

# 面向数据中心网络的分层式故障诊断算法

宣恒农<sup>1</sup>, 张润驰<sup>1</sup>, 左 苗<sup>2</sup>, 刘田田<sup>1</sup>

(1. 南京财经大学信息工程学院, 江苏南京 210046; 2. 国家计算机网络应急技术处理协调中心安徽分中心, 安徽合肥 230041)

**摘 要:** 首次提出一种面向数据中心网络的分层式故障诊断算法---HFD(Hierarchical Fault Diagnosis)算法. 算法根据数据中心网络的结构特点, 将网络系统分为两个逻辑层. 在第一层, 通过对“基本单元组”进行多次最简测试, 将其划分为“相对故障单元组”与“相对正常单元组”两大类; 在第二层, 结合本文首次提出的“正-反关联性”思想和笔者曾经提出的“绝对故障基”思想, 设计出 AD-FAFD 算法进行各单元间的相互测试与诊断. 仿真实验结果表明, HFD 算法在保持良好的诊断精确度的同时, 有效降低了时间复杂度.

**关键词:** 系统级故障诊断; 数据中心网络; HFD 算法; 正-反关联性; FAFD 算法

**中图分类号:** TP393, TP301 **文献标识码:** A **文章编号:** 0372-2112 (2014)12-2536-07

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2014.12.029

## A Hierarchical Fault Diagnosis Algorithm for Data Center Networks

XUAN Heng-nong<sup>1</sup>, ZHANG Run-chi<sup>1</sup>, ZUO Miao<sup>2</sup>, LIU Tian-tian<sup>1</sup>

(1. School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, Jiangsu 210046, China;

2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Hefei, Anhui 230041, China)

**Abstract:** We first propose the hierarchical fault diagnosis algorithm-HFD algorithm for data center network. According to the structural characteristics of data center network, the algorithm divides the network system into two logical layers. On the first layer, we perform multiple simplest tests on the "basic unit groups", which will be divided into "relative fault unit groups" and "relative fault-free unit groups". On the second layer, combining the "positive-negative correlation" idea firstly proposed in this paper and the "absolute fault base" idea previously proposed by the authors, we design the AD-FAFD algorithm for mutual test and diagnosis among units. The simulation result shows that HFD algorithm reduces the time complexity effectively while keeping good diagnosis accuracy.

**Key words:** system-level fault diagnosis; data center network; HFD algorithm; positive-negative correlation; FAFD algorithm

## 1 引言

数据中心是计算机领域近年来最为热门的研究方向之一<sup>[1,2]</sup>. 当前研究较为成熟的数据中心网络拓扑结构主要包括 Tree 结构、Fat-tree 结构、Bcube 结构、DCell 结构、FiConn 结构等, 这些结构都采用了模块化、层次化、扁平化的设计思路与虚拟化的分割管理技术<sup>[3,4]</sup>, 将上万个设备单元划分成诸多“基本功能单元组”(以下简称“基本单元组”), 以分层、递归式的拓扑结构相连接. 网络的高连通度避免了存在过多的所谓“关键节点”, 由此带来的良好冗余、容错性使得少数故障单元未被检测出不至于影响数据中心的整体运行. 然而过分的

冗余大大降低了数据中心设备的使用效率, 据 IBM 分析, 当前全球范围内数据中心网络服务器的最高使用效率保守估计不会超过 30%. 网络中的设备单元出现故障在所难免, 为了适度降低冗余, 提高数据中心设备的使用效率, 应当及时对网络进行故障检测、诊断和修复. 考虑到目前的技术水平完全允许网络中的单元进行相互通讯, 从节省诊断时间及资源消耗的角度出发, 我们可以让系统中的单元相互测试, 通过一定的诊断算法对测试结果进行分析, 从而找出故障单元. 于是, 一系列通过测试、诊断来确定系统中故障单元分布的系统级故障诊断方法, 成为保证整个系统安全与稳定的重要手段.

