(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 119166370 B
(45)授权公告日 2025.02.11

(21)申请号 202411679633.X

(22)申请日 2024.11.22

(65)同一申请的已公布的文献号
申请公布号 CN 119166370 A

(43)申请公布日 2024.12.20

(73)专利权人 福建中信网安信息科技有限公司
地址 350000 福建省福州市晋安区鼓山镇福光路318号2号楼8层

(72)发明人 高翔 翁武焰 金华松 张传辉

(74)专利代理机构 安徽新越诚途专利代理事务所(普通合伙) 34261
专利代理师 李浩宇

(51)Int.Cl.
G06F 9/50(2006.01)
G06N 3/04(2023.01)
G06N 3/08(2023.01)

(56)对比文件
CN 117827614 A,2024.04.05
CN 118945170 A,2024.11.12

审查员 刘锦英

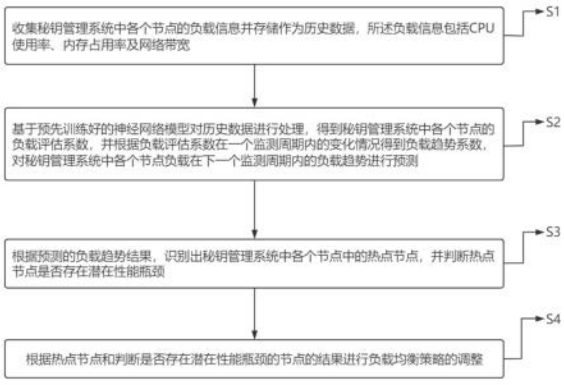
权利要求书2页 说明书7页 附图1页

(54)发明名称

一种基于AI的密钥管理系统负载均衡方法

(57)摘要

本发明涉及密钥管理领域,公开了一种基于AI的密钥管理系统负载均衡方法,包括先收集密钥管理系统中各个节点的负载信息并存储作为历史数据,所述负载信息包括CPU使用率、内存占用率及网络带宽;再基于预先训练好的神经网络模型对历史数据进行处理,得到密钥管理系统中各个节点的负载评估系数,并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数,对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测;再根据预测的负载趋势结果,识别出密钥管理系统中各个节点中的热点节点,并判断热点节点是否存在潜在性能瓶颈;最后根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整。



CN 119166370 B

1.一种基于AI的密钥管理系统负载均衡方法,其特征在于,包括:

S1、收集密钥管理系统中各个节点的负载信息并存储作为历史数据,所述负载信息包括CPU使用率、内存占用率及网络带宽;

S2、基于预先训练好的神经网络模型对历史数据进行处理,得到密钥管理系统中各个节点的负载评估系数,并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数,对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测;

S3、根据预测的负载趋势结果,识别出密钥管理系统中各个节点中的热点节点,并判断热点节点是否存在潜在性能瓶颈;

S4、根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整;

获取负载趋势系数的过程为:

步骤一、将一个监测周期等分为 n个子周期;

步骤二、拟合获取每个子周期内每个节点的负载评估系数随时间变化的实际曲线以及参考曲线;

步骤三、通过公式:

$$F_i = \gamma_i * K_i / n;$$

计算得到第 i个节点的负载趋势系数 F_i;

其中,K_i为变化系数,γ_i为转化系数,通过公式:

$$K_i = \left(\frac{\int_{t_1}^{t_2} |f_{i1}(t) - f_{i1th}(t)|dt}{(t_2 - t_1)} / \Delta\emptyset_{i1} + \ldots + \frac{\int_{t_{n-1}}^{t_n} |f_{in}(t) - f_{inth}(t)|dt}{(t_n - t_{n-1})} / \Delta\emptyset_{in} \right);$$

计算得到;

t_1、t_2分别为第1个子周期第 i个节点的起始时刻和结束时刻,t_{n-1}、t_n分别为第 n个子周期第 i个节点的起始时刻和结束时刻,$\Delta\emptyset_{i1}$、$\Delta\emptyset_{in}$分别为第1个子周期、第 n个子周期的变化参考值;$f_{i1}(t)$、$f_{in}(t)$分别为第1个子周期、第 n个子周期内负载评估系数随时间变化的实际曲线,$f_{i1th}(t)$、$f_{inth}(t)$分别为第1个子周期、第n个子周期内负载评估系数随时间变化的参考曲线;

