户数据的基地址;

1.3一种可容量超大数据的存储系统,可以将大数据打散,可以按制定hash算法散列在一种特殊的节点容器分布式存储,节点与父节点以2平方的关系,节点之间是遵循1.1算法可迭代;每段数据都存在唯一主键,通过制定hash算法求得基地址,就好比该用户被锚定,系统自动去指定地址去寻址,再进行定向 hash读取基地址数据;因为hash节点分布在不同的数据节点中,故每个节点负载均衡,数据均衡;

1.4基于1.1一种可以无限级路由扩展的分布式存储系统:其特征节点扩容是按2的n次方进行父子级联扩容;

参照图3结构为多层路;该分层可无限级路由下去;

2一种宇宙全息可迭代式的分布式系统

独立运作的操作系统;每个子系统与父系统有联系,即子系统可被父系统调度;

子系统与父系统共享库参照图2;

子系统通过父系统传递指令与结果vme实现forward,backward,router协议进行指令数据交换,层层迭代进行信息交互,这种路由协议被afs继承与共享,参照图3 架构;

每个afs集成在vme(可迭代分布式操作系统)下对数据进行分布式存储;

宇宙全息:宇宙全息,宇宙的个体与整体是统一继承关系,个体无论再小都具有宇宙整体信息片段DNA;故此,每个节点包含全息的架构体系,最小的个体单元计算,包含整体集群的计算组件参照图2;

每个节点数据是所有节点信息的一部分,个体迭代计算输出给整体集群,个体与集群是可迭代关系;

所有节点组合超大规模分布式存储,但节点与节点是有统一的指令系统调度.

[0011] 3最小单元vme(n)可迭代程序集参照图2

4vme (n) 可迭代分布式操作系统。401linuxOS操作系统。402java wrapper 服务。
403jvm。404ioserver通讯框架。405wscomm. 公共模块。406ppg虚拟机引擎服务。407ppgmt接
口服务。408application. 用户端应用。409mbs 可迭代商城。

[0012] ioserver, ppg, ppgmt, mbs组件服务必须实现1.1中约定的算法实现。

[0013] 一种可执行智能合约程序的可迭代分布式存储系统

其特征如下:

afs (n) 简写afs

4.1 afs分布式存储对核心数据制定算法加密存储, 同时支持制定算法解密, 该算
法快速有效, 并支持高倍压缩算法生成密文; 4.11对核心数据进行快速加密

4.12 4.11加密的数据再封装压缩(二进制压缩) 高倍压缩算法

4.13并支持反向解密

4.2 afs支持入存用户个体属性数据, 数据应包含制定智能合约, 包含制定智能函
数, 包含本节约定的数据加密、解密读写服务;

4.3 afs支持制定智能合约解析器, 并能够解析4.2中约定的智能合约, 智能函数,
解密并运算, 运算结果输出给afs, 并支持写入前计算, 并存储;

4.4智能合约: 一种类java的计算机小程序, 由afs自带的智能合约引擎解析并执
行, 支持简单数学运算, 以及条件判断; 智能合约是书面合同中涉及到双方利益部分的数学
表达式, 可以在限定的条件下由系统触发执行, 以程序化电子化保持在afs节点用户数据块
下, 只能被afs智能合约解析引擎执行, 存储智能合约需要双方的数字有效签名;

4.5 afs支持4.3中数据计算过程不出afs存储底层, 完成, 智能合约约定的加密,
解密, 解析并运算执行, 其结果写回存储器; 这种方式可以达到高效率执行; 运算过程不离
底层保证了数据隔离访问的安全性;

4.6 afs智能合约编译解析器, 为自定实现算法; 采用沙箱观察者模式执行, 所有
文件操作被严格控制;

本节的意义: 可以给用户智能合约, 让世界可编程执行, 最终使得智慧互联网成为
可能。

[0014] 一种可支持超大容量且能秒存秒取的可迭代分布式存储系统

其特征如下:

自操作系统图2, 401, 到上层mbs商城服务图2, 409, 均为可迭代; vme做操作系统层
参图2, 4, afs做分布式存储层, 以及智能合约; mbs为基础商城服务; afs服务端为ioserver
框架图2, 406, afs存储模块为ppg系统模式引擎参图2, 406, afs启动实例为ppgmt图2, 407业
务模式引擎服务。

