代　　号	10701	学　　号	0911120728
分 类 号	TP391.41	密　　级	公开

题 (中、英 文) 目 云计算平台的任务资源调度调度的设计与实现

The Desgined and Implentments of Scheduler In Cloud Computing Plantform

作 者 姓 名	方祯	**指导教师姓名、职务**	马建峰 教授
学 科 门 类	工学	**学科、专业**	计算机系统结构

提交论文日期 二〇一四年十二月

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学风与优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别中以标和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：_____　　日期_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。

(保密论文在解密后遵守此规定)

本学位论文属于保密，在____ 年解密后适用本授权书。

本人签名：_____　　导师签名：_____

日　　期 _____　　日　　期 _____

摘 要

图像是多媒体信息时代的主要数字信息资源。如何从海量的图像数据中迅速而准确地搜寻到我们所需的信息成为研究热点。...。本文的主要研究内容包括以下几部分：

(1) 针对人类视觉系统更加关注视觉信息丰富的图像区域，提出基于视觉信息量的图像显著性检测算法。首先根据图像像素间的相关性，度量图像内容的视觉冗余程度；接着，根据像素的分布特性，计算图像内容的信息熵；然后，从信息熵中去除图像的视觉冗余，获得图像内容的视觉信息量；最后，采用视觉信息量来度量图像显著性，从而建立显著性检测模型。实验结果表明提出的基于视觉信息熵的图像显著性检测算法能准确检测出任何潜在图像特征下的显著性内容，且所得显著图与主观视觉关注区域高度吻合。

(2) 针对人类视觉系统对具有规则结构的图像区域高度敏感，提出基于结构自相似性的恰可识别失真阈值估计算法。视觉系统非常善于提取图像的结构信息，并通过结构比对及模式匹配来理解图像内容，因此视觉系统对具有自相似结构区域分辨能力强。根据相邻像素间的相似性，首先度量图像内容的结构自相似程度；然后，根据结构自相似性提出新的空域掩膜方程；最后，结合现有的亮度敏感度方程和所提空域掩膜方程，建立恰可识别失真阈值估计模型。实验结果表明，所提算法能准确估计纹理区域的恰可识别失真阈值，而现有其它算法无法准确估计该区域。

（3）...

（4）...

（5）...

上述研究成果从主观视觉感知的角度对图像处理进行分析与研究，具有一定的前瞻性和挑战性。本文在理论分析上取得一些突破，在技术实现上具有一些创新，为基于主观视觉感知的客观图像处理开辟了新的思路，具有重要的理论意义及实用价值。

关键词： **人类视觉系统 视觉关注 恰克识别失真阈值 结构自相似 视觉敏感度 不确定信息 图像质量评价**

ABSTRACT

In the multimedia era, image information plays a very important role in our daily life...The main contributions of this thesis can be summarized as follows:

(1) According to that the HVS pays more attention to these regions with abundant visual information, we introduce a visual information based saliency detection model. Firstly, the visual redundancy of an image is measured based on the correlations among pixels. And according to the distribution of pixels, the entropy of the image is computed. Then, by removing the visual redundancy from the entropy, the quantity of visual information of the image is acquired. Finally, the visual information is used to estimate the saliency of the image. Experimental results show that the proposed saliency detection model can accurately estimate the salient object under any potential image feature, and the saliency map from the proposed model is highly coincided with the subjective visual attention.

(2) According to that the HVS is highly sensitive to regular regions, we introduce a structural self-similarity based JND threshold estimation model. The HVS is highly adapted to extract structural information for image perception and understanding, and the HVS is highly sensitive to these regions with self-similar structures. According to the similarities among nearby pixels, we firstly measure the structural self-similarity of an image. And then, a novel spatial masking function is introduced based on structural self-similarity. Finally, combining the existing luminance adaptation function and the proposed spatial masking function, a new JND threshold estimation model is built. Experimental results demonstrate that the proposed JND model can accurately estimate the JND threshold of the textural region, where the existing JND models always failed.

(3) ...

(4) ...

(5) ...

