# 用双目标加权遗传算法解决网络磁盘阵列系统下校验散布布局优化问题的研究*)

杨 敏 王 刚 刘 璟 陈北莲
(南开大学计算机科学与技术系 天津300071)

**摘 要** 廉价磁盘冗余阵列(RAID)作为一种提高存储系统可靠性和性能的技术,已经得到了广泛的应用,有关磁盘阵列结构和数据布局的研究也一直很活跃,但有关网络磁盘阵列下的数据布局的研究还不太多。本文首先概述了校验散布布局的技术和遗传算法的相关知识,提出了利用双目标加权遗传算法的思想解决网络磁盘阵列系统校验散布布局优化的问题。然后以"重构负载均匀分布"和"校验均匀分布"为双目标,使用改变的NSGA来解决网络磁盘阵列系统下校验散布布局的优化问题。最后给出了实验结果。

**关键词** 廉价磁盘冗余阵列,数据布局,校验散布,Pareto 占优,Niching 技术,多目标目标遗传算法,加权校验散布布局

## Solving Parity Declustering Optimal Problems in Network Based RAID by Using Two-Weighted-Objective Genetic Algorithms

YANG Min WANG Gang LIU Jing CHEN Bei-Lian
(Department of Computer Science and Technology, Nankai University, Tianjin 300071)

**Abstract** RAID, a technology to improve the reliability and performance of storage system, has been got a widely using. Although the researchers about the structure of RAID and data layout are very popular, the researchers about the data layout of network based RAID are little. In this paper, first, we mainly introduce some knowledge about parity declustering and genetic algorithms, present a new thought to solving parity declustering optimal problems in the network based RAID by two-weighted-objective genetic algorithms. Then we take "Balance reconstruction workload" and "Balanced parity overhead"as two objectives, used the changed NSGA to solving parity declustering optimal problems in the network based RAID system. At last we describe the experimental results.

**Keywords** RAID, Data layout, Parity declustering, Pareto optimal, Niching, Multiobjective genetic algorithms, Parity declustering layout with weight

## 1 引言

Patterson 等人于20世纪80年代末提出的廉价磁盘冗余阵列(RAID)技术,因其具有磁盘容量大、数据传输率高和磁盘可靠性高等特点,得到了广泛的研究和发展。Muntz 和 Lui 为了解决 RAID5在故障状态下性能较差的问题提出了校验散布布局技术,后来 Holland 和 Gibson 对其进行了详细的研究。校验散布局通过校验条纹的散布,重构负载的均匀分布,改进了 RAID5结构降级模式下和重构模式下性能差的缺点,提高了磁盘阵列系统的可用性。由于同时加快了磁盘重构过程的速度,校验散布布局还提高了磁盘阵列系统的可靠性。同时 Holland 等人还提出了评价校验散布布局的六条标准,满足这些标准的布局被称为理想布局。但 Alvarez 等人已经证明,构造理想布局是非常困难的,只在极少数参数下才存在理想布局。在校验散布布局方法的研究工作中,都是试图构造更为接近理想的布局,即尽量满足更多的标准,不能满足的标准也尽量接近最优。校验散布布局方法的研究工作有很多,如,基于平衡不完全区组设计(BIBD)的布局方法, PRIME 和 RELPR[4], PDDL 布局方法[5],随机排列布局[6]等。但这些校验散布布局技术依据的标准侧重不一、复杂程度不同,而且在不同的参数/条件的选取以及实际情况下其性能各有优劣。南开大学软件实验室提出了一种新的思路来解决校验散布布局优化问题:使用多目标遗传算法,来解决校验散布布局优化问题[1]。

另外上面提到的校验散布布局技术都是只针对单机磁盘阵列系统,没有涉及网络磁盘阵列系统。对于单机磁盘阵列系统来说,所有的磁盘设备都是本地连接,可以直接访问。而对于网络磁盘阵列来说,组成磁盘阵列的磁盘设备,一部分是本地连接的,可以直接访问。但另一部分则位于其它节点甚至不同的地理位置,不能直接进行访问,必须使用网络存储访问协议(如 NBD、iSCSI 等),经由网络设备进行访问。而访问网络磁盘,除包括访问本地磁盘所需开销外,还包括网络协议、网络传输时间、网络延迟等开销,并且这些开销较大,与磁盘操作时间相比不能忽略。针对这些问题,南开大学软件实验室提出了"加权校验散布布局方法"来解决网络磁盘阵列系统下的校验散布布局的优化问题[2,3]。