[0015] 5.0其数据被制定hash算法, 散列存储在集群下的分布式节点进行容量存储; 每用
户只要确定用户唯一身份主键, 用户信息即被锚定在某一节点; 直接寻址到目标节点进行
数据读取, 写入操作; 读写操作是基于共享队列进行, 可保证了顺序性与数据唯一性, 隔离
性等特征。

5.1每扩展一维子节点按2几何倍数扩充一倍;

集群按层级以2的n次方进行路由扩展; 当前最低vm维度4即16个节点单vm. 只需要
扩展物理集群即可快速扩容存储能力; 而单元集群不需要改动; 这有助于批量工程部署能

力进行迭代升级。

5.2系统扩容后,原数据节点自动升级为当前父节点,原节点数据自动重新按制定hash算法均衡的流入,当前父节点下所有子节点数据,原寻址算法同样适用制定hash算法调用(亦可寻址定位到原始数据),制定hash算法要能包括父子集群维度因子;

哪里数据归那里集群;这样可以保证升级数据不会覆盖打乱全网,如果不考虑集群维度因子,则升级后原算法将路由不到具体数据。

5.3每层节点依次实现分布式forawrd,backward,router协议,实现与上级节点通讯与数据传输;

这三大协议保证各个集群的消息通讯能力。

5.4各兄弟节点调动算法完全相同,即同时符合f可迭代函数规则;

5.5各数据节点满足制定hash算法,数据分散性特点,同时兼顾集群升维因子,故在子集群升级节点过程原数据确保不会被散列到其他集群;散列算法原则,那里来回那里去;

5.6各数据节点负载满足均衡性特点;

因数据存储是采用制定hash算法,该算法也是散列均衡算法,故最终存储数据都会分布在不同节点;即符合数据分散性。

5.7数据一致性,幂等性写运算: $f(f(x)) = f(x)$,意义在于可重复运算n次,只是第一次成功写入为准;

有时候交易网络差的情况下,用户重复提交,该算法保证了只是第一次成功提交的有效,后面再重复提交不会被执行,保障了数据唯一性。

5.8 afs支持用户空间数据独立性,私密性,以及用户数据保护;

用户数据被散列在不同存储节点,即使地域相近的俩人可能也是在不同节点,寻址必须依据制定hash算法才能访问,私密性:核心数据被加密保护。

5.9 afs支持用户区块属性数据的动态改写,并支持存储空间的自动扩容,自动扩容旧数据空间被自动回收;

大用户被分布式存储在大的afs文件系统上,并做了索引,块地址;当用户改写会引起扩容导致地址改变,系统会自动开一个新块取代原始块,并copy数据到新块地址,重新索引;其他用户索引不变;已经溢出的块被标记垃圾回收,以填充更小的用户数据存储,避免过多的垃圾碎片。

5.10 afs支持用户区块数据读取秒存秒取,且不受用户规模的限制,因集群扩展与节点扩展遵循 2^n 次方规则;

读写队列采用高速队列进行操作,并支持多路迭代寻址;该算法保障快存快取操作数据能力。

5.11 afs支持定量用户区块数据的块写,闪存;

为了节省CPU避免减少磁盘IO,有专门的写线程,负责数据写,单个用户的写不会立即写入,如果访问空闲会触发超时写;如果用户多系统会迅速累计成快,然后一并提交产生块数据一次写到磁盘上,这个过程叫闪存.该算法降低了磁盘IO,提升了写效率,并提升了CPU使用率。

5.12 afs支持多用户的并发写,并保证数据的唯一性;

因为采用共享队列,所有数据先提交到内存,然后由队列进行写操作,队列支持并发先进先出FIFO操作,此算法保障数据唯一性.

5.13 afs支持异地集群的远程读写,以及服务调用,mapsend数据代理发送,mapsum代理汇总(类google GFS mapreduce),数据每提升一层数据规模缩减一半;

mapsend封装了远程调用一个客户端对服务器发起连接提交数据的能力,通过指定method=afs.xxxx方法告知调用哪个方法;系统会自动路由到该节点vme执行相应方法;代理客户端发起请求,完成指令发送,指令路由到目标节点完成指令执行.mapsum为统计所有节点注册用户数,会向全网发起请求,每个节点同步执行mapsum操作并汇总.