The results above are image processing researches from the perspective of subjective visual perception, which are forward looking and full of challenges. This thesis has some breakthrough in theory and some innovation in technology. This work opens up a new way for visual perception based image processing, which has extremely important theoretical significance and application value.

Keywords: **The Human Visual System, Visual Attention, Just Noticeable Distortion, Structural Self-Similarity, Visual Sensitivity, Uncertainty, Image Quality Assessment**

目 录

第一章 引言

1.1 研究的背景及意义

1.2 国内外研究现状

1.3 本论文的研究内容

1.4 本论文的组织结构

第二章 云计算资源管理与调度相关技术

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量的硬件资源, 而硬件的资源的代价往往比较昂贵，所以如何充分挖掘硬件资源潜力从而增加硬件的利用率 shi yi

2.1　OpenStack 云计算平台

OpenStack 是一个美国国家航空航天局和 Rackspace 合作研发的，以 Apache 许可证授权，并且是一个自由软件和开放源代码项目。

OpenStack 是一个云平台管理的项目，它不是一个软件。这个项目由几个主要的组件组合起来完成一些具体的工作。

OpenStack 是一个旨在为公共及私有云的建设与管理提供软件的开源项目。它的社区拥有超过 130 家企业及 1350 位开发者，这些机构与个人都将 OpenStack 作为基础设施即服务（简称 IaaS）资源的通用前端。OpenStack 项目的首要任务是简化云的部署过程并为其带来良好的可扩展性。OpenStack 的核心组件有以下 9 个:

- 计算（Compute）: Nova，Nova 是 OpenStack 云中的计算组织控制器。支持 OpenStack 云中实例（instances）生命周期的所有活动都由 Nova 处理。其中的调度器 nova-schedule 作为一个守护进程运行，通过恰当的调度算法从可用资源池获得一个计算服务。nova-scheduler 会根据诸如负载、内存、可用域的物理距离、CPU 构架等作出调度决定。
- 对象存储（Object）: Swift，其最初是由 Rackspace 公司开发的高可用分布式对象存储服务，并于 2010 年贡献给 OpenStack 开源社区作为其最初的核心子项目之一，为其 Nova 子项目提供虚机镜像存储服务。Swift 支持多租户模式、容器和对象读写操作，适合解决互联网的应用场景下非结构化数据存储问题。
- 镜像（Image）: Glance, 用来管理在 OpenStack 集群中的镜像，但不负责实际的存储。它为从简单文件系统到对象存储系统（如 OpenStack Swift 项目）的多种存储技术提供了一个抽象。除了实际的磁盘镜像之外，它还保存描述镜像的元数据和状态信息。
- 身份（Identity）: Keystone（OpenStack Identity Service）是 OpenStack 框架中，负责身份验证、服务规则和服务令牌的功能，它实现了 OpenStack 的 Identity

API。

- 网络地址管理（Network）：Neutron 是 OpenStack 核心项目之一，提供云计算环境下的虚拟网络功能.
- 块存储 (Block)：Cinder–提供块存储（Block Storage），类似于 Amazon 的 EBS 块存储服务，OpenStack 中的实例是不能持久化的，需要挂载 volume，在 volume 中实现持久化。Cinder 就是提供对 volume 实际需要的存储块单元的实现管理功能。
- UI 界面 (Dashboard)：Horizon,Horizon 套件提供 IT 人员一个图形化的网页接口，让 IT 人员可以综观云端服务目前的规模与状态，并且，能够统一存取、部署与管理所有云端服务所使用到的资源。
- 测量 (Metering)：Ceilometer, 主要负责监控数据的采集，采集的项目包括虚拟机的性能数据，n eutron-l3-router 使用的网络带宽，glance，cinder，swift 等租户使用信息，甚至是通过 snmp 采集物理机的信息，以及采集支持 opendaylight 的网络设备信息。
- 编配 (Orchestration)：Heat 类似于 AWS 的 CloudFormation, heat 实现了一种自动化的通过简单定义和配置就能实现的云部署方式。可以在 heat 模板中定义连串相关任务，然后交由 heat，由 heat 按照一定的顺序执行 heat 模板中定义的一连串任务。利用 heat 还可以连接到 neutron 来帮助编排负载均衡和其他网络功能。