本文继续南开大学软件实验室的工作,提出了使用双目标加权遗传算法解决网络磁盘阵列系统下的校验散布布局的优化问题。

*)本文得到国家自然科学基金项目的资助(项目编号60273031)和高校博士学科点专项科研基金项目的资助(项目编号20020055021)。杨 敏 硕士研究生,主要研究方向为并行技术及应用。王 刚 讲师,博士,主要研究方向为网络存储、并行与分布式系统。刘 璟 教授,博士生导师,主要研究方向为并行与分布式系统、算法分析、VISL 等。陈北莲 硕士研究生,主要研究方向为并行技术及应用。

## 2 相关理论基础介绍

### 2.1 多目标优化和多目标遗传算法的基础知识

多目标优化问题是指在求解一个问题时存在多于一个的目标函数，并且各目标函数之间往往相互冲突，如果求最优解，就应该综合考虑这多个目标函数。多目标优化问题中的目标矢量具有一种偏序关系，相应的其解集合上也具有一种偏序关系。如果解 $A$ 的所有目标函数值都优于或相同于解 $B$，则称解 $A$ 优于解 $B$，解 $B$ 劣于解 $A$。这种关系我们称之为 Pareto 占优关系。所有满足特定约束条件的所有解集合，称为可行解集合 $F$。解 $a$ 如果不劣于 $F$ 中的任何解，则称 $a$ 是 Pareto 最优。由 $F$ 中所有 Pareto 最优解构成的集合称为 Pareto 最优集合。

遗传算法由 Bagley J. D. 在1967年提出，主要的思想就是通过模拟生物界“适者生存”和染色体遗传变异的过程来选择出最优的解个体，其涉及的操作主要包括：选择、交叉和变异。伴随着遗传算法的发展，1984年 Shaffer 首次将其引入了多目标优化问题中，这就出现了所谓的“多目标遗传算法”(MOGA)。适用于多目标优化问题的 MOGA 是在简单遗传算法的基础上修改得到的，它主要是在适应度分配策略上不同于简单遗传算法。

Niche 原来是生物学的一个术语，其本意是指在一个局部的小环境中，小群体相互间争夺有限的资源，形成具有自己特色、区别于其它小群体的一些特征，从而在一个全局的生态环境中保持物种的多样性。实现 Niche 的技术，即所谓的 Niching 即是借鉴这种思想提出的一种保持群体多样性的有效办法。在单目标遗传算法中主要用于维持群体的多样性，以避免陷入局部最优解。而在多目标遗传算法中，Niching 首先被用于偏序关系全序化，或者在两个解不可比时借 Niching 做出选择。同样，当需要对一个数量过多的非劣解集合进行过滤时，往往也是借助 Niching 决定要剔除的解。实现 Niching 的最典型的两个方法有 Goldberg 提出的适应值共享(fitness sharing)和 De Jong 提出的拥挤策略(crowing)。

### 2.2 双目标遗传算法解决校验散布布局优化问题

南开大学软件实验室提出的用双目标遗传算法解决校验散布布局优化问题，采用 Srinivas 和 Deb 提出的 Non-dominated Sorting Genetic Algorithm(NSGA)[10]这一多目标遗传算法的思想，来解决校验散布布局优化问题。考虑到理想布局六条标准[9]中的标准2“重构负载均匀分布”是校验散布技术的核心，标准3“校验均匀分布”对性能影响也较大，因此以这两条标准作为 NSGA 的双目标，来对随机排列布局进行优化。NSGA 算法通过将目标空间上的偏序关系转换为决策空间上的 Pareto 占优关系，对群体中的个体划分出 non-domination 等级，再利用 Niching 技术对同一级别的不同个体赋予不同的共享适应值，接下来采用简单遗传算法那样的选择、杂交和变异操作。重复上面的步骤 $n$ 代之后，得到一个 Pareto 最优解集，从而可以进一步挑选出最优的校验散布布局。用 NSGA 优化随机排列布局的方法对布局参数没有任何限制，适用范围最广。而且校验散布布局经过优化后，小规模的布局也可以达到很好的性能，更加符合“高效映射”(理想布局标准4)的要求，这种方法具有很强的实用价值。

### 2.3 加权校验散布布局方法

南开大学软件实验室提出的“加权校验散布布局方法”其基本思想是，为处于不同节点的磁盘赋予相应的权值，将原来使用的重构负载度量函数进行修改，从 $\sum_{L\text{中所有磁盘对}i,j}(X_{i,j})^2$，变化为 $\sum_{L\text{中所有磁盘对}i,j}(X_{i,j}\cdot e_{i,j})^2$。在初始生成随机的校验散布布局之后，利用模拟退火的思想生成新的布局，重复执行多次，最终实现加权校验散布布局。通过具体的实验结果可以看出，该方法能够很好地解决分布式磁盘阵列系统下的校验散布布局问题。