5.14安全性:afs支持自动部署并aDDoS(反DDoS攻击)入侵检测的能力,非法入侵自动进入黑名单绑定;

aDDoS为vmeserver自动的反DDoS攻击拦截器;会自动在限定时间让用户提交输入指令,如果错误,则自动主动关闭连接.该算法可有效率的抵制大量非法连接,减少无效的socket连接.

5.15最小单元集群,单物理机集群,支持1x(8+1)x16:1个物理机,下配置8个虚拟机,1个虚拟机root,每个虚拟机下配置16个vme,root不存数据只配置3个vme负责监控调度;

5.16支持5.16单集群克隆扩展多多物理机集群的能力;

5.17最小单元集群5.16中虚拟机网络规划规则如下:如图3

#名称 地址 进程名 功能

vm.0 192.168.1.179:9000 ppg0 root

vm.0.1 192.168.1.180:9100 vm

vm.0.2 192.168.1.181:9200 vm

...

vm.0.8 192.168.1.187:9800 vm

从虚拟机1-8依次分配vme

虚拟机1:

#名称 端口 进程名 功能

vm.0.1.0 9100 ppg0 root

vm.0.1.1 9101 ppg1 vme

vm.0.1.2 9102 ppg2 vme

...

vm.0.1.16 9116 ppg16 vme

虚拟机2:

#名称 端口 进程名 功能

vm.0.2.0 9200 ppg0 root

vm.0.2.1 9201 ppg1 vme

vm.0.2.2 9202 ppg2 vme

...

vm.0.2.16 9216 ppg16 vme

.....

虚拟机8:

```
#名称 端口 进程名 功能
vm.0.8.0 9800 ppg0 root
vm.0.8.1 9801 ppg1 vme
vm.0.8.2 9802 ppg2 vme
...
vm.0.8.16 9816 ppg16 vme
```

注:

5.17.1 1-16为数据节点;

5.17.2倒数第2位点分隔符元素为虚拟机编号;

5.17.3端口号9xxx第2位即虚拟机编号;

5.17.4将上述5.17.3端口号第一位9换成8,即8xxx的端口号,其他规则如上,则表示给vme shell管理用的,9xxx是给程序接口afs调用

5.17.5一种快速网络寻址算法,依据权利5.17.1,5.17.2,5.17.3,5.17.4可以通过虚拟机编号,快速算得ip,亦可通过ip轻易算得编号;亦可通过端口号算得节点编号,ip末尾号,通过动态映射算法可轻易算得寻址;

5.18 mapport端口映射服务

物理机宿主操作系统:Windows Server 2012 (Fix)

通过portproxy制定mapport.bat进行端口路由映射物理机如图网络适配器;要将9xxx,8xxx端口陆续映射给内网各个虚拟机使用

```
mapport$oip$vmeip$oport$vmeport
eg:mapport 192.168.1.170 192.168.1.179 9000 9000
```

\$oip物理机ip\$vmeip虚拟机vmeip\$oport物理机端口\$vmeport vme节点端口;mapportbs.bat可批量在本集群下实现所有虚拟机,vme端口9xxx,shell 8xxx;通过mapport服务打通物理机与虚拟机,vme之间的ip通路;

5.19基于5.18联通各个节点与物理之间连接,物理机与外网绑定,虚拟机与物理机采用桥连,虚拟机与物理机共享外网ip,通过ip+端口即可访问任意虚拟机,以及vme节点,因端口唯一分配;

5.20基于5.19完成集群内联网,重复迭代复制物理机更换ip,即可复制集群,且集群内无需要再额外配置,只要更换ip,即可迭代到多集群实现升维概念,实现权利1,权利2中可迭代升级。

[0016] 5.21最小集群:

1物理机至少含9个虚拟机0为根不存储数据图3,500vm.0.0 ip:192.168.1.179:9000

其他1-8个虚拟机为子节点虚拟机以存储数据参图3,501-3,508对应vm.0.1-vm.0.8,ip:192.168.1.180:9100->192.168.1.187:9800

每个数据虚拟机下有17个vme (ppg) ppg0为当前根
依次如下:

```

vm.0.1.0=ppg0=9100
vm.0.1.1=ppg1=9101
..
vm.0.1.16=ppg16=9116
…依次迭代到最后一个节点
vm.0.8.0=ppg0=9800
vm.0.8.1=ppg1=9801
..
vm.0.8.16=ppg16=9816

