基于高校服务器信息的网络空间资源分类

潘星宇1) 喻明理1) 姜伟峰1)

1) 清华大学计算机科学与技术系，北京，100084

摘 要 利用nmap爬取高校全部服务器所得信息，对以服务器和端口为代表的网络空间资源进行了多种分类尝试。以清华大学、北京大学的20余万条信息为例，设计有四种不同的分类方法，并通过前端展示和更新每日的服务器状态及其分类。

关键词 网络空间，资源分类，服务器，高校，nmap，清华大学，北京大学

The classification of cyberspace resources based on

the servers’ information of colleges

Pan Xingyu1) Yu mingli1) Jiang Weifeng1)

1)Department of Computer Science and Technology, Tsinghua University, Beijing, 100084

**Abstract** Based on the information from the servers of colleges via nmap, we have used different ways to classify the cyberspace resources, especially servers and their ports. Using above 200 thousand pieces of information from Tsinghua University and Peking University, we have designed four different classification methods and shown the servers’ status which is updated every day on our website.

**Key words** cyberspace, source classification, server, college, nmap, Tsinghua University, Peking University

# **1** 引言

随着网络的进一步普及化的复杂化，网络空间中的资源呈现爆发性的增长。一般认为，网络空间资源是对于网络空间中存在的万物的总称，既包括网络空间中存在的路由器等硬件，也包括网站、ftp等服务器端的软件，内容庞杂。为了便于进一步统计和管理网络空间资源，对网络空间资源进行合适的分类变得极为重要。当前，学术界尚未能够对于网络空间资源的分类方法达成共识，也尚未有一种得到公认的分类标准，因而，这一方向存在较大的研究价值。

网络空间资源的分类需要遵循一定规律，即对于网络空间万物需兼具正交性、可用性、完备性等特点，能够将网络空间资源能够以有限种类别进行全覆盖式的分类。

考虑到网络空间数据量巨大，本次研究聚焦于高校的服务器信息，希望能够通过对高校的服务器信息提出一种或几种符合要求的、合理的分类方法，以能够一窥网络空间资源分类的全局。考虑到高校服务器信息的完备性和资源获取的可操作性，本次研究重点从清华大学、北京大学、中国人民大学、复旦大学、上海交通大学等国内知名的五所高校入手，通过获取上述高校的服务器信息，提出基于端口状态、基于网络资源位置、基于网络设备的端口空闲程度、基于网络设备的服务名称等四种分类方式。

而为了实现能够在前端呈现至少每日更新一次的结果，减轻服务器在更新信息方面的压力、保证服务器信息的时效性，本次研究重点为清华大学和北京大学两所高校，并以检索所得的两所高校的超过20万条服务器信息为样本进行分析。

# **2** 网络空间资源获取

## **2.1** 获取情况概述

本次研究采用nmap获取清华大学和北京大学的服务器的全部实时信息。Nmap作为一种高效的网络连接端扫描工具，能够充分扫描服务器开放的网络服务端，并确定运行在各自连接端的服务。由于Nmap提供了极为方便快捷的服务器信息获取方式，能够使本研究团队在有限的时间内获取足够多的可供分析的信息，因而在本次研究中关于网络空间资源获取方面所采用的主要方法即为nmap。

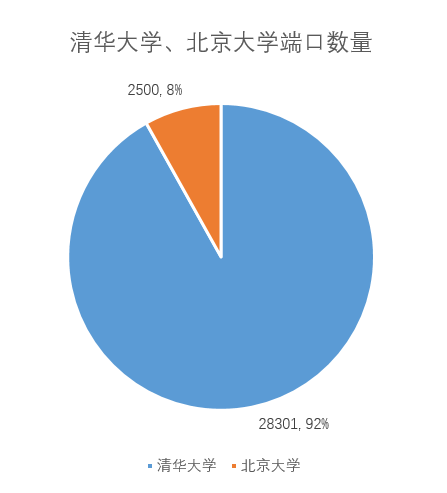
本次研究从清华大学、北京大学、中国人民大学、复旦大学、上海交通大学等五所高校获取其服务器信息。其中，以清华大学和北京大学两所高校共搜集的20余万条信息和检所获取的6万余条信息作为样本进行分析。其中两校在端口数量方面的情况如下图所示。