人们对系统级故障诊断的研究是建立在故障模型

基础之上的. 根据 PMC (Preparata-Metze-Chien) 模型的定义<sup>[5]</sup>, 任意一对互测单元的测试结果根据双方的故障状态可以分为四种情况: 正常单元测试正常单元, 其结果必为正常; 正常单元测试故障单元, 其结果必为故障; 故障单元测试正常单元或故障单元, 其结果可能为正常, 亦可能为故障. 设存在一个由有限个单元组成的系统  $X = \{x_1, x_2 \cdots x_n\}$ ,  $\forall x_i \in X$ , 令  $x_i = 0$  表示  $x_i$  正常,  $x_i = 1$  表示  $x_i$  有故障. 令  $w_{ij} = 0$  表示  $x_i$  测试  $x_j$  结果为正常,  $w_{ij} = 1$  表示  $x_i$  测试  $x_j$  结果为故障. 将全部这样的  $x_i$  以  $i$  为行号、 $j$  为列号的规则存储在一个  $n$  行  $n$  列的矩阵  $A$  中, 称矩阵  $A$  为系统  $X$  的一个测试症候矩阵. 系统级故障诊断算法按照主要诊断工具的选取可以分为三大类: 基于图论的诊断算法<sup>[5]</sup>、基于方程的诊断算法<sup>[6,7]</sup>和基于群体智能算法的诊断算法<sup>[8-10]</sup>. 其中文献<sup>[10]</sup>提出的基于萤火虫算法<sup>[11]</sup>的 FAFD 算法, 与其它群体智能诊断算法相比, 具有迭代次数少、收敛速度快、不易陷入局部最优解等特点, 且在单元数目较少的小型系统中诊断效果尤为显著.

目前, 系统级故障诊断的理论和方法并未被广泛应用于数据中心网络, 主要是由于尚无专门针对数据中心网络结构特点的系统级故障诊断算法. 若将现有的各类诊断算法直接应用于数据中心网络, 往往会因网络中单元数过多导致诊断时间过高. 当前数据仔心网络结构的最大特点在于分层式布局, 若干个网络单元在功能与逻辑上组成一个基本单元组, 作为任务调度与分配的主体. 基本单元组中的单元存在故障表现为该基本单元组存在故障, 同时基本单元组的数目远远少于系统中总单元数目. 本文基于 PMC 模型, 研究出一种适用于当代数据中心网络的分层式诊断算法——HFD (Hierarchical Fault Diagnosis) 算法. HFD 算法的主要思想是: 先在第一层 (由模块化的各基本单元组组成的外层), 由网络拓扑结构生成涵盖所有单元的最小生成树  $X$ -tree, 按其路径组织多次测试. 根据测试结果, 求出各基本单元组的故障概率, 依此将系统中所有单元组划分为存在故障概率较大的“相对故障单元组”与存在故障概率非常小的“相对正常单元组”两类; 后在第二层 (在相对故障单元组内部由各单元组成的内层), 考虑到每个基本单元组内设备单元数量一般不超过百台且连通度较高<sup>[12]</sup>, 选择一种适合小型网络故障诊断的 FAFD 算法进行改进, 提出 AD-FAFD 算法, 用于该层的诊断.

## 2 面向数据中心网络的 HFD 算法

### 2.1 第一层的测试与诊断算法

为了进行数据中心网络的第一层测试与诊断, 我

们首先给出如下定义:

**定义 1** 如果在某个时间段  $T$  内, 所有基本单元组  $x_i \in X (i \in \{1, 2, \dots, n\})$  都完成了给定的测试任务且均只执行一次, 则称时间段  $T$  为  $X$  上的一个测试周期. 若干个测试周期按测试的先后顺序排序, 组成一个测试序列  $T_m$ .

考虑到数据中心的庞大规模与良好的冗余、容错性, 对于因设备老化、硬件损坏等原因引起的局部性故障, 由于故障基本单元组数目较少, 一般不会对整个数据中心造成太大的功能影响, 这类故障源的测试与诊断可在较长的测试周期中完成. 在此期间, 任一初始为正常的基本单元组  $x_i$  可能一直保持正常, 亦可能随时出现故障. 为了既符合实际状况又简化模型, 我们作出假设: 在测试过程中, 任一故障基本单元组  $x_i$  不会在无外界干预的情况下, 自动恢复成正常基本单元组.

设  $p$  表示任一故障基本单元组测试其它基本单元组后给出正确判断的概率 ( $p \in (0, 1)$ ).  $\forall i, j \in \{1, 2, \dots, n\}$ , 当  $x_i$  与  $x_j$  相邻时, 用  $E(i, j)$  表示基本单元组  $x_i$  相对于  $x_j$  为故障基本单元组的概率 (简称  $x_i$  相对于  $x_j$  的故障概率),  $T_{ij}^{(k)}$  表示在第  $k$  个测试周期内基本单元组  $x_i$  对  $x_j$  的测试结果 (0 或 1),  $T\text{-final}_{ij}^{(m)}$  表示  $m$  个测试周期的平均结果  $\frac{1}{m} \sum_{k=1}^m T_{ij}^{(k)}$ , 则  $T\text{-final}_{ij}^{(m)}$  具有如下性质:

**性质 1**  $T\text{-final}_{ij}^{(m)} = 0$  的充要条件是  $T_{ij}^{(k)} = 0 (\forall k \in \{1, 2, \dots, m\})$ .

**性质 2**  $T\text{-final}_{ij}^{(m)} = 1$  的充要条件是  $T_{ij}^{(k)} = 1 (\forall k \in \{1, 2, \dots, m\})$ .

**性质 3**  $T\text{-final}_{ij}^{(m)} \in (0, 1)$  的充要条件是  $\exists p, q \in \{1, 2, \dots, m\}$ , st.  $T_{ij}^{(p)} \neq T_{ij}^{(q)}$ .

