基于数字签名的DNA篡改提示方法研究*)

李 光 王亚东 苏小红 李 霞
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 为有效发现DNA数据是否被恶意篡改,保证研究者所使用的DNA数据是真实可靠的,即未经过篡改的,本文提出一种基于数字签名的DNA数据篡改提示方法,用向原有信息中加入数字签名的方法来解决该问题。设计了一整套加入、检测以及滤除签名的方法。该方法将数字签名嵌入DNA序列,在使用前通过校验数字签名来发现DNA是否篡改,以便给出篡改提示。实验表明,该方法可以有效发现篡改者对序列的恶意篡改,而且经添加签名后的序列与原始序列有一定的相似性,不易引起篡改者的注意。

关键词 DNA序列,篡改提示,数字签名

Research on Method of Prompting the Juggling of DNA Sequences Dased on Digital Signature

LI Guang WANG Ya-dong SU Xiao-hong LI Xia
(Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China)

Abstract It is an important question that how to find the juggling to the DNA data and how to ensure the data is right. This paper brings forward a method to deal with this question based on digital signature, and designs a full set of the means to add, examine and filter the digital signature. This method adds digital signature to the DNA sequences, and finds the juggling by checking it before using the data. The experiments show that this means can find the juggling effectively and the sequence adding the digital signature is same with the primal sequence, which can let the attacker pay less attention to them.

Keywords DNA sequences, Prompting the juggling, Digital signature

1 引言

随着生物学的发展,DNA测序技术取得了长足的进步。测序过程所需的时间变短,成本变低。大量的DNA序列已经被测定。这些已经测出的DNA序列都是宝贵的资源,应当与世界上其它的研究者共享。目前的做法是将它们放在因特网上,建立对世界各地的研究者开放的DNA数据库。这时我们如何保证所获得的数据是安全可靠的和是没有经过篡改的?这个问题已经成为一个非常重要的问题。一方面,研究者如果利用被篡改过的数据进行研究,那么将会得出错误的结论;另一方面,医生如果利用被篡改过的数据进行诊病,那么将会使患者的生命和健康受损。

目前,最常采用的保证数据安全可靠的方法就是进行访存控制,只允许经过特许的用户(一般是数据库管理员)对数据进行增、删、改操作,其它用户都只能访问和查找数据。这种方法是否可以奏效取决于是否正确地选择了特殊用户。显然,这种方法是有风险的。而且,即使一切正常,它也只能保证数据从数据库取出时是真实可靠的,对从数据库中取出到真正被用户使用的过程中是无法提供保护的。如果我们能对数据是否被人篡改进行判断,以发现数据是否正确,那么用户只需在使用数据前对数据进行检验,即可保证使用的数据是正确的了。

发现数据错误的基本方法之一是向受保护的数据中加入一些附加的数据,通过事先已知的附加数据与受保护数据之间的关系来检测数据是否被篡改,例如添加校验码以及加入数字水印等。目前,人们已设计出多种校验码,某些种类适合于检测随机错误,某些种类适合于检测连续错误,该方法被广泛地应用于存储器和通信领域。然而,由于校验码方法能够检测的错误位数是受到限制的,而对于人为篡改的情形,错误的出现并不服从某一确定的分布,很难预计篡改造成的错误位数,甚至攻击者可以针对某种校验码自身的弱点进行篡改,因此该方法对于人为篡改的适用性并不强。数字水印技术和信息隐藏技术紧密相连,它通过将特制的信息秘密载入原始作品(如数字图像)中来达到保护作品知识版权或证明产品的真实可靠性等目的。与加密术相比,数字水印技术处理后的作品与原作品没有明显的区别,不会引起人们去怀疑作品中可能添加了重要的隐含信息[1]。目前,数字水印技术主要应用于数字图像[2,3]、多媒体[4-6]、文本[7]以及地理信息系统[8,9]等领域。

本文使用添加数字签名的方法来解决DNA序列的篡改提示问题。数字签名是密码学研究的成果之一。在证明作品真实可靠性时,它可以作为一种嵌入信息。它是将作品进行单向哈希函数处理,然后再进行加密的结果。单向哈希函数是指那些正向计算比较简单,但逆运算是十分复杂的哈希函数,即给定函数的输入,很容易计算出输出,但给定一个想要得到的输出,基本上不可能找到相应的输入。签名是必须与查证的作品一起传输的元数据。因此,如果将数字签名和原数据分开存储和传输,就会存在丢失元数据的风险[10]。如果把签名嵌入到作品中,就可以防止这一问题的发生。嵌入签名以进行作品查证时,查证主要是通过检验签名和作品是否