## 3 用双目标加权遗传算法解决网络磁盘阵列系统下的校验散布布局的优化问题

### 3.1 基本思想

双目标遗传算法解决校验散布布局优化问题，综合考虑了标准2和标准3，可以在两者都能达到最优的情况下实现校验散布布局，但其只考虑了本地磁盘阵列系统，这具有很大的局限性。加权校验散布布局方法解决了网络磁盘阵列系统下的校验散布布局，但只选取了一个目标函数。因此，本文吸收两者的思想，将它们相融合，提出了用双目标加权遗传算法解决磁盘网络阵列系统下的校验散布布局的优化问题。在该算法中，综合考虑标准2和标准3，使用加权校验散布方法提出的重构负载度量函数和校验分布度量函数作为进行 Pareto 占有关系划分的标准。然后利用变化了的 NSGA 的思想，经过多代的交叉、变异操作，获得网络磁盘阵列系统下较优的校验散布布局。

### 3.2 具体实现

#### 3.2.1 目标函数的选取

(1) 权值的选取

这里权值的选取，参考文[2,3]中提到的权值确定方法：

·对于本地磁盘阵列，所有 $e_{i,j}$ 的值都取1；

·对于类似 NRAID 的具有中心节点的网络磁盘阵列系统，磁盘位于不同的节点，但磁盘阵列系统软件只运行于中心节点，重构进程也运行于中心节点，与故障磁盘的位置是无关的。假定节点0为中心节点，则具有中心节点的网络磁盘阵列系统中 $e_{i,j}$ 的取值可表示为：

$$e_{i,j}=\begin{cases}1,(j \bmod m)=0\\ e,(j \bmod m)\neq 0\end{cases}$$

·对于类似 Petal 的分布式网络磁盘阵列系统，重构进程可运行于不同节点。因此，如果磁盘 $j$ 与磁盘 $i$ 位于同一个节点(也就是重构进程所在节点)，则 $e_{i,j}$ 的取值为1，否则，$e_{i,j}$ 的取值为 $e$。

$$e_{i,j}=\begin{cases}1,(j \bmod m)=(i \bmod m)\\ e,(j \bmod m)\neq(i \bmod m)\end{cases}$$

(2) 重构负载分布度量函数

$$H_{WEIGHTEL}(L)=\sum_{L\text{中所有磁盘对}j}(X_{i,j}\cdot e_{i,j})^2$$

这里 $X_{i,j}$ 表示当磁盘 $i$ 出现故障时需要从磁盘 $j$ 上读取的数据单数数目；$e_{i,j}$ 表示当磁盘 $i$ 出现故障时从磁盘 $j$ 上读取一个数据单元需要的代价，即权值。显然，$X_{i,j}$ 趋向于均值即重构负载均匀分布，以及 $X_{i,j}\cdot e_{i,j}$ 即重构总代价减小，这两方面的因素都会使目标函数值减小。因此，最小化目标函数的过程，可能会使布局向着重构负载均匀分布和重构总代价减小两个方面发展，从而使网络磁盘阵列的重构性能得到优化。

(3) 校验分布度量函数

$$P_{WEIGHTED}(L)=\sum_{L\text{中所有磁盘}i}P_i^2$$

这里 $P_i$ 表示第 $i$ 个磁盘上的校验单元数。显然，使所有 $P_i$ 的值均趋于平均值，即最小化目标函数，可以使布局的校验分布趋于均匀。

3.2.2 染色体方案的表示 每个布局以二维 $r\times v$ 矩阵的形式存储，行表示 $r$ 个布局行，列表示 $v$ 个物理磁盘；每行

共有 $v$ 个元，代表 $v/k$ 个不同条纹；各个元的绝对值代表条纹序号，负数元代表该条纹中的校验单元。

下面给出一个布局的例子(磁盘数 $v=6$，条纹长度 $k=3$，布局行数 $r=5$)：

| | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| r1 | −1 | 2 | −2 | 1 | 2 | 1 |
| r2 | 4 | 3 | 4 | −3 | −4 | 3 |
| r3 | −5 | 5 | 6 | 5 | 5 | −6 |
| r4 | 7 | 8 | −8 | −7 | 8 | 7 |
| r5 | 9 | −10 | 9 | −9 | 10 | 10 |