2.2 资源管理抽象模型

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.3 资源管理与调度系统范型

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.3.1 集中式调度器

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.3.2 两级调度器

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.3.3 状态共享调度器

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.4　资源调度策略

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.4.1 FIFO 调度策略

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.4.2 公平调度策略

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.4.3 能力调度器

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.4.4 延迟调度策略

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

2.5　本章小结

对于企业和公司，为了完成各种对外的服务以及公司内部业务逻辑，需要大量

第三章 CBDRF 算法的相关技术

3.1　图的联通性算法

3.1.1 强联通分量

在有向图 G 中，如果两个顶点可以相互通达，则称两个顶点强连通 (strongly connected)。如果有向图 G 的每两个顶点都强连通，称 G 是一个强连通图。非强连通图有向图的极大强连通子图，称为强连通分量 (strongly connected components)。

下图中，子图 1，2，3 为一个强连通分量，因为顶点 1，2，3 两两可达。4，5，6 也分别是两个强连通分量。

直接根据定义，用双向遍历取交集的方法求强连通分量，时间复杂度为 O(N*N+M)。更好的方法有算法有，Tarjan 算法，Kosaraju 算法。其中 Tarjan 算法和 Kosaraju 算法均是是基于对图深度优先搜索的算法, 与 Kosaraju 算法不同，Tarjan 算法只进行一遍深度优先搜索索，较 Kosaraju 算法进行两遍深度优先搜索有 30% 的性能提升。

Tarjan 算法以一个有向图 G 作为输入，每个强连通分量为搜索树中的一棵子树首先定义结点 u 的深度优先搜索标号 DFN(u)，表示节点 u 是被访问的次序编号。此外，每个结点 u 还有一个值 Low(u)，表示从 u 出发经有向边可到达的所有结点中最小的次序号。显然 Low(u) 总是不大于 DFN(u)，且当从 v 出发经有向边不能到达其他结点时，这两个值相等，此时，以 u 为根的搜索子树上的所有节点属于同一个强连通分量。其中 Low(u) 在深度优先搜索的过程中求得，通过上述可以发现以 u 为根的搜索子树上的所有节点属于同一个强连通分量当且仅当 DFN(u)=Low(u) 时。而 Low(u) 的计算由以下公式给出:

$$Low(u) = \min \begin{cases} DFN(u) & \\ Low(v) & \text{(u,v) 为树枝边，u 为 v 的父节点} \\ DFN(v) & \text{(u,v) 为指向栈中节点的后向边 (非横叉边)} \end{cases}$$

由此可得，Tarjan 算法的基本流程如下:

1. 任选图 G 中的未被访问的结点 u 开始进行深度优先搜索遍历，如果深度优先搜索结束后仍有未访问的结点，则再从中任选一点再次进行。
2. 搜索过程中标记已访问的节点，如果是已访问的结点不再访问。

3. 搜索时，把当前搜索树中未处理的节点加入一个堆栈，回溯时可以判断栈顶到栈中的节点是否为一个强连通分量。

由此可以得到 Tarjan 算法在搜索时的主要的伪代码

Algorithm 1 Tarjan Algorithm

tarjan(u)

```
1  Index ← time + 1
2  DFN[u] ← time
3  Low[u] ← time
4  Stack.push(u)
5  for each(u, v) in E[u]
6      if visted(v)
7          tarjan(v)
8          Low[u] ← min(Low[u], Low[v])
9      elseif inStack(v)
10         Low[u] ← min(Low[u], DFN[v])
11 if DFN[u] == Low[u]
12     while u! = v
13         v ← Stack.pop()
14         print(v)
```

3.1.2 拓扑排序

在图论中，如果一个有向图无法从某个顶点出发经过若干条边回到该点，则这个图是一个有向无环图（DAG 图）。因为有向图中一个点经过两种路线到达另一个点未必形成环，因此有向无环图未必能转化成树，但任何有向树均为有向无环图。如右图，不为有向树，但为有向无环图。