*)Supported by the 863 High Technology Foundation of China under Grant No. 2003AA231010,生物信息数据共享体系平台建设。李 光 博士研究生,主要研究领域为生物信息学等;王亚东 教授,主要研究领域为知识工程、专家系统、生物信息技术等;苏小红 教授,主要研究领域为色彩匹配技术、计算机图形学、人工神经网络、数字图像处理等;李 霞 教授,主要研究领域为生物信息学等。

一致来进行的，所以对嵌入的方法没有特殊的要求，可以是鲁棒的，也可以是易碎的。

本文提出的方法，其基本思想就是把原序列的数字签名嵌入到DNA序列中来，形成新的序列，存储和传输新的序列。在用户使用前，进行数字签名的检测和滤除，提取出数字签名并进行校验，以判断DNA序列是否被人篡改。实验表明，该方法能够正确地检测出序列是否经过篡改，而且经添加签名后的序列与原序列具有一定的相似性，这就满足了不可见性，使这种方法处理后的序列不易引起攻击者的注意。

2 DNA序列的数字签名嵌入方法

添加数字签名的原始序列为DNA序列，由DNA单字符表示法的兼并性可知它是一个由15种字符组成的字符串。DNA单字符表示法的兼并性由表1来进行说明。

表1 DNA单字符表示法的兼并性

a	腺嘌呤	g	鸟嘌呤	t	胸腺嘧啶
c	胞嘧啶	m	a或c	r	a或g
w	a或t	s	c或g	y	c或t
k	g或t	v	a或c或g	h	a或c或t
d	a或g或t	b	c或g或t	n	a或c或g或t

设原始DNA序列为A，添加签名的方法如下。

Step1 计算A的数字签名s1，s1是一串二进制码，长度由选择的签名算法以及密钥来决定。

Step2 将s1转化成一串四进制码，并用“a”表示0，“g”表示1，“t”表示2，“c”表示3。经过这样的转化，得到的字符串记为B。

Step3 将B的长度转化成15进制，并按照表2的对应关系表示，得到字符串C，如果C的长度为奇数，在最左边填补一位0(用“a”来表示)。然后，将C从中间划分成等长的两个子串，记左侧的一个子串为C1，右侧的一个为C2。

表2 15进制表示法

a	0	g	1	t	2
c	3	m	4	r	5
w	6	s	7	y	8
k	9	v	10	h	11
d	12	b	13	n	14

Step4 参照图1的方法进行嵌入。拼接各字符串得到新的序列S，用S来代替A。其中K是一位标志位，其意义见表3。

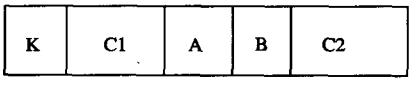

图1 对签名的嵌入方法的说明

表3 对图1中标志位K的说明

K的取值	表示的意义
a	没有添加签名
g	C1与C2的长度相等，为1
t	C1与C2的长度相等，为2
c	C1与C2的长度相等，为3

例如：选用现代智人的线粒体DNA中的一段(GenBank编号为AF392063，GenBank的网址为：http://www.ncbi.nlm.nih.gov/Web/Genbank)作为原始DNA序列，原序列为

“attctaatttaaactattctctgttctttcatggggaagcagatttgggtaccacccaagtattgactcacccatcaacaaccgctatgtatttcgtacattactgccagccaccatgatattgtacggtaccataaatacttgaccacctgtagtacataaaaacccaatccacatcaaaaccccctccccatgcttacaagcaagtacagcaatcacccctcagctatcacacatcaactgcaactccaaagccacccctcacccactaggataccaacaaacctacccaccctcaacagtacatagtacataaagccatttaccgtacatagcacattacagtcaaatcccttctcgcccccatggatgacccccctcagataggggtcccttgaccaccatcctccgtgaaatcaatatcccgcacaagagtgctactctcctcgctccgggcccataacacttgggggtagctaaagtgaactgtatcc”

计算出的数字签名B为

“ccgggtggcatcactgccttcagcactgacggaagcaccgacgatgtatgcaccttatagttgactccccattcctctcagccagcaggacctagccccaaatccctaaggaatggcatgattactatagagtctccccgtgcacaaatcagaagctgaaataaatacatactactactatttgcgagccaccgtgtgcacttatattatgccaggggagtcagcagctagtttctccgacgttgtactataatactatcttgtaaatgggtggatcctctctagaatgagggtttggcgtgggggcatcaggaaggaccttgattgcctaggggttcatactatagccataagcttggccgttaaatctattaagcaagccgcaaccgtgaggttctgattcctcagagatcgttactctcatgattgattgttcggtactcgtcatcattttagtctatatttcgtccttcgctaaatcgtcatagtgagacgtaactgacgttatgg”