#### 2.1.1 基本单元组故障概率的计算方法

按测试生成树  $X$ -tree 进行测试之后, 先对其中每一条边  $(x_i, x_j)$  求出  $m$  个测试周期的平均测试结果  $T\text{-final}_{ij}^{(m)}$  与  $T\text{-final}_{ji}^{(m)}$ , 再分别求出  $x_i$  相对于  $x_j$  的故障概率  $E(i, j)$  和  $x_j$  相对于  $x_i$  的故障概率  $E(j, i)$ , 并对每一个基本单元组  $x_i$  算出相对于各相邻点故障概率的算术平均值, 可得到  $x_i$  的最终故障概率, 记为  $E[i]$ , 即  $E[i] = \frac{1}{\text{degree}(i)} \sum_j \{E(i, j) | x_i, x_j \text{ 在 } X\text{-tree 中相邻}\}$ , 其中  $\text{degree}(i)$  为  $x_i$  在  $X$ -tree 中的度数.

当有  $T_{ij}^{(m)} = 1, T_{ji}^{(m)} \in (0, 1)$  时, 由性质 3 知, 必存在至少一次测试结果  $T_{ji}^{(k)}$  与其余的测试结果不同,  $x_j$  必为故障基本单元组; 又  $T_{ij}^{(m)} = 1$ , 即  $x_i$  对  $x_j$  的判断始终是正确的, 若  $x_i$  实际为故障基本单元组, 则其每一次判断结果都是正确的概率为  $p^m$ , 于是我们计算出了表 1

中的第 8 行.  $T_{ij}^{(m)}$  与  $T_{ji}^{(m)}$  的其余情形可以按类似的方式分析推导, 整理后得  $x_i$  与  $x_j$  的故障概率分布情况如表 1 所示:

表 1 基本单元组故障概率表

$T\text{-final}_{ij}^{(m)}$	$T\text{-final}_{ji}^{(m)}$	$E(i)$	$E(j)$
0	0	$(1-p)^{2m}$	$(1-p)^{2m}$
0	(0,1)间	1	1
0	1	1	$p^m(1-p)^m$
(0,1)间	[0,1)间	1	1
(0,1)间	1	1	$p^m$
1	0	$p^m(1-p)^m$	1
1	(0,1)间	$p^m$	1
1	1	$1/2 + p^{2m}/2$	$1/2 + p^{2m}/2$

显然, 当  $p$  值不变时, 随着测试周期数  $m$  的增大, 实际故障基本单元组计算所得的故障概率越来越大, 实际正常单元组被误诊为故障单元组的概率越来越小. 多次测试能有效地减少因故障单元参与测试导致的测试结果稳定性与价值的降低, 提高诊断结果的可信度. 实际正常基本单元组与实际故障基本单元组之间的故障概率之差越来越大, 也有利于提升下一步基本单元组分类的精确度.

### 2.1.2 基本单元组的分类

在得到每个基本单元组的最终故障概率后, 可以选用合适的聚类划分算法, 将全部基本单元组划分为“相对正常单元组”和“相对故障单元组”两类. 可行的划分方法包括直接指定划分阈值法、 $k$ -means 聚类划分算法、基于网格的聚类划分算法等等. 为了提高划分精度, 我们先剔除绝对故障基  $F$ .

文献[6]提出了绝对故障基的概念及两种求法:

**方式 1** 对于  $x_i$  与  $x_j$  间的相互测试结果, 若  $w_{ij} = 0, w_{ji} = 1$ , 则  $x_i$  属于绝对故障基;

**方式 2** 若已知  $x_i$  属于绝对故障基, 且  $\exists x_j, \text{st. } w_{ji} = 0$ , 则  $x_j$  属于绝对故障基.

我们对方式 1 加以改进, 提出求绝对故障基的另一方法:

**方式 3** 对于任一基本单元组  $x_j$ , 若  $E[j] = 1$ , 则  $x_j$  属于绝对故障基.

方式 3 的证明暂略.