对一个有向无环图 (Directed Acyclic Graph 简称 DAG)G 进行拓扑排序，是将 G 中所有顶点排成一个线性序列，使得图中任意一对顶点 u 和 v，若边 (u,v)□E(G)，则 u 在线性序列中出现在 v 之前。通常，这样的线性序列称为满足拓扑次序 (Topological Order) 的序列，简称拓扑序列。简单的说，由某个集合上的一个偏序得到该集合上的一个全序，这个操作称之为拓扑排序。

一个较大的工程往往被划分成许多子工程，我们把这些子工程称作活动

(activity)。在整个工程中，有些子工程 (活动) 必须在其它有关子工程完成之后才能开始，也就是说，一个子工程的开始是以它的所有前序子工程的结束为先决条件的，但有些子工程没有先决条件，可以安排在任何时间开始。为了形象地反映出整个工程中各个子工程 (活动) 之间的先后关系，可用一个有向图来表示，图中的顶点代表活动 (子工程)，图中的有向边代表活动的先后关系，即有向边的起点的活动是终点活动的前序活动，只有当起点活动完成之后，其终点活动才能进行。通常，我们把这种顶点表示活动、边表示活动间先后关系的有向图称做顶点活动网 (Activity On Vertex network)，简称 AOV 网。

在 AOV 网中，若不存在回路，则所有活动可排列成一个线性序列，使得每个活动的所有前驱活动都排在该活动的前面，我们把此序列叫做拓扑序列 (Topological order)，由 AOV 网构造拓扑序列的过程叫做拓扑排序 (Topological sort)。AOV 网的拓扑序列不是唯一的，满足上述定义的任一线性序列都称作它的拓扑序列。

由 AOV 网构造拓扑序列的拓扑排序算法主要是循环执行以下两步，直到不存在入度为 0 的顶点为止。

Algorithm 2 Topological Sort

TopologicalSort(G)

```
1  Queue.clear()
2  for each vertex u in G
3      if u.indeg == 0
4          Q.push(u)
5          isVisited[u] ← True
6  while Queue.size() > 0
7      u = Queue.front()
8      Queue.pop()
9      for each (u, v) in Edge[v]
10         v.indeg ← v.indeg − 1
11         if v.indeg ==0 and isVisited[v] == False
12             Q.push(v)
13             isVisited[v] ← True
```

1. 选择一个入度为 0 的顶点并输出之；

2. 从网中删除此顶点及所有出边。
3. 循环结束后，若输出的顶点数小于网中的顶点数，则输出“有回路”信息，否则输出的顶点序列就是一种拓扑序列。

3.1.3 并查集

在计算机科学中，并查集是一种树型的数据结构，其保持着用于处理一些不相交集合（Disjoint Sets）的合并及查询问题。有一个联合 -查找算法（union-find algorithm）定义了两个操作用于此数据结构：

- Find：确定元素属于哪一个子集。它可以被用来确定两个元素是否属于同一子集。
- Union：将两个子集合并成同一个集合。

因为它支持这两种操作，一个不相交集也常被称为联合 -查找数据结构（union-find data structure）或合并 -查找集合（merge-find set）。其他的重要方法，MakeSet，用于建立单元素集合。有了这些方法，许多经典的划分问题可以被解决。

为了更加精确的定义这些方法，需要定义如何表示集合。一种常用的策略是为每个集合选定一个固定的元素，称为代表，以表示整个集合。接着。Find(x) 返回 x 所属集合的代表，而 Union 使用两个集合的代表作为参数。

并查集实现方式有多种，包括并查集链表和并查集森林，其中并查集森林是并查集的高效实现。并查集森林是一种将每一个集合以树表示的数据结构，其中每一个节点保存着到它的父节点的引用。这个数据结构最早由 Bernard A. Galler 和 Michael J. Fischer 于 1964 年提出，但是经过了数年才完成了精确的分析。

在并查集森林中，每个集合的代表即是集合的根节点。“查找”根据其父节点的引用向根行进直到到底树根。“联合”将两棵树合并到一起，这通过将一棵树的根连接到另一棵树的根。实现这样操作的一种方法是：