B的长度为512，所以C=“atmt”，C1=“at”，C2=“mt”。因为C1，C2的长度相等，为2，所以标志位K为“t”。因此，最终的结果S为

“tatattctaatttaaactattctctgttctttcatggggaagcagatttgggtaccacccaagtattgactcacccatcaacaaccgctatgtatttcgtacattactgccagccaccatgaatattgtacggtaccataaatacttgaccacctgtagtacataaaaacccaatccacatcaaaaccccctccccatgcttacaagcaagtacagcaatcacccctcagctatcacacatcaactgcaactccaaagccacccctcacccactaggataccaacaaacctacccaccctcaacagtacatagtacataaagccatttaccgtacatagcacattacagtcaaatcccttctcgcccccatggatgacccccctcagataggggtcccttgaccaccatcctccgtgaaatcaatatcccgcacaagagtgctactctcctcgctccgggcccataacacttgggggtagctaaagtgaactgtatccccgggtggcatcactgccttcagcactgacggaagcaccgacgatgtatgcaccttatagttgactccccattcctctcagccagcaggacctagccccaaatccctaaggaatggcatgattactatagagtctccccgtgcacaaatcagaagctgaaataaatacatactactactatttgcgagccaccgtgtgcacttatattatgccaggggagtcagcagctagtttctcccgacgttgtactataatactatcttgtaaatgggtggatcctctctagaatgagggttggcgtgggggcatcaggaaggaccttgattgcctaggggttcatactatagccataagcttggccgttaaatctattaagcaagccgcaaccgtgaggttctgattcctcagagatcgttactctcatgattgattgttcggtactcgtcatcattttagtctatatttcgtccttcgctaaatcgtcatagtgagacgtaactgacgttatggmt”。

3 签名检测及滤除方法

由签名的嵌入方法容易得到签名的检测及滤除方法。

检测及滤除签名时，输入的原始信息为添加了签名的DNA序列，即S，检测方法如下。

Step1 读入S的第一位K，由表3的对应关系确定C1和C2的长度，设其长度为p1。将K从S中去掉，得到新的S。

Step2 读入S最左侧长为p1的子串C1以及最右侧长为p1的子串C2，拼合C1与C2，得到C，将C看成是一个15进制的数，由表2的对应关系计算出签名序列B的长度，设其为p2。去掉C1和C2，得到新的S。

Step3 S中最右侧长为p2的子串就是签名序列B，得到它并从S中去掉它，剩下的序列为原始序列A。

Step4 还原B为二进制，得到A的签名，用它检验A是否被改动。

以上一节例子中计算出的最终序列为例。

校验前的序列为

“tatattctaatttaaactattctctgttctttcatggggaagcagatttgggtacca

cccaagtattgactcacccatcaacaaccgctatgtatttcgtacattactgccagccaccat
gaatattgtacggtaccataaatacttgaccacctgtagtacataaaaacccaatccacatc
aaaaccccctccccatgcttacaagcaagtacagcaatcacccctcagctatcacacatcaa
ctgcaactccaaagccacccctcacccactaggataccaacaaacctacccacccctcaaca
gtacatagtacataaagccatttaccgtacatagcacattacagtcaaatcccttctcgcccc
catggatgacccccctcagatagggtcccttgaccaccatcctccgtgaaatcaatatcc
cgcacaagagtgctactctcctcgctccgggcccataacacttgggggtagctaaagtga
actgtatccccgggtggcatcactgccttcagcactgacggaagcaccgacgatgtatgc
accttatagttgactccccattcctctcagccagcaggacctagccccaaatccctaaggaa
tggcatgattactatagagtctccccgtgcacaaatcagaagctgaaataaatacatacta
ctactatttgcgagccaccgtgtgcacttatattatgccaggggagtcagcagctagtttc
tcccgacgttgtactataatactatcttgtaaatgggtggatcctctctagaatgagggttt
ggcgtgggggcatcaggaaggaccttgattgcctaggggttcatactatagccataag
cttggccgttaaatctattaagcaagccgcaaccgtgaggttctgattcctcagagatcgt
tactctcatgattgattgttcggtactcgtcatcattttagtctatatttcgtccttcgctaaa
tcgtcatagtgagacgtaactgacgttatggmt”