设识别出的绝对故障基为  $F$ , 通过划分被归类为故障基本单元组的集合为  $F'$ , 则将  $F \cup F'$  所含的全部基本单元组认定为可能存在故障的基本单元组, 进行第二层的测试与诊断.

### 2.1.3 算法步骤及效率分析

下面给出 HFD 算法在第一层完整的测试与诊断过

程:

**step1** 设定测试周期  $m$  与故障基本单元组正确判断的概率  $p$ .

**step2** 用 prim 算法或 kruskal 算法生成网络测试树  $X$ -tree.

**Step3** 由数据中心任务调度系统安排进行  $m$  个测试周期的测试, 并记录相应结果.

**Step4** 根据  $X$ -tree 中每一条测试边  $(x_i, x_j)$  的相互测试结果, 计算  $T\text{-final}_{ij}^{(m)}$  与  $T\text{-final}_{ji}^{(m)}$ .

**Step5** 利用表 1 对  $X$ -tree 中每一条边  $(x_i, x_j)$  求出  $E(i)$  和  $E(j)$ , 并对每一个  $x_i$  求得最终故障概率  $E[i]$ .

**Step6** 利用方式 2、3, 求出绝对故障基.

**Step7** 选用合适的聚类划分算法, 将基本单元组划分为“相对正常基本单元组”和“相对故障基本单元组”两类.

由于数据中心网络的拓扑结构相对固定(与测试周期无关), 网络拓扑结构生成树可在使用算法之前生成得到, 因此第一层诊断的时间复杂度为  $O(mn)$ . 又由于  $m$  为常数且  $m \ll n$ , 故有  $O(mn) \approx O(n)$ , 即测试与诊断可在与基本单元组个数成正比的时间内完成.

## 2.2 第二层的诊断算法——AD-FAFD 算法

在确定了相对故障基本单元组之后, 考虑到数据中心网络良好的冗余性, 将其从等待任务分配的调度队列中移出, 对这些单元组进行内部单元间的测试与诊断. 根据文献[12], 当前数据中心的每个基本单元组内基本采用全联通的拓扑结构, 且内部单元数目一般在 50 至 100 左右, 相当于一个小型系统. 文献[10]提出的基于萤火虫算法的 FAFD 算法对小型系统而言, 具有良好的自适应性、不记录历史最优值、自动划分为小种群等特点. 我们对 FAFD 算法进行改进, 提出 AD-FAFD 算法, 以便更加适合于数据中心网络.

### 2.2.1 预处理与萤火虫个体的初始化

FAFD 算法通过随机指定位点故障状态值的方式初始化萤火虫. 为了降低 FAFD 算法中萤火虫个体解空间的维数, 我们先作预处理: 按照前述方式 1 与方式 2 求出绝对故障基, 并将绝对故障单元和测试症候矩阵中含有绝对故障单元的测试边删除.

此后系统中仅剩下  $x_i \xrightarrow{0} x_j$  或  $x_i \xrightarrow{1} x_j$  两类测试边, 再进行初始化: 对  $x_i \xrightarrow{1} x_j$  测试边, 随机指定  $x_i$  (或  $x_j$ ) 为故障节点; 对  $x_i \xrightarrow{0} x_j$  测试边, 令  $x_i = x_j$  (或  $x_j = x_i$ ). 这种方法大大增加了初始萤火虫接近最终解的可能性.

### 2.2.2 关联性优化

在现有的诸多群体智能诊断算法(包括 FAFD 算

法)中,并未考虑到系统级故障诊断问题本身所包含的个体间联系,即相邻单元故障状态分布的“关联性”.在进行诸如移动、结合、位点突变等演化操作时,仅以各单元个体为独立的操作单位,相邻单元间的状态值存在相互矛盾的可能性较大.假设系统中包含如图 1 所示的拓扑结构,由于 A、B 间为 0/0 边,故二者属于同一集团<sup>[5]</sup>;而 A、C 间为 1/1 边,故 A、C 中至少有一个为故障单元.如果经过若干次的演化之后,A、B、C 三个位点上的状态值分别为 0、1、0,则这时候 A、B 间和 A、C 间的状态是相互矛盾的.此时继续进行下一次演化,很容易导致迭代过程在某个局部位置反复震荡,增加了算法的迭代次数与运行时间.