```

注:9xxx端口换成8xxx端口,规则参上则为vme shell远程登录端口,可以远程对某个vme (n) =ppg (n) 进行登录,操作。

[0017] vme父子关系

1级vm.0=root

2级vm.0.0=上级 (vm.0) ,当前子集:vm.0.1->vm.0.8

3级vm.0.1.0=上级 (vm.0.1) 当前子集:vm.0.1.1->vm.0.1.16

其他节点如是。

[0018] 特征意义:

a) 所有最小单元集群ip都是如此规划(规则不变),其好处是只需要复制克隆集群即可(可批量工程制造),然后在每个ppg属性文件,修改外网的ip.即可实现外网ip复用,实现万物级联互联互通;

b) 通过端口号即知道ip号,通过虚拟机号即知道ip号,根据ip末尾号即知虚拟机号,端口号。可以实现一号多用,一号复用;

c) 不需要查表。通过内置制定算法即可快速定位虚拟机vme子节点;

d) 多层路由有助于实现并行计算,因为是多通路几乎同时下达集群指令,所有子集群依次通知下达任务。

afs启动过程:

vme (n) 参图2中4->linuxOS启动参图2中401->参图2中402java wrapper->参图2中403jvm->参图2中406ppg->ppgServer启动->参图2中404ioserver框架->vmeServer启动参图2中407,406->VME (n) 启动->afs (n) 参图2中5启动 ->vme (n) ->遍历vm.0.0—vm.0.8启动虚拟机参图3,500-508

->每个虚拟机0节点遍历vm.0.\$x.0-vm.0.\$x.16启动VME参图3,50\$x00-50\$x16 \$x为虚拟机号。

注:\$x表示虚拟机编号,\$yy:ppg进程名号(vme编号)

虚拟机vme (n) ,afs (n) 处于就绪状态,侦听9\$x0\$yy\$x为虚拟机编号,\$yy为vme节点编号(下同)。同时ppgserver侦听8\$x0\$yy端口,此端口为远程shell控制台操作,vme有自己的独特认证体系。9\$x0\$yy的端口是给afs给程序远程存储数据用的。

[0019] 本发明现实意义:

本发明:利用了斐波那契数列迭代变换在分布式操作系统,以及分布式存储系统上的具体实现。本发明可广泛用于超大集团国家和企业对核心数据的分布式存储,实现算

力共享,分布式存储能力共享;对核心数据的调度共享;对大用户数据行为分析以及隐私保护;对用户智能合约调度;打通企业信息孤岛,对构建和谐可编程智慧世界实现智慧互联网有重要意义。