首位为“t”，C1 和 C2 的长度均为 2，C1 =“at”，C2 =“mt”，所以 C=“atmt”，签名长度为 512，得到签名序列 B 为

“ccgggtggcatcactgccttcagcactgacggaagcaccgacgatgtatgcacc
ttatagttgactccccattcctctcagccagcaggacctagccccaaatccctaaggaatgg
catgattactatagagtctccccgtgcacaaatcagaagctgaaataaatacatactactac
tatttgcgagccaccgtgtgcacttatattatgccaggggagtcagcagctagtttctccc
gacgttgtactataatactatcttgtaaatgggtggatcctctctagaatgagggtttggcg
tgggggcatcaggaaggaccttgattgcctaggggttcatactatagccataagcttgg
ccgttaaatctattaagcaagccgcaaccgtgaggttctgattcctcagagatcgttactct
catgattgattgttcggtactcgtcatcattttagtctatatttcgtccttcgctaaatcgtca
tagtgagacgtaactgacgttatgg”

同时得到原始序列 A 为

“attctaatttaaactattctctgttctttcatggggaagcagatttgggtaccaccc
aagtattgactcacccatcaacaaccgctatgtatttcgtacattactgccagccaccatga
atattgtacggtaccataaatacttgaccacctgtagtacataaaaacccaatccacatcaa
aaccccctccccatgcttacaagcaagtacagcaatcacccctcagctatcacacatcaact
gcaactccaaagccacccctcacccactaggataccaacaaacctacccacccctcaacagt
acatagtacataaagccatttaccgtacatagcacattacagtcaaatcccttctcgccccc
atggatgacccccctcagatagggtcccttgaccaccatcctccgtgaaatcaatatccc
gcacaagagtgctactctcctcgctccgggcccataacacttgggggtagctaaagtgaa
ctgtatcc”

在这个例子中，经计算，B 确实是 A 的签名序列，所以 A 没有被篡改，可以使用。

如果计算出 B 不是 A 的签名序列，那么 A 不能被使用，应该抛弃。

4 实验结果分析

采用三个数据集进行实验：第一个数据集包含的序列 GenBank 编号为 AF387914-AF387969[11]，本文中该数据集编号为 1；第二个数据集包含的序列 GenBank 编号为 AF125053-AF125089[12]，本文中该数据集编号为 2；第三个数据集包含的序列 GenBank 编号为 AF392063-AF392434[13]，本文中该数据集编号为 3。实验时，数字签名算法选择 MD5 加 RSA。

表 4 是分别向三个数据集中的序列加入签名，不经篡改就直接进行检验的结果（其中正确率是指算法得到正确检验结果的比率）由判断为未被篡改的例子个数除以例子总数来计算。表 5 是分别向三个数据集中的序列加入签名，然后对每个序列随机选择 2000 种篡改方案，对经每种篡改方案篡改后的序列进行检验的结果（其中篡改发现率是发现篡改的比率）由判断为被篡改的例子个数除以例子总数来得到。实验结果表明在三个数据集上，本文方法都可以对序列是否被篡改做出正确的判断。

表 4 向不同数据集中的序列加入签名并检验的结果

数据集编号	例子数	正确检验的例子数	正确率
1	56	56	100%
2	37	37	100%
3	372	372	100%

表 5 向不同数据集中的序列加入签名并随机篡改后检验的结果

数据集编号	例子数	发现篡改的例子数	篡改发现率
1	112,000	112,000	100%
2	74,000	74,000	100%
3	744,000	744,000	100%

表 6 是用朴素贝叶斯分类器对经使用本文方法添加签名后的序列进行分类的结果。贝叶斯分类器由原始序列进行训练，该分类器用于判定一个新序列在字母排列上更接近于哪一个数据集中的序列，如果某个添加签名后的序列被分到它的原始序列所在的那一类中，则认为对该序列的分类正确。分类结果受密钥选择的影响，表中数据是 1000 次随机选择密钥后的平均结果。实验结果证明，经本文方法处理后的序列与原始序列具有较好的相似性。