对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测的过程为:

步骤一、将每个节点计算得到的上一个监控周期的负载趋势系数 F_i与预设的负载趋势阈值区间$(F_{i0min},\ F_{i0max})$进行比较;

步骤二、若$F_i \geq F_{i0max}$,则预测当前节点的负载处于升高趋势,否则进入步骤三;

步骤三、若$F_i \in (F_{i0min},\ F_{i0max})$,则预测当前节点的负载处于稳定趋势,否则进入步骤四;

步骤四,若$F_i \leq F_{i0min}$,则预测当前节点的负载处于下降趋势;

所述S3的工作过程为:

S31、将负载评估系数 f_i 与预设负载评估阈值 f_{i0} 进行比较，若 $f_i \geqslant f_{i0}$ 且 $F_i \geq F_{i0max}$，则判断当前节点为热点节点；

S32、基于AI实时获取当前节点的各项性能参数，所述性能参数为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率；

S33、将各项性能参数与预设的性能预警值进行比较，若至少存在一项超出性能预警值的性能参数项，则初步判定当前热点节点存在潜在性能瓶颈；

S34、将初步判断存在潜在性能瓶颈的热点节点列入二次分析队列中再次进行二次分析，并根据分析结果最终确定存在潜在性能瓶颈的节点名单；

S34中进行二次分析的过程为：

通过公式：

$$\varphi_i = \frac{\omega_i}{\omega_{i0}} * \sqrt{\frac{\sum_{k=1}^{N}(f_{ik} - f_{i0})^2}{N}};$$

计算得到当前节点的性能指数 φ_i；

其中，ω_i 为当前节点的性能指标，ω_{i0} 为当前节点的性能指标预警值，N 为采样总个数，f_{ik} 为第 k 个采样点获取的对应负载评估系数 f_i；

将当前节点的性能指数 φ_i 与当前节点的预设性能阈值 φ_{i0} 比较，若 $\varphi_i \geqslant \varphi_{i0}$，则判断当前节点存在潜在性能瓶颈，否则，判断当前节点不存在潜在性能瓶颈；

所述当前节点的性能指标 ω_i 通过公式：

$$\omega_i = \frac{\frac{A_i}{A_{i0}} * \alpha_1 + \frac{B_i}{B_{i0}} * \alpha_2 + \frac{C_i}{C_{i0}} * \alpha_3 + \frac{D_i}{D_{i0}} * \alpha_4}{\frac{E_i}{E_{i0}} * \alpha_5};$$

计算获得；

其中，A_i、B_i、C_i、D_i、E_i 分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率，A_{i0}、B_{i0}、C_{i0}、D_{i0}、E_{i0} 分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率的预警值，α_1、α_2、α_3、α_4、α_5 为预设权重系数；

进行负载均衡策略调整的过程为：

将热点节点中存在潜在性能瓶颈划分为一级高负载节点、将不存在潜在性能瓶颈的划分为二级高负载节点，其余满足 $f_i \geqslant f_{i0}$ 条件的节点划分为三级高负载节点；

将每一级的高负载节点按照负载评估系数降序排列后形成优先队列；

按照一级＞二级＞三级的排序，将每一级的优先队列中的高负载节点按顺序向满足 $f_i < f_{i0}$ 条件的低负载节点进行任务迁移且低负载节点按照负载评估系数升序排列来依次接收任务。

2.根据权利要求1所述的基于AI的密钥管理系统负载均衡方法，其特征在于，所述负载信息通过系统监控工具、日志记录或性能监控软件进行获得。

一种基于AI的密钥管理系统负载均衡方法

技术领域

[0001] 本发明涉及密钥管理领域,具体涉及一种基于AI的密钥管理系统负载均衡方法。

背景技术

[0002] 在数字化时代,随着数据量的激增和数据泄露风险的增加,密钥管理系统的重要性愈发凸显。它不仅是保护静态数据的关键,也是确保数据在传输过程中安全的基础。

[0003] 密钥管理系统在数据安全中扮演着至关重要的角色,是确保加密算法安全性的基础。密钥管理系统负责密钥的生成、存储、分发、使用、更新和销毁等全生命周期管理,确保密钥的安全性和合规性。密钥管理系统通过保护密钥免受未授权访问和泄露,维护数据的机密性、完整性和可用性,从而保护用户隐私和企业资产。