决策空间上的 Pareto 占优关系如下：

布局 $A$ dominate 布局 $B$(即布局 $A$ 优于布局 $B$)⇔

$((H(A)\leqslant H(B))\&\&(P(A)\leqslant P(B)))\&\&((H(A)<H(B))\|(P(A)<P(B)))$

3.2.3 *布局参数和算法参数* 给定 $v$(磁盘数目)，$k$(条纹长度)，$r$(布局行数)，(这里只考虑 $v$ 是 $k$ 的整数倍)，并且可以根据 $v\cdot r=k\cdot s$，计算出 $s$(条纹数)。

杂交操作和变异操作时，均采用单点杂交的方式。为了防止进行杂交和变异操作后产生的布局不属于合法布局，这里采用的杂交和变异操作有一定的限制。对于杂交操作，是对两个不同布局的同一行进行整行的互换。而对于变异操作则是对某一布局的同行中的两个不同单元进行互换。

3.2.4 *NSGA 解决网络磁盘阵列系统下的校验散布布局的优化问题的主要步骤*：

(1)随机生成校验散布布局，布局规模为 $N$；

(2)计算出重构负载度量值和校验分布度量值；

(3)根据 Pareto 占优关系，将这 $N$ 个布局划分等级，由高到低依次为等级一、等级二、…、等级 $m$；

(4)为每个等级中的布局成员赋值：

·根据 Niching 的思想为等级一赋共享适应值；

·根据 Niching 的思想为等级二赋共享适应值；

·直到所有等级赋值完成；//低等级的共享适应值要小于高等级的共享适应值；

(5)挑选出 $N$ 个较优布局进行保存；

(6)使用轮盘赌的方法，进行选择操作；

(7)染色体(布局)之间的杂交操作；

(8)染色体(布局)内部的变异操作；//交叉、变异后生成 $N$ 个新布局；

(9)检查实际重构负载值和实际校验分布值与理论值的偏差是否均小于某一足够小的值?如果小于，则算法终止；否则，转去执行步骤(2)。

(这里，除执行第一代时划分等级和分配共享适应值的规模为 $N$，其余各代的规模均为 $2N$。但每代的选择、交叉和变异操作的规模都是为 $N$，且这 $N$ 个布局就是保存的较优布局。)

## 4 实验结果

这里列出的实验结果分为三组，第一组是本地磁盘阵列环境下的优化布局结果，第二组是具有中心节点的磁盘阵列的优化布局结果，第三组是分布式磁盘阵列的优化布局结果。

(1) 本地磁盘阵列环境下的优化布局结果。这里选取的参数是：磁盘数 $v=12$，条纹长度 $k=6$，群体规模 $N=50$。

根据文[7]中的定理可以算出，在理论最优布局中，在本地磁盘阵列环境下，对所有的磁盘对 $i$ 和 $j$，$X_{ij}$ 的值都相等，为定值 $r*(k-1)/(v-1)$。对所有磁盘 $i$，$P_i$ 的值都相等，为定值 $r/k$。表1列出了实际实验中的最大 $X_{ij}$ 值和 $P_i$ 的值。从表1可以看出，$P_i$ 值已经达到了一个实际最优结果，而 $X_{ij}$ 的值也保持一个很小的偏差。

(2) 具有中心节点的磁盘阵列的优化布局结果。这里选取条纹数 $r=117$，条纹长度 $k=10$，磁盘数 $v=40$，本地磁盘与网络磁盘速度之比设为3:1，即 $e=3$。节点数为2,4,5,8。表2列出了重构目标函数值的实验结果及最大的 $P_i$ 值，并且和文[2]中的数据进行了对比。

表1 本地磁盘阵列环境下的优化布局结果

| 布局行数 | 理论最优 $X_{ij}$ | 实际最优 $X_{ij}$ | 偏差 | 理论最优 $P_i$ | 实际最优 $P_i$ | 偏差 |
|---|---|---|---|---|---|---|
| 117 | 53.1818 | 55 | 3.309% | 19.5 | 20 | 2.25% |
| 308 | 140 | 142 | 1.428571% | 51.333 | 52 | 1.346% |
| 1121 | 509.5454 | 512 | 0.482% | 186.8333 | 187 | 0.35761% |
| 4873 | 2215 | 2219 | 0.18% | 812.16666 | 813 | 0.10254% |