表 6 用朴素贝叶斯分类器对添加签名的序列进行分类的结果

数据集编号	例子个数	正确分类的例子个数	分类正确率
1	56	56	100.00%
2	37	37	100.00%
3	372	365	98.12%

表 7 是用随机生成的例子测试本文方法正确检验未经篡改的序列以及发现篡改的能力，其中，正例添加了签名后直接进行校验，反例添加了签名后经过篡改再进行校验，正、反例的原始序列以及加密时使用的密钥都是随机生成的。检验错误率是指算法错误的判断序列是否被篡改的例子在总例子中所占的比例。在正例的情形下，由被判断为被篡改的例子数除以总例子数得到；在反例的情形下，由被判断为未经篡改的例子数除以总例子数计算。实验结果证明，本文方法可以正确地检验出未经篡改的序列，并可以有效地发现篡改。

表 7 用随机例子进行测试的结果

例子类别	例子个数	错误处理的例子个数	检验错误率
正例	12,500,000	0	0.00000%
反例	9,800,000	2	0.00002%

结束语 本文通过将数字签名技术应用于 DNA 序列篡改提示领域，设计了一整套加入，检测以及滤除签名的方法。具体来讲，是将原始序列的数字签名进行编码，嵌入到 DNA 序列中来。实验证明，该方法能有效地发现对于 DNA 序列的篡改，经添加签名后的序列与原始序列具有一定的相似性。

目前，该方法只是将序列的数字签名嵌入到序列中来，只能检测序列是否被篡改，不能进行错误定位，也不能进行恢复。进一步的研究可以考虑如何实现错误定位以及自我恢复功能。

参 考 文 献

[1] 孙圣和，陆哲明. 数字水印处理技术. 电子学报，2000，28(8)：1-3

[2] Guo Huiping, Georganas N D. A Novel Approach to Digital Image Watermarking Based on a Generalized Secret Sharing Scheme. Multimedia Systems, 2003, 9(3): 249-260
[3] Guo Huiping, Georganas N D. Jointly Verifying Ownershi Pof an Image Using Digital Watermarking. Multimedia Tools and Applications, 2005, 27(3): 323-349
[4] 刘连山，李人厚，高琦. 视频数字水印技术综述. 计算机辅助设计与图形学学报，2005，17(3)：379-386
[5] 向辉. 图形数据数字水印技术. 系统仿真学报，2002，14(12)：1649-1651
[6] 蒋天发，周熠，何秉姣，等. 基于感知数字水印对音频信息稳健性影响的研究. 武汉大学学报(工学版)，2004，37(6)：93-96
[7] Brassil J T, Low S H, Maxemchuk N F. Copyright Protection for the Electronic Distribution of Text Documents// Proceedings of the IEEE. 1999, 87: 1181-1196
[8] 李媛媛，许录平. 用于矢量地图版权保护的数字水印. 西安电子科技大学学报(自然科学版)，2004，31(5)：719-723
[9] 王勋，林海，鲍虎军. 一种鲁棒的矢量地图数字水印算法. 计算机辅助设计与图形学学报，2004，16(10)：1377-1381
[10] Cox I J, Miller M L, Bloom J A. 数字水印. 王颖，黄志蓓，译. 北京：电子工业出版社，2003
[11] Makova K D, Ramsay M, Jenkins T, et al. Human DNA sequence variation in a 6. 6-kb region containing the melanocortin 1 receptor promoter, Genetics, 2001, 158: 1253-1268
[12] Harris E E, Hey J X. Chromosome evidence for ancient human histories, PNAS, USA, 2000, 96: 3320-3324
[13] Yao Y G, Nie L, Harpending H, et al. Genetic relationshi Pof Chinese ethnic populations revealed by mtDNA sequence diversity. Am J Phys Anthropol, 2002, 118: 63-76

(上接第 198 页)

心函数，用以建立特定应用领域的模拟器。这些函数包括对处理机、存储等资源的描述，以及对数据传输、计算和依赖关系等任务的描述，同时提供了诸如调度、顶测、时间和模拟的功能函数。通过随机生成的主机处理能力、存储资源和通信延迟可以描述实际的异构式分布式环境[4]；测试在一台酷睿双核 T2300(1. 66GHz)、内存 512M 的主机上完成，每个测试结果均取 3 次模拟的平均值。