[0004] 为了提高系统可用性、优化资源利用,现有的密钥管理系统也有采用负载均衡策略,其目的是:通过负载均衡,密钥管理系统可以将工作负载分散到多个节点上,这样即使某个节点出现故障,其他节点仍能继续提供服务,从而保证系统的高可用性,同时,负载均衡可以有效地避免单点过载,确保系统中的计算资源和数据存储被均匀利用,提高整体的资源利用率。

[0005] 但是在实际使用中还是存在一些问题,例如,虽然负载均衡旨在优化资源利用率,但不当的配置可能导致某些节点成为瓶颈,影响整体性能。

发明内容

[0006] 本发明的目的在于提供一种基于AI的密钥管理系统负载均衡方法,解决以上技术问题。

[0007] 本发明的目的可以通过以下技术方案实现:

[0008] 一种基于AI的密钥管理系统负载均衡方法,包括:

[0009] S1、收集密钥管理系统中各个节点的负载信息并存储作为历史数据,所述负载信息包括CPU使用率、内存占用率及网络带宽;

[0010] S2、基于预先训练好的神经网络模型对历史数据进行处理,得到密钥管理系统中各个节点的负载评估系数,并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数,对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测;

[0011] S3、根据预测的负载趋势结果,识别出密钥管理系统中各个节点中的热点节点,并判断热点节点是否存在潜在性能瓶颈;

[0012] S4、根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整。

[0013] 作为进一步的技术方案,获取负载趋势系数的过程为:

[0014] 步骤一、将一个监测周期等分为 n 个子周期;

[0015] 步骤二、拟合获取每个子周期内每个节点的负载评估系数随时间变化的实际曲线以及参考曲线;

[0016] 步骤三、通过公式：

[0017] $$F_i = {\gamma_i * K_i}/{n}$$

[0018] 计算得到第i个节点的负载趋势系数F_i；

[0019] 其中，K_i为变化系数，γ_i为转化系数，通过公式：

[0020] $$K_i = \left(\frac{\int_{t_1}^{t_2} |f_{i1}(t) - f_{i1th}(t)|dt}{(t_2 - t_1)} / \Delta\emptyset_{i1} + \ldots + \frac{\int_{t_{n-1}}^{t_n} |f_{in}(t) - f_{inth}(t)|dt}{(t_n - t_{n-1})} / \Delta\emptyset_{in} \right)$$ 计算得到；

[0021] t_1、t_2分别为第1个子周期第i个节点的起始时刻和结束时刻，t_{n-1}、t_n分别为第n个子周期第i个节点的起始时刻和结束时刻，$\Delta\emptyset_{i1}$、$\Delta\emptyset_{in}$分别为第1个子周期、第n个子周期的变化参考值；$f_{i1}(t)$、$f_{in}(t)$分别为第1个子周期、第n个子周期内负载评估系数随时间变化的实际曲线，$f_{i1th}(t)$、$f_{inth}(t)$分别为第1个子周期、第n个子周期内负载评估系数随时间变化的参考曲线。

[0022] 作为进一步的技术方案，对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测的过程为：

[0023] 步骤一、将每个节点计算得到的上一个监控周期的负载趋势系数F_i与预设的负载趋势阈值区间(F_{i0min}, F_{i0max})进行比较；

[0024] 步骤二、若$F_i \geq F_{i0max}$，则预测当前节点的负载处于升高趋势，否则进入步骤三；

[0025] 步骤三、若$F_i \in (F_{i0min}, F_{i0max})$，则预测当前节点的负载处于稳定趋势，否则进入步骤四；

[0026] 步骤四，若$F_i \leq F_{i0min}$，则预测当前节点的负载处于下降趋势。

[0027] 作为进一步的技术方案，所述S3的工作过程为：

[0028] S31、将负载评估系数f_i与预设负载评估阈值f_{i0}进行比较，若$f_i \geqslant f_{i0}$且$F_i \geq F_{i0max}$，则判断当前节点为热点节点；

[0029] S32、基于AI实时获取当前节点的各项性能参数，所述性能参数为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率；