表2 具有中心节点的磁盘阵列的优化布局结果对比

| 布局方法 / 节点数 | 模拟退火 | 加权校验散布 | 双目标加权遗传算法散布结果 | 理论 $P_i$ 值 | 实际最大 $P_i$ 值 |
|---|---|---|---|---|---|
| 2 | 5690480 | 5689900 | 5690048 | 12 | 12 |
| 4 | 7966670 | 7966040 | 7966340 | 12 | 12 |
| 5 | 8422050 | 8421530 | 8421200 | 12 | 13 |
| 8 | 9104840 | 9104650 | 9104620 | 12 | 13 |

表3 分布式磁盘阵列的优化布局结果

| 布局方法 / 节点数 | 模拟退火 | 加权校验散布 | 双目标加权遗传算法散布结果 | 理论 $P_i$ 值 | 实际最大 $P_i$ 值 |
|---|---|---|---|---|---|
| 2 | 5805760 | 2092270 | 2105600 | 12 | 12 |
| 4 | 8140350 | 3601790 | 3766204 | 12 | 12 |
| 5 | 8605310 | 4211840 | 4335244 | 12 | 13 |
| 8 | 9306610 | 5720940 | 5828024 | 12 | 13 |

(下转第100页)

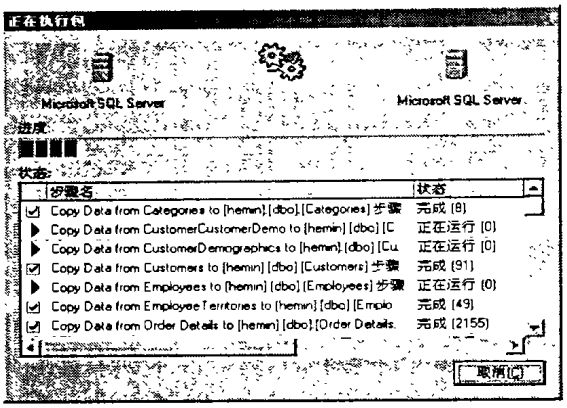


图4 DTS 包执行示意图

数据抽取成功执行后，我们就可以在数据缓存区中对数据进行转换和加载的操作。由于转换操作比较复杂，因此我们采用 DTS 包设计器来实现。包设计器也是图形化界面，与导入向导原理区别不大，主要的不同就在于转换操作可以通过 VBScript 脚本语言编程实现。在图5中选择新建按钮，在弹出的界面中输入用户编写的脚本语言即可。

**结论** 本文对数据预处理效率的提高方面进行了一定的研究，通过对原有数据仓库的体系结构下进行的预处理的不足的分析，引入了数据缓存区的概念，从而把原本复杂的数据预处理过程分解为两个阶段来实施，在一定程度上解决了原有的不足，提高了数据抽取、转换和加载的效率。但由于能力与时间的问题，对数据预处理在这种体系结构下的具体实现还缺乏更深入的研究，这也将是我们今后努力的方向。

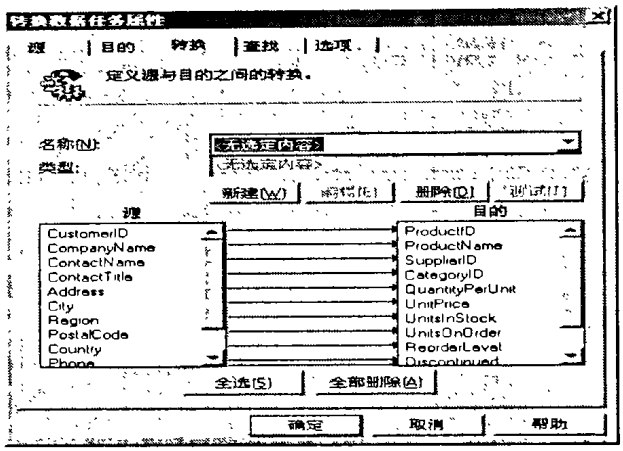


图5 转换数据任务属性

## 参 考 文 献


1 Inmon W H. Building the data warehouse [M]. Slough: QED Publishine Group, 1996
2 (美)Han Jiawei, Kamber M 著，范明，孟小峰，等译. 数据挖掘概念与技术. 北京：机械工业出版社，2001
3 Pereira E. Slash Business Intelligence Development Time and Costs by 80%. datawarehouse. ittoolbox. com, 2002
4 (美)Gunderloy M, Jorden J L 著，仲潘，等译. SQL Server 2000从入门到精通. 北京：电子工业出版社，2001


(上接第75页)