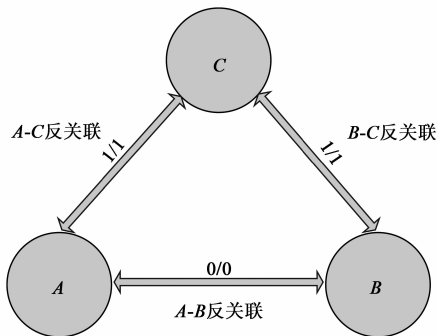


图1 拓扑结构示例

为此,我们在每一次迭代后对萤火虫个体中位点的故障状态进行正、反关联性检测,对不满足正、反关联条件的故障状态分布进行适度的“矫正”,以期减少无意义的操作,加速迭代进程.为了不过多的增加时间复杂度,在每一代萤火虫中,我们对单元的关联性仅优化一次,剩余不符合关联性的故障状态分布,通过算法的后续迭代及累次优化得到解决.所谓关联性的定义、性质及其优化过程如下:

**定义 2** 对于任一单元  $x_i$ ,若存在相邻单元  $x_j$ ,使得  $x_i \xrightarrow{0} x_j$ ,即  $x_i$  与  $x_j$  属于同一集团,则称  $x_i$  与  $x_j$  互为正关联单元;若存在与之相连的单元  $x_k$ ,有  $x_i \xrightarrow{1} x_k$ ,则称  $x_i$  与  $x_k$  互为反关联单元.

正关联单元的性质:互为正关联的任意两单元,故障状态相同.

反关联单元的性质:在互为反关联的任意两单元中,至少有一个为故障单元.

关联性优化过程见算法 1.

#### 算法 1 关联性优化算法

**Step1** 从队列 Queue 中取出首元素  $x_i$ .

**Step2** 检查每一个与  $x_i$  相邻的单元  $x_j$  及测试边,若不符合正关

联性条件,则置  $x_j = x_i$ ;若不符合反关联性条件,则置  $x_j = 1$ .

**Step3** 若队列非空,则回到 step1;否则,算法结束.

在全联通网络中,上述优化算法的时间复杂度介于  $O(n)$  至  $O(n^2)$  之间,具体取决于每一代萤火虫个体与上一代相比状态变化位点的个数,最坏情况下,该算法的时间复杂度与萤火虫算法的时间复杂度属于同一级别.

#### 2.2.3 AD-FAFD 算法的描述

AD-FAFD 的全部过程见算法 2

#### 算法 2 AD-FAFD 算法

输入:故障症候矩阵  $\sigma_F$ .

输出:符合故障症候的萤火虫  $\vec{X}_i^*$ .

**Step1** 初始化各参数,迭代次数  $\text{iteration} \leftarrow 0$ .

**Step2** 根据故障症候矩阵  $\sigma_F$ ,剔除绝对故障基,得到无绝对故障单元的症候矩阵  $\sigma_{F'}$ ,令  $\sigma_F = \sigma_{F'}$ .

**Step3** 初始化  $M$  只萤火虫,  $\text{iteration} \leftarrow \text{iteration} + 1$ .

**Step4** 计算每只萤火虫的适应度.

**Step5** 找出适应度最高的萤火虫  $x_i$  及其对应的适应度  $I(x_i)$ ;若  $I(x_i) = 1$ ,则算法结束.

**Step6** 计算每只萤火虫的自适应光吸收因子  $\gamma_{\max}(x_i)$ ,根据吸引度移动非最亮的萤火虫,最亮的萤火虫  $x_i$  随机移动.

**Step7** 对每只萤火虫进行二进制映射,根据 2.2.2 节的关联性优化算法,调整  $\vec{X}_i$  的故障状态关联性分布值.

**Step8** 如果故障单元数大于  $t$ ,处理不可行解<sup>[10]</sup>.

**Step9** 如果  $\text{iteration} \geq \text{maxIter}$ ,算法结束;否则,返回 Step4.

### 3 实验仿真与分析

我们分别对 HFD 算法中第一层的测试与诊断算法以及第二层的 AD-FAFD 算法进行了实验仿真,仿真实验在一台联想计算机 (Intel Core i5-2400 3.1GHz CPU, 4GB 内存) 上于 MATLAB R2012b 环境下实现.

#### 3.1 第一层测试与诊断算法的实验与分析

鉴于当前实际数据中心网络第一层所含的基本单元组数目较多,我们分别在随机生成的含有 1000、10000 个基本单元组的网络中进行第一层测试与诊断算法的实验仿真.实验首先随机指定  $k(k \in \{1, 2, \dots, t\})$  个基本单元组为故障单元组,之后根据网络结构生成最小测试树,进行  $m$  个测试周期的测试,根据平均测试结果计算出各基本单元组为故障单元组的概率,选择  $k$ -means 聚类划分算法进行基本单元组的划分(故障单元组的初始簇中心为 1,正常单元组的初始簇中心为 0,距离计算采用欧几里德计算方法),在不同的  $k$  和  $m$  取值下,对诊断结果的正确率进行了研究,其结果分别如图 2、图 3 所示:

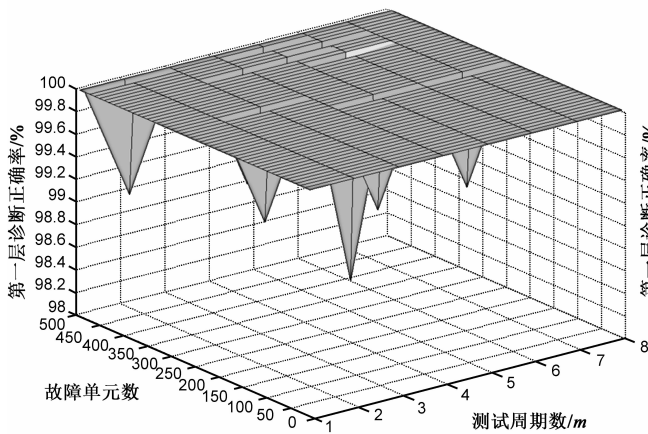


图2 在含有1000个基本单元组的第一层网络中算法的诊断正确率

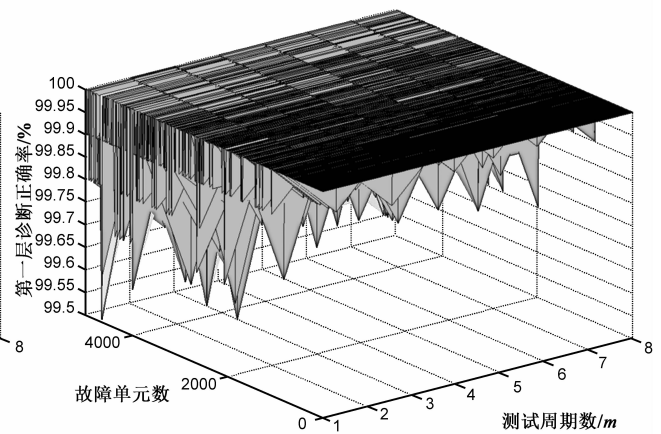


图3 在含有10000个基本单元组的第一层网络中算法的诊断正确率

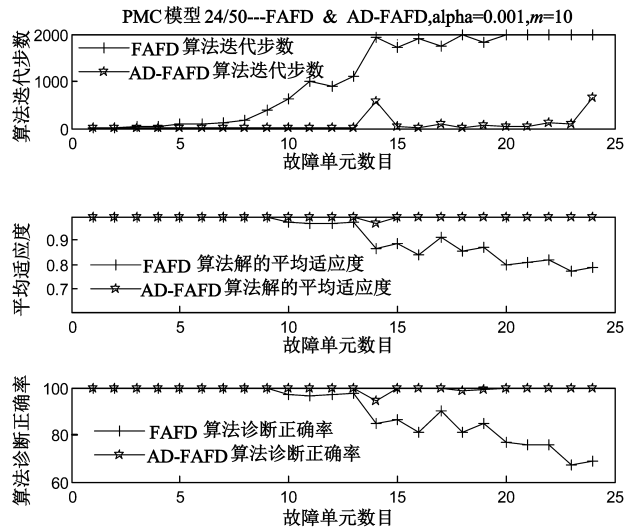


图4 在含有50个单元的网络中两种算法的比较

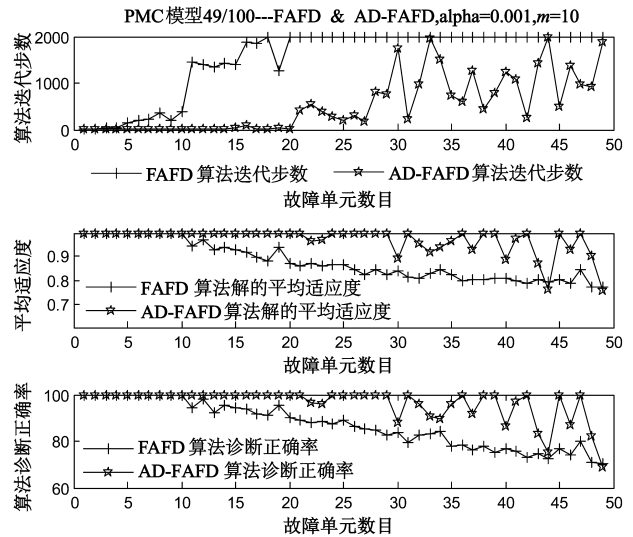


图5 在含有100个单元的网络中两种算法的比较

实验结果表明:

(1)在同一个网络中,随着故障单元数目的上升,第一层测试与诊断算法的诊断正确率下降十分缓慢,始终接近于100%,即使当故障单元数目接近系统总单元数目一半时,算法依然能够保证98%以上的诊断正确率,故障单元数目对算法的诊断正确率影响不大。

(2)在不同网络中,随着系统总单元数目的提高,第一层测试与诊断算法的诊断正确率并未出现明显变化,系统总单元数目对算法的诊断正确率影响不大。

(3)在网络单元数与其中故障单元数不变的情况下,测试周期数 $m$ 越大,诊断的正确率越高,当 $m > 5$ 时,算法的诊断正确率趋近于100%。

### 3.2 第二层 AD-FAFD 算法的实验与分析

我们对 AD-FAFD 算法与 FAFD 算法进行比较实验仿真。为了便于比较,AD-FAFD 算法的参数设置与 FAFD 算法的参数设置完全相同,均完全采用文献[10]

中的设置:萤火虫个体数  $M = 10$ ,最大迭代步  $\maxIter = 2000$ ,随机漫步因子  $\alpha = 10^{-3}$ ,初始光吸收因子  $\gamma_0 = 1$ ,萤火虫自身亮度参数  $\beta_0 = 1$ 。考虑到数据中心网络中基本单元组内的单元数普遍较少,分别对每个基本单元组含有50、100个单元,且故障单元数目符合 $t$ 可诊断性的小型网络进行实验,从1至 $t$ 每次生成不同的故障单元数目,通过模拟测试得到对应的测试报告,两个算法根据同一份测试报告进行诊断,在同一个故障单元数目下重复进行50次。对两种算法的迭代步数、解的平均适应度与诊断正确率分别进行了比较,结果如图4、图5所示:

实验结果表明:

(1)在同一个网络中,随着故障单元数目的上升,FAFD 算法迭代次数增长较快,很快达到2000次的最大限制,而 AD-FAFD 算法的迭代次数增加相对缓慢;FAFD 算法所得解的适应度逐渐下降(表明更多的诊断

过程是由于达到最大迭代次数的限制而停止,而非得到了适应度为 1 的个体),而 AD-FAFD 算法的解的适应度基本保持在 1 左右;两个算法的诊断正确率均随着故障单元数目的上升而下降,但 AD-FAFD 算法的下降趋势明显缓于 FAFD 算法。

(2)在不同网络中,随着系统总单元数目的提高,FAFD 算法的迭代次数上升明显快于 AD-FAFD 算法;AD-FAFD 算法的解的适应度基本保持在 1 左右,而 FAFD 算法所得解的适应度逐渐下降。同时,AD-FAFD 算法的诊断正确率始终高于 FAFD 算法。系统总单元数目越多,故障单元数目越多,这种优势就愈发明显。

综上,HFD 算法在第一层的诊断正确率受网络总单元数、故障单元数的影响较小,当测试周期数较大时,其正确率始终接近 100%。第二层的 AD-FAFD 算法在迭代次数、解的适应度、诊断正确率方面均优于 FAFD 算法。

## 4 总结与展望

本文提出了面向数据中心网络的分层式故障诊断算法-HFD 算法。算法充分结合当前数据中心网络的分层式结构特点,提出了分层诊断思路,为两层设计不同的算法进行诊断。在第一层,以测试生成树为依据组织基本单元组间的最简测试,通过多次测试提高测试结果的可信度,根据测试结果推导出各基本单元组为故障单元组的概率,最后用聚类算法划分出各故障单元组;在第二层,结合其系统规模较小的特征与提出的单元间故障状态分布的关联性思想,设计出 AD-FAFD 算法对故障单元组的内部单元进行诊断。

与现有的各类诊断算法相比,HFD 算法大大减少了网络中整个测试与诊断阶段参与的单元数目,降低了网络因测试任务而承载的负荷,在不失诊断精度的前提下,大幅度削减了诊断的时间复杂度。同时,HFD 算法不仅仅适用于当前的数据中心网络,对绝大多数具有分层式结构特征的系统,HFD 算法亦可应用。

本文对 HFD 算法按 PMC 故障模型叙述,对于其它故障模型,只要根据模型的特性,将算法进行相应的调整,亦可适用,因篇幅所限,不再赘述。此外,我们在 2.2 节对 FAFD 算法做出的三点改进,是否适用基于其它群体智能诊断算法,还有待进一步研究。

## 参考文献

[1] Bari M F, Boutaba R, Esteves R, et al. Data center network virtualization: A survey[J]. Communications Surveys & Tutorials, IEEE, 2013, 15(2): 909 - 928.