[0030] S33、将各项性能参数与预设的性能预警值进行比较，若至少存在一项超出性能预警值的性能参数项，则初步判定当前热点节点存在潜在性能瓶颈；

[0031] S34、将初步判断存在潜在性能瓶颈的热点节点列入二次分析队列中再次进行二次分析，并根据分析结果最终确定存在潜在性能瓶颈的节点名单。

[0032] 作为进一步的技术方案，S34中进行二次分析的过程为：

[0033] 通过公式：

[0034] $$\varphi_i = \frac{\omega_i}{\omega_{i0}} * \sqrt{\frac{\sum_{k=1}^{N} (f_{ik} - f_{i0})^2}{N}}$$

[0035] 计算得到当前节点的性能指数 φ_i；

[0036] 其中，ω_i为当前节点的性能指标，ω_{i0}为当前节点的性能指标预警值，N为采样总个数，f_{ik}为第 k个采样点获取的对应负载评估系数 f_i；

[0037] 将当前节点的性能指数 φ_i与当前节点的预设性能阈值 φ_{i0}比较，若$\varphi_i \geq \varphi_{i0}$，则判断当前节点存在潜在性能瓶颈，否则，判断当前节点不存在潜在性能瓶颈。

[0038] 作为进一步的技术方案，所述当前节点的性能指标 ω_i通过公式：

[0039] $$\omega_i = \frac{\frac{A_i}{A_{i0}} * \alpha_1 + \frac{B_i}{B_{i0}} * \alpha_2 + \frac{C_i}{C_{i0}} * \alpha_3 + \frac{D_i}{D_{i0}} * \alpha_4}{\frac{E_i}{E_{i0}} * \alpha_5}$$

[0040] 计算获得；

[0041] 其中，A_i、B_i、C_i、D_i、E_i分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率，A_{i0}、B_{i0}、C_{i0}、D_{i0}、E_{i0}分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率的预警值，α_1、α_2、α_3、α_4、α_5为预设权重系数。

[0042] 作为进一步的技术方案，所述负载信息通过系统监控工具、日志记录或性能监控软件进行获得。

[0043] 作为进一步的技术方案，进行负载均衡策略调整的过程为：

[0044] 将热点节点中存在潜在性能瓶颈划分为一级高负载节点、将不存在潜在性能瓶颈的划分为二级高负载节点，其余满足$f_i \geq f_{i0}$条件的节点划分为三级高负载节点；

[0045] 将每一级的高负载节点按照负载评估系数降序排列后形成优先队列；

[0046] 按照一级＞二级＞三级的排序，将每一级的优先队列中的高负载节点按顺序向满足$f_i < f_{i0}$条件的低负载节点进行任务迁移且低负载节点按照负载评估系数升序排列来依次接收任务。

[0047] 本发明的有益效果：

[0048] 本发明能够通过对每个节点负载趋势进行预测，再根据预测的结果识别出热点节点，通过预先判断哪些节点存在潜在性能瓶颈的方式，为及时调整预先设定好的负载均衡策略进行实时动态调整，防止对热点节点的不当配置导致在下一监测周期运行时出现性能瓶颈，导致密钥管理系统性能受限的问题。

附图说明

[0049] 下面结合附图对本发明作进一步的说明。

[0050] 图1为本发明的方法步骤图。

具体实施方式

[0051] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其它

实施例,都属于本发明保护的范围。

[0052] 请参阅图1所示,本发明为一种基于AI的密钥管理系统负载均衡方法,包括:

[0053] S1、收集密钥管理系统中各个节点的负载信息并存储作为历史数据,所述负载信息包括CPU使用率、内存占用率及网络带宽;

[0054] S2、基于预先训练好的神经网络模型对历史数据进行处理,得到密钥管理系统中各个节点的负载评估系数,并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数,对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测;

[0055] S3、根据预测的负载趋势结果,识别出密钥管理系统中各个节点中的热点节点,并判断热点节点是否存在潜在性能瓶颈;

[0056] S4、根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整。

[0057] 获取负载趋势系数的过程为:

[0058] 步骤一、将一个监测周期等分为n个子周期;

[0059] 步骤二、拟合获取每个子周期内每个节点的负载评估系数随时间变化的实际曲线以及参考曲线;

[0060] 步骤三、通过公式:

[0061] $$F_i = {\gamma_i * K_i}/{n}$$

[0062] 计算得到第i个节点的负载趋势系数F_i;

[0063] 其中,K_i为变化系数,γ_i为转化系数,基于历史实验数据选择确定,通过公式:

[0064] $$K_i = \left(\frac{\int_{t_1}^{t_2} |f_{i1}(t) - f_{i1th}(t)|dt}{(t_2 - t_1)} / \Delta\emptyset_{i1} + \ldots + \frac{\int_{t_{n-1}}^{t_n} |f_{in}(t) - f_{inth}(t)|dt}{(t_n - t_{n-1})} / \Delta\emptyset_{in} \right)$$ 计算得到;

[0065] t_1、t_2分别为第1个子周期第i个节点的起始时刻和结束时刻,t_{n-1}、t_n分别为第n个子周期第i个节点的起始时刻和结束时刻,$\Delta\emptyset_{i1}$、$\Delta\emptyset_{in}$分别为第1个子周期、第n个子周期的变化参考值,基于历史数据和实验数据综合分析确定;$f_{i1}(t)$、$f_{in}(t)$分别为第1个子周期、第n个子周期内负载评估系数随时间变化的实际曲线,$f_{i1th}(t)$、$f_{inth}(t)$分别为第1个子周期、第n个子周期内负载评估系数随时间变化的参考曲线。

[0066] 本实施例中,提供了一种能够避免不当配置导致某些节点成为性能瓶颈的密钥管理系统负载均衡方法并给出获取负载趋势系数的具体方法,具体的,先收集密钥管理系统中各个节点的负载信息并存储作为历史数据,所述负载信息包括CPU使用率、内存占用率及网络带宽,再基于预先训练好的神经网络模型对历史数据进行处理,得到密钥管理系统中各个节点的负载评估系数,并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数,对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测,再根据预测的负载趋势结果,识别出密钥管理系统中各个节点中的热点节点,并判断热点节点是否存在潜在性能瓶颈,最后根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整;通过上述技术方案能够通过对每个节点负载趋势进行预测,再根

据预测的结果识别出热点节点,通过预先判断哪些节点存在潜在性能瓶颈的方式,为及时调整预先设定好的负载均衡策略进行实时动态调整,防止对热点节点的不当配置导致在下一监测周期运行时出现性能瓶颈,导致密钥管理系统性能受限的问题;

[0067] 同时,通过公式:$K_i=\left(\frac{\int_{t_1}^{t_2}|f_{i1}(t)-f_{i1th}(t)|dt}{(t_2-t_1)}/\Delta\emptyset_{i1}+\ldots+\frac{\int_{t_{n-1}}^{t_n}|f_{in}(t)-f_{inth}(t)|dt}{(t_n-t_{n-1})}/\Delta\emptyset_{in}\right)$

计算得到第i个节点的负载趋势系数F_i,通过累积每个子周期内负载评估系数随时间变化的实际曲线与参考曲线之间偏差情况,再通过平均计算的方式得到当前节点准确的负载评估情况,从而为后续潜在性能瓶颈的判断提供有效的数据支持,降低误判的情况发生概率。

[0068] 对密钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测的过程为:

[0069] 将每个节点计算得到的上一个监控周期的负载趋势系数F_i与预设的负载趋势阈值区间$(F_{i0min},\ F_{i0max})$进行比较;若$F_i\geq F_{i0max}$,则预测当前节点的负载处于升高趋势;若$F_i\in(F_{i0min},\ F_{i0max})$,则预测当前节点的负载处于稳定趋势;若$F_i\leq F_{i0min}$,则预测当前节点的负载处于下降趋势;通过上述技术方案,实现了对当前节点负载趋势准确且快速的预测。

[0070] 所述S3的工作过程为:

[0071] S31、将负载评估系数f_i与预设负载评估阈值f_{i0}进行比较,若$f_i\geqslant f_{i0}$且$F_i\geq F_{i0max}$,则判断当前节点为热点节点;

[0072] S32、基于AI实时获取当前节点的各项性能参数,所述性能参数为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率;

[0073] S33、将各项性能参数与预设的性能预警值进行比较,若至少存在一项超出性能预警值的性能参数项,则初步判定当前热点节点存在潜在性能瓶颈;