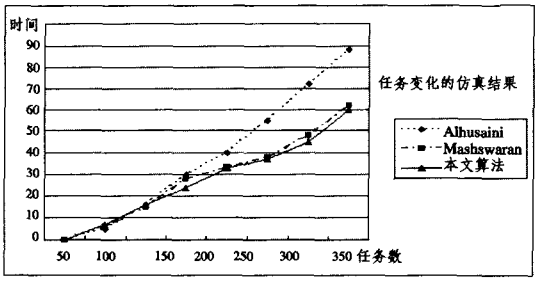

图 3　任务数变化时的仿真结果

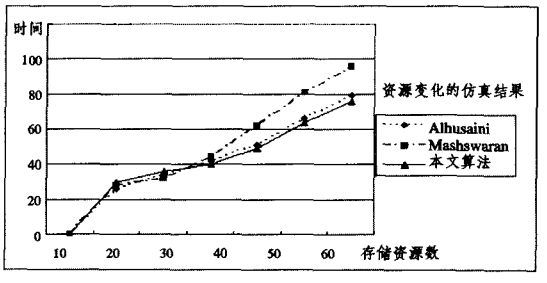

图 4　存储资源数变化时的仿真结果

图 3 显示了将变化的任务数映射到 50 个计算资源和 25 个存储资源的模拟结果。图 4 是固定任务中的任务数量为 200、计算资源为 50 个时变化存储资源数量的模拟结果，其中时间单位是以 Simgrid Toolkit 中的虚拟时间单位计量。

从仿真结果中可以看出所提出的动态资源映射算法比 Alhusaini 算法和 Maheswaran 算法总体性能更为稳定。原因是该算法在任务执行阶段完成的资源映射，更好地解决了 Alhusaini 算法在任务运行之前就进行资源映射，无法更为准确地确定资源在运行阶段是否可用，而导致的任务不能按照调度时间正常完成；另外，虽然 Maheswaran 是动态调度算法，在任务数变化的情况下可以达到较好的资源映射效果，但是该算法仅仅考虑了计算资源而没有考虑非计算资源，因此与所提出的动态资源映射算法比较，在存储资源数变化的仿真实验中明显要差。

结束语　Alhusaini 算法采用基于任务相容加权图进行资源映射。然而该算法在构建最大独立集合时并没有区分关键任务，而且所有的映射操作都是在任务执行之前完成，因此达不到实际的最佳效果。而动态资源映射算法采用最大权值的关键任务优先，而其它任务随机选取的原则相结合的方法，一方面尽可能达到资源映射的要求，另一方面也大大降低了时间复杂度和减少资源映射所需的时间。

动态资源映射算法在任务执行阶段利用加权函数进行任务的资源映射。通过仿真实验发现，与其它资源映射算法相比，它可以更有效地减少调度时间；从效率来说，算法时间复杂度是 $O(N^2R)$，相对于其它资源映射算法来说具有较低的时间复杂度。

参 考 文 献

[1] Alhusaini A H, Prasanna V K, Raghavendra C S. A framework for mapping with resource co-allocation in heterogeneous computing system[C]// 9th Heterogeneous Computing Workshop. 2000: 273-286
[2] Alhusaini A H, Prasanna V K, Raghavendra C S. Run-Time Adaptation for grid environments[C]// Proceedings 15th International Parallel and Distributed Processing Symposium. 2001: 864-874
[3] Christofides N. Graph Theory: an Algor ithm ic Approach [M]. London: Academic press, 1975
[4] Maheswaran M, Siegel H J. A dynamic matching and scheduling algorithm for heterogeneous computing systems [A]// 7th IEEE Symposium on Heterogeneous Computing Workshop(HCW'98) [C]. Orlando: IEEE Computer Society, 1998: 57-69
[5] Casanova H. Simgrid——A toolkit for the simulation of application scheduling[C]// Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid (CCGrid'01). 2001
[6] Azzedin F, Maheswaran M, Arnason N. A synchronous co-allocation mechanism for grid computing system s. Cluster Computing [J], 2004, 7 (1): 39-49
[7] Yang Juan, Bai Yun, Qiu Yuhui. A decentralized resource allocation policy in minigrid[J]. Future Generation Computer Systems, 2006
[8] Topcuoglu H, Hariri S, Wu M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans. on Parallel and Distributed System, 2002, 13: 260-274
[9] Rehn C. Dynamic mapping of cooperating tasks to nodes in a distributed system [J]. Future Generation Computer Systems, 2006: 35-45
[10] Wang L Z, Cai W T , Lee B S, et al. Resource co-allocation for parallel task s in computational grids [A]// Proceedings of the International Workshop on Challenges of Large Application s in Distributed Environments [C]. Seattle: IEEE Computer Society, 2003: 88-95
[11] 丁箐，陈国良，顾钧. 计算网格环境下的一个统一的资源映射策略[J]. 软件学报，2002，13 (7)：1303-1308
[12] 刘丽，杨扬，田志民. 网格计算环境下资源联合分配的映射策略与机制[J]. 计算机工程，2005，16(31)：130-133