表2显示的数据相差无几，可以认为是实验中的误差，这种情况也是在预料之中。关于这一点文[2]中已有详细说明。原因是，在中心节点网络磁盘阵列中，重构进程运行于中心节点，本地磁盘和网络磁盘都是固定的，而与故障磁盘的位置无关。中心节点磁盘即为本地磁盘，其他节点磁盘为网络磁盘。如果试图使校验条纹聚集到本地磁盘，则另一部分条纹必然聚集到网络磁盘。当中心节点磁盘发生故障，则重构过程中磁盘访问多数为本地磁盘访问，性能较佳。但如果其他节点磁盘发生故障，重构过程中磁盘访问多数为网络磁盘访问，性能会很差。总体来说，布局优化过程中没有条纹聚集的趋势，条纹是完全散布的。从上表还可以看出，校验单元能够较均匀的分布到各个磁盘上。

(3)分布式磁盘阵列的优化布局结果

这里选取条纹数 $r=117$，条纹长度 $k=10$，磁盘数为 $v=40$，本地磁盘与网络磁盘速度之比设为3:1，即 $e=3$。节点数分别为2,4,5,8。表3列出了重构目标函数值的实验结果及最大的 $Pi$ 值，并且和文[2]中的数据进行了对比。

从表3可以看出使用本文中提到的双目标加权遗传算法得出的校验单元分布非常均匀，重构函数值比模拟退火得到的值要小得多，与加权校验散布的重构负载值有所差别。之所以会产生重构负载值比加权校验散布要差的原因是，本文提出的算法考虑了两个目标函数，而加权校验散布只考虑了一个目标函数。在双目标加权遗传算法只考虑重构负载目标函数时，计算的结果与加权校验散布的结果相差无几。这就说明，双目标加权遗传算法同时兼顾了重构负载值最小和校验散布均匀两个标准，较之于模拟退火和加权校验散布在分布式磁盘阵列系统环境下，更加符合理想布局的标准。

**结束语** 本文选取理想布局标准中的2和3作为双目标，将用于本地磁盘阵列系统的目标函数进行加权变化，揉合遗传算法的思想，提出了使用双目标加权遗传算法解决网络磁盘阵列系统下的校验散布布局优化的问题，并给出了实验结果。通过以上的理论分析和实际结果可以看出使用多目标遗传算法解决该问题的可行性及有效性。

现在很多人认为“遗传算法、自适应系统、细胞自动机、混沌理论与人工智能一样，都是对今后十年的计算技术有重大影响的关键技术”，多目标遗传算法的应用也已成为人们越来越关注的问题。本文提出的使用双目标加权遗传算法解决网络磁盘阵列系统下的校验散布布局优化的问题，是多目标遗传算法的又一个应用。关于该问题的研究还有很多值得深入和探讨的地方，例如，使用多目标遗传算法解决双故障容错数据布局的问题等。

## 参 考 文 献


1 董雅莉：[硕士研究生毕业(学位)论文]. 南开大学，2003
2 王刚：[博士研究生毕业(学位)论文]. 南开大学，2002
3 王刚，刘晓光，刘璟. 网络 RAID 布局研究. 计算机科学，2002(5)：11～13
4 Alvarez G A, Burkhard W A, Stockmeyer L J, Cristian F. Declustered Disk Array Architectures with Optimal and Near-Optimal Parallelism. In: Proc. of the 25th Annual ACM/IEEE Intl. Symposium on Computer Architecture, June 1998
5 Schwarz T J E, Steinberg J, Burkhard W A. Permutation Development Data Layout (PDDL) Disk Array Declustering. In: Proc. of the Fifth Intl. Symposium on High-Performance Computer Architecture, 1999. 214～217
6 Merchant A, Yu P. Design and Modeling of Clustered RAID. In: Proc. of the Intl. Symposium on Fault-Tolerant Computing, 1992. 140～149
7 Schwabe E J, Sutherland I M, Holmer B K. Evaluating Approximately Balanced Parity-Declustered Data Layouts for Disk Arrays. Parallel Computing, 1997, 23(4-5): 501～523
8 Deb K, Pratap A, Afarwal S, Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, April 2002, 6(2)
9 Holland M, Gibson G A, Sieworuk D P. Architectures and Algorithms for On-Line Failure Recovery in Redundant Disk Arrays. Journal of Parallel and Distributed Databases 2, 1994
10 Srinivas N, Deb K. Multiobjective function optimization using nondominated sorting genetic algorithms. Evol. Comput., 1995, 2(3): 221～248