[0074] S34、将初步判断存在潜在性能瓶颈的热点节点列入二次分析队列中再次进行二次分析,并根据分析结果最终确定存在潜在性能瓶颈的节点名单。

[0075] S34中进行二次分析的过程为:

[0076] 通过公式:

[0077] $$\varphi_i=\frac{\omega_i}{\omega_{i0}}*\sqrt{\frac{\sum_{k=1}^{N}(f_{ik}-f_{i0})^2}{N}}$$

[0078] 计算得到当前节点的性能指数φ_i;

[0079] 其中,ω_i为当前节点的性能指标,ω_{i0}为当前节点的性能指标预警值,N为采样总个数,f_{ik}为第k个采样点获取的对应负载评估系数f_i;所述当前节点的性能指标ω_i通过公式:

[0080] $$\omega_i=\frac{\frac{A_i}{A_{i0}}*\alpha_1+\frac{B_i}{B_{i0}}*\alpha_2+\frac{C_i}{C_{i0}}*\alpha_3+\frac{D_i}{D_{i0}}*\alpha_4}{\frac{E_i}{E_{i0}}*\alpha_5}$$

[0081] 计算获得；

[0082] 其中，A_i、B_i、C_i、D_i、E_i分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率，A_{i0}、B_{i0}、C_{i0}、D_{i0}、E_{i0}分别为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率的预警值，α_1、α_2、α_3、α_4、α_5为预设权重系数，基于历史数据选择拟定；

[0083] 当前节点的性能指数φ_i与当前节点的预设性能阈值φ_{i0}比较，若$\varphi_i\geqslant\varphi_{i0}$，则判断当前节点存在潜在性能瓶颈，否则，判断当前节点不存在潜在性能瓶颈。

[0084] 本实施例中，提供了一种对节点是否存在潜在性能瓶颈的判断方法，具体的，先将负载评估系数f_i与预设负载评估阈值f_{i0}进行比较，若$f_i\geqslant f_{i0}$且$F_i\geq F_{i0max}$，则判断当前节点为热点节点，再通过AI实时获取当前节点的各项性能参数，所述性能参数为每秒密钥事务数、每秒密钥查询请求数、吞吐量、并发用户数、错误率；随后通过将各项性能参数与预设的性能预警值进行比较，从而对当前热点节点是否存在潜在性能瓶颈进行初步判定，显然若存在，则表示该热点节点在未来会出现性能问题，最终影响整个系统的性能，但是为了防止预判，因此又通过二次分析的方式进行复判，结合初次判断和复判两次过程，达到准确判断存在潜在性能瓶颈的热点节点的目的；具体的二次分析过程即通过公式$\omega_i=\frac{\frac{A_i}{A_{i0}}*\alpha_1+\frac{B_i}{B_{i0}}*\alpha_2+\frac{C_i}{C_{i0}}*\alpha_3+\frac{D_i}{D_{i0}}*\alpha_4}{\frac{E_i}{E_{i0}}*\alpha_5}$计算获得当前节点的性能指标$\omega_i$，再代入到$\varphi_i=\frac{\omega_i}{\omega_{i0}}*\sqrt{\frac{\sum_{k=1}^{N}(f_{ik}-f_{i0})^2}{N}}$计算得到当前节点的性能指数$\varphi_i$，显然若每秒密钥事务数、每秒密钥查询请求数，吞吐量、并发用户数越大，则性能指标ω_i越大，若错误率越小，则性能指标ω_i越大，而性能指标ω_i占性能指标预警值ω_{i0}比值越大，则说明该热点节点越趋向于满载运行，缺少冗余，一旦再继续新增，则必然出现性能问题，因此该热点节点的性能指标直接导致整个密钥管理系统存在性能瓶颈，导致性能受影响，而$\sqrt{\frac{\sum_{k=1}^{N}(f_{ik}-f_{i0})^2}{N}}$则表示上一个监测周期内该节点的负载评估系数的波动性，波动性越大，则表示该节点的稳定性越差，因此越容易出现突然满载的情况；

[0085] 需要进行说明的是，本发明中通过AI实时获取当前节点的各项性能参数为现有技术中深度学习或机器学习，在此不再多做赘述；而每秒密钥事务数则为每秒能够处理秘钥管理相关事务的数量；每秒密钥查询请求数为每秒处理秘钥查询请求的数量；吞吐量为在单位时间内能够处理的数据量或请求任务数；并发用户数为能够同时处理的最大用户数