[2] Curtis A R, Carpenter T, Elsheikh M, et al. REWIRE: An optimization-based framework for unstructured data center network

design[A]. INFOCOM, 2012 Proceedings IEEE [C]. Orlando, FL: IEEE, 2012. 1116 - 1124.

[3] Rosa R V, Rothenberg C E, Madeira E. Virtual data center networks embedding through Software Defined Networking[A]. Network Operations and Management Symposium (NOMS), 2014 IEEE[C]. Krakow: IEEE, 2014. 1 - 5.

[4] 魏祥麟, 陈鸣, 范建华, 张国敏, 卢紫毅. 数据中心网络的体系结构[J]. 软件学报, 2013, 24(02): 295 - 316.

Wei Xianglin, Chen Ming, Fan Jianhua, Zhang Guomin, Lu Ziyi. Data center network architecture [J]. Journal of Software, 2013, 24(02): 295 - 316. (in Chinese)

[5] 张大方, 江招生. 基于集团的系统级故障诊断研究[J]. 计算机学报, 1998, 21(04): 308 - 314.

Zhang Dafang, Jiang Zhaosheng. Node grouping in system-level fault diagnosis[J]. Journal of Electronic, 1998, 21(04): 474 - 479. (in Chinese)

[6] 宣恒农, 张大方, 张明. PMC 故障模型的方程诊断[J]. 电子学报, 2003, 31(05): 694 - 697.

Xuan Hengnong, Zhang Dafang, Zhang Ming. PMC fault model diagnostic equation [J]. Journal of Electronic, 2003, 31(05): 694 - 697. (in Chinese)

[7] 宣恒农, 韩忠愿, 张大方. 基于互测 PMC 模型的故障诊断方法及其应用[J]. 电子学报, 2007, 35(05): 987 - 990.

Xuan Hengnong, Han Zhongyuan, Zhang Dafang. PMC Model-based fault diagnosis method and its application[J]. Journal of Electronic, 2007, 35(05): 987 - 990. (in Chinese)

[8] Falcon R, Almeida M, Nayak A. A binaryparticle swarm optimization approach to fault diagnosis in parallel and distributed systems[A]. Evolutionary Computation (CEC), 2010 IEEE Congress on [C]. Barcelona: IEEE, 2010. 1 - 8.

[9] Elhadeif, M. Solving the PMC-based system-level fault diagnosis problem using hopfield neural networks [A]. Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on [C]. Biopolis: IEEE, 2011. 216 - 223.

[10] Falcon R, Almeida M, Nayak A. Fault identification with binary adaptive fireflies in parallel and distributed systems[A]. Evolutionary Computation (CEC), 2011 IEEE Congress on. IEEE [C]. New Orleans, LA: IEEE, 2011. 1359 - 1366.

[11] X S Yang. Firefly algorithms for multimodal optimization [A]. Stochastic Algorithms: Foundations and Applications [C]. Sapporo, Japan: Springer, 2009. 169 - 178.

[12] 王聪, 王翠荣, 王兴伟, 蒋定德. 面向云计算的数据中心网络体系结构设计[J]. 计算机研究与发展, 2012, 49(2): 286 - 293.

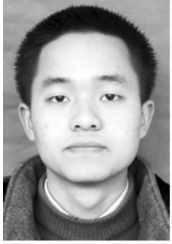
Wang Cong, Wang Cuirong, Wang Xingwei, Jiang Dingde. Oriented cloud computing data center network architecture design [J]. Computer Research and Development, 2012, 49(2): 286 - 293. (in Chinese)

## 作者简介



**宣恒农** 男,1958年11月出生,江苏淮安人,二级教授,现为南京财经大学信息工程学院副院长,江苏省软件人才培养教指委委员,江苏省重点学科负责人.主要从事系统级故障诊断、未来网络、符号计算等方面的研究工作.

E-mail: hnxuan@21cn.com



**张润驰** 男,1990年3月出生,江苏南京人,现为南京财经大学信息工程学院在读硕士研究生,主要从事系统级故障诊断、数据挖掘、量化交易方向的学习和研究.

E-mail: zhangrunchi@aliyun.com

**左苗** 男,1989年1月出生,安徽合肥人,2013年毕业于南京财经大学,获工学硕士学位.现在国家计算机网络应急技术处理协调中心安徽分中心工作,主要从事网络系统故障诊断等方向的研究.

E-mail: 675641679@qq.com

**刘田田** 女,1990年6月出生,山东济宁人,现为南京财经大学信息工程学院在读硕士研究生,主要从事系统级故障诊断、未来网络等方向的学习和研究.

E-mail: liutianliurong@163.com