这是并查集森林的最基础的表示方法，这个方法不会比链表法好，这是因为创建的树可能会严重不平衡；然而，可以用两种办法优化。

第一种优化方式是按秩合并，即总是将更小的树连接至更大的树上。因为影响运行时间的是树的深度，更小的树添加到更深的树的根上将不会增加秩除非它们的秩相同。在这个算法中，术语“秩”替代了“深度”，因为同时应用了路径压缩时（见下文）秩将不会与高度相同。单元素的树的秩定义为 0，当两棵秩同为 r 的树联合时，它们的秩 r+1。只使用这个方法将使最坏的运行时间提高至每个 MakeSet、Union 或 Find 操作 $O(\log n)$。采用按秩合并的 Union 伪代码如下所示：

Algorithm 3 Union-Set Union

Union(u, v)

```
1  rootu = Find(u)
2  rootv = Find(v)
3  if rootu ! = rootv
4       if u.rank < v.rank
5            u.parent ← v.parent
6            v.rank ← v.rank + u.rank
7       else
8            v.parent ← u.parent
9            u.rank ← v.rank + u.rank
10 return u.parent
```

另外一种优化的方式是采用路径压缩，这是一种在执行“查找”时扁平化树结构的方法。关键在于在路径上的每个节点都可以直接连接到根上；他们都有同样的表示方法。为了达到这样的效果，Find 递归地经过树，改变每一个节点的引用到根节点。得到的树将更加扁平，为以后直接或者间接引用节点的操作加速。采用路径压缩的 Find 伪代码如下所示：

Algorithm 4 Union-Set Find

Find(u)

```
1  if u.parent ! = root
2       u.parent ← Find(u.parent)
3  return u.parent
```

这两种技术可以互补，可以应用到另一个上，每个操作的平均时间仅为 O($\alpha(n)$)，$\alpha(n)$ 是 n = f(x) = A(x,x) 的反函数，并且 A 是急速增加的阿克曼函数。因为 α(n) 是其的反函数，α(n) 对于可观的巨大 n 还是小于 5。因此，平均运行时间是一个极小的常数。

3.2 公平性的度量

3.2.1 max-min 公平

3.2.2 简氏公平

3.3 本章小结

第四章 CBDRF 调度系统的设计与实现

4.1 CBDRF 调度系统的提出

4.1.1 通信任务的关联性

4.1.2 系统的负载均衡

4.1.3 DRF 分配策略

4.2 CBDRF 调度系统的设计

4.2.1 任务调度解析

4.2.2 调度迭代控制

4.2.3 调度迭代评估

4.3 CBDRF 调度的实现

4.3.1 CBDRF 调度的流程

4.3.2 关联任务的解析

4.3.3 基于 DRF 算法的分配

4.3.4 关联任务的合并

第五章 实验与结果分析

5.1　实验环境

5.2　实验结果和分析

第六章 总结与展望

插图索引

表格索引

致 谢

在此论文完成之际，回首过往五年硕博连读学习时光，有欣慰也有艰辛，有成功也有失败，太多的感激、太多的谢意想要表达。在此，谨向所有在我生活中留下色彩的人表示由衷的感谢！

首先，我要衷心感谢我的指导老师...

...

感谢齐飞副教授提供 xdthesis 模版，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

...

攻读博士学位期间的研究成果

期刊论文

1. **Jinjian Wu**, Fei Qi, and Guangming Shi. “Self-Similarity Based Structural Regularity for Just Noticeable Difference Estimation,” Journal of Visual Communication and Image Representation, 23(6): 845-852, August, 2012.(SCI)
2. ...

会议论文

1. **Jinjian Wu**, Fei Qi, and Guangming Shi. “Image Quality Assessment Based on Improved Structural Similarity,” PCM 2012, Singapore.(EI)
2. ...

已授权专利

- 齐飞，**吴金建**，石光明，刘焱。基于人类视觉系统的图像感兴趣区域自动提取方法。授权专利号：ZL 2009 1 0022191.0
- ...

参加研究的科研项目

- **国家 863 计划项目**, 无线传感器网络协同获取融合与表达, 2008.1.-2009.12. (2007AA01Z307);
- ...