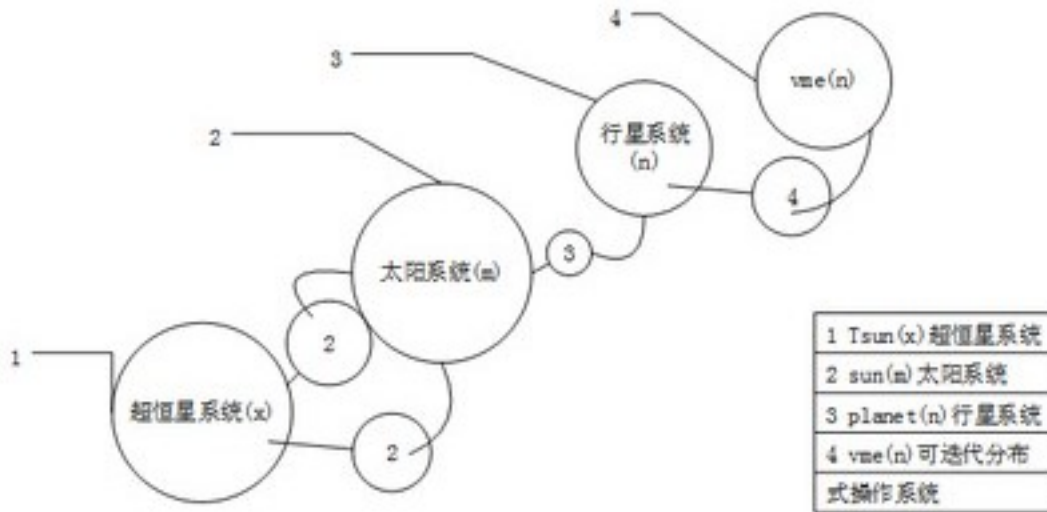


图1

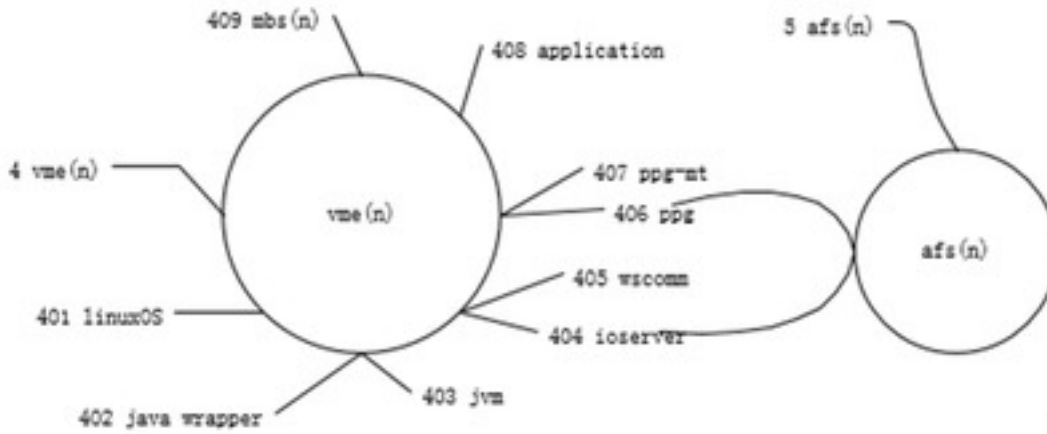


图2

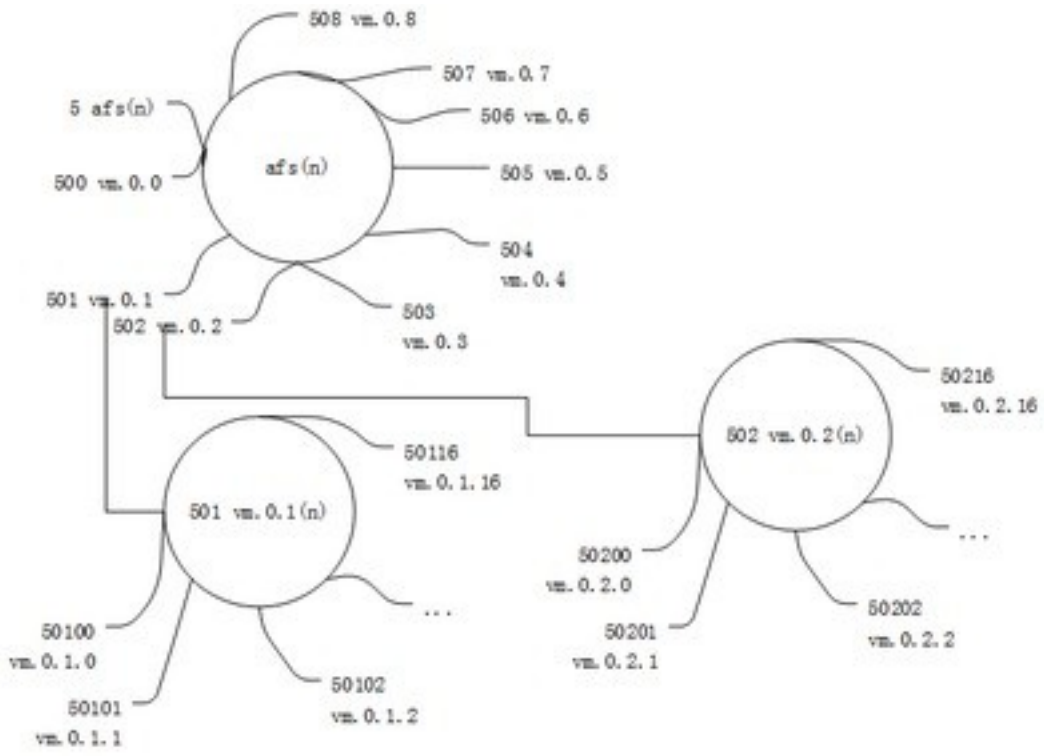


图3