量;错误率为在单位时间内处理请求时发生的错误数与总请求数的比例。

[0086] 所述负载信息通过系统监控工具、日志记录或性能监控软件进行获得,性能监控软件可采用NewRelic、Datadog或Prometheus,能够实现性能参数的监控即可,在此不多做赘述。

[0087] 进行负载均衡策略调整的过程为:

[0088] 将热点节点中存在潜在性能瓶颈划分为一级高负载节点、将不存在潜在性能瓶颈的划分为二级高负载节点,其余满足$f_i \geqslant f_{i0}$条件的节点划分为三级高负载节点;

[0089] 将每一级的高负载节点按照负载评估系数降序排列后形成优先队列;

[0090] 按照一级>二级>三级的排序,将每一级的优先队列中的高负载节点按顺序向满足$f_i < f_{i0}$条件的低负载节点进行任务迁移且低负载节点按照负载评估系数升序排列来依次接收任务。

[0091] 本实施例中,通过将高负载节点的任务向低负载节点进行迁移,从而实现整个密钥管理系统中的每个节点均衡负载,以提高整体的性能和资源利用率。

[0092] 需要说明的是:本发明中的计算公式及各个参与运算的参数均预先经过无量纲处理,而无量纲处理的过程为行业公知,在此不进行叙述。

[0093] 以上对本发明的一个实施例进行了详细说明,但所述内容仅为本发明的较佳实施例,不能被认为用于限定本发明的实施范围。凡依本发明申请范围所作的均等变化与改进等,均应仍归属于本发明的专利涵盖范围之内。

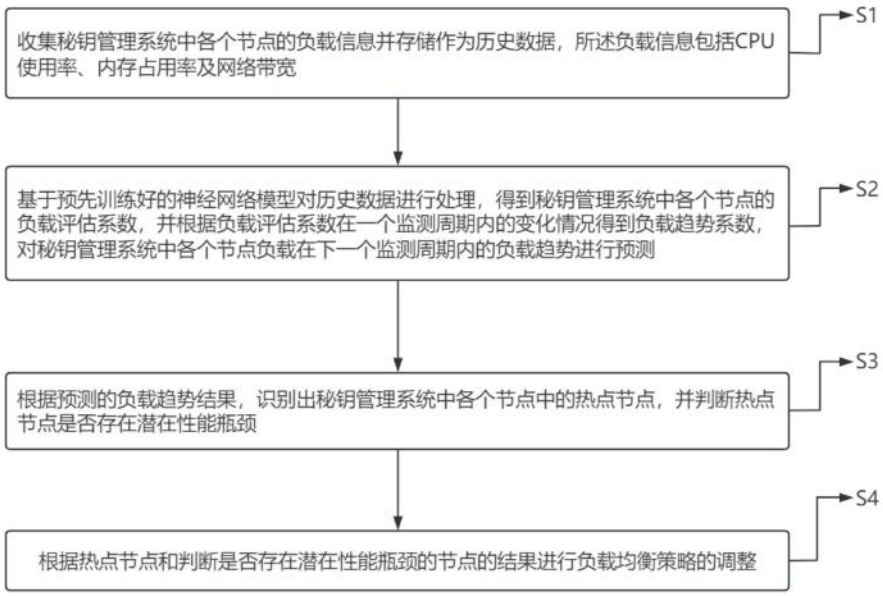

收集秘钥管理系统中各个节点的负载信息并存储作为历史数据，所述负载信息包括CPU使用率、内存占用率及网络带宽
S1
基于预先训练好的神经网络模型对历史数据进行处理，得到秘钥管理系统中各个节点的负载评估系数，并根据负载评估系数在一个监测周期内的变化情况得到负载趋势系数，对秘钥管理系统中各个节点负载在下一个监测周期内的负载趋势进行预测
S2
根据预测的负载趋势结果，识别出秘钥管理系统中各个节点中的热点节点，并判断热点节点是否存在潜在性能瓶颈
S3
根据热点节点和判断是否存在潜在性能瓶颈的节点的结果进行负载均衡策略的调整
S4

图1