图1 清华大学、北京大学端口数量情况

为了提供更好的检索服务，本研究团队除了通过搭建前端展示两所高校检所得到的全部服务器信息外，还通过每日更新保持信息的时效性。考虑到更新数据的规模和速度，因而重点以清华大学、北京大学两所高校的服务器信息作为样本进行整理、展示和分析。

## **2.2** 网络资源的选择标准

本次研究所选择的网络资源主要是服务器信息，这一选择的主要原因是服务器信息在网络空间资源中的代表性，同时也是基于对现实因素的考虑。

为了保持网络资源的更新频率，因而对于网络资源的数据量和获取速度提出了很高的要求，而主机、路由器等其他资源信息的获取难度较大，一方面主要是爬取更详细的服务信息存在较大的不确定性，容易导致不同时期爬取所得的信息存在较大的差别，另一方面则是爬取主机、路由器等速度较慢，在本次研究中每次爬取需要耗时9s以上，甚至在100s内都无法返回结果，因而难以满足本次实验对于样本数量规模的要求。

即便如此，仅清华大学和北京大学的服务器信息获取仍然面临数量过大的问题。需要获取的IP地址区段为59.66.0-255.0-255和166.111.0-255.0-255，使用nmap的情况下，每个平均耗时仍然在5s左右，在串行情况下每次更新的总耗时超过110小时，信息量庞大、获取耗时大。

## **2.3** 未能找到的网络资源原因分析

即便基于现实因素和实际需要的考虑，我们选取相对更容易稳定获取的两所高校的服务器信息作为研究对象，但在搜集这部分网络资源的过程中，仍然存在部分资源无法找到的情况。主要原因包括以下三点：

① 没有主机。寻找到的资源可能是一个空IP，因而无法返回符合要求的服务器信息；

② 访问权限问题。有些服务器没有完全开放，因而存在对于访问权限的限制，这一点在高校中仍然是比较普遍的，因而访问权限问题同样部分阻碍了获取全部网络资源信息；

③ 防火墙问题。由于目前高校一般都拥有自身的校园网，很多服务器仅面向校园网开放，对于外网设置有防火墙，本研究团队难以获取这些服务器的网络资源。

仍然有其他如网络状况等原因，在此不再列举。尽管有部分网络资源由于种种原因未能获取，但仍然不影响本次研究的推进。

## **2.4** 实验问题

在本次研究过程中，也存在一些难以避免的问题，在研究结束时仍然没有获得理想的解决方案。这些问题的出现，一些是由于网络状况所致，一些则是由于服务器等各种原因所致。主要遇到的问题包括：

① nmap稳定性不足。在数据量较大的时候，nmap会存在跑网站不稳定的情况，为了提高数据搜集的效率，在研究所使用的爬取脚本中将时间上限设为120s。而在这种情况下，nmap在获取一些服务器资源的过程中，仍然存在超时的情况，且有些时候超时原因并不明确；

② 执行条件和服务器状态不稳定。由于执行条件的不稳定，导致返回结果存在很大的差距。如在实验中，同样的指令分别爬取IP和100个IP一起爬取所获得的结果经常完全不同，而即便是100个IP一起爬取，每次爬取所获得的结果有时也是不完全一样的、甚至存在比较大的差别。

这些实验问题的出现导致网络资源的获取并不像预想中顺利；且一些问题出现原因不明也影响了获取的服务器信息的不稳定。

# **3** 网络空间资源的分类

在获取网络空间资源的基础上，本研究团队进行了符合要求的分类。

分类标准需要符合正交性、可用性、完备性等特点，尤其是正交性特点，即所有分类需要涵盖全部信息；基于这一点，我们考虑的几种分类形式均为排他性或采用完全对立的指标。

## **3.1** 分类标准

根据检索得到的数据量，本研究团队采用了如下四种符合要求的分类标准：

① 基于端口状态。将获取的三万余端口的状态根据返回信息分为open、closed、filtered三类，覆盖全部端口的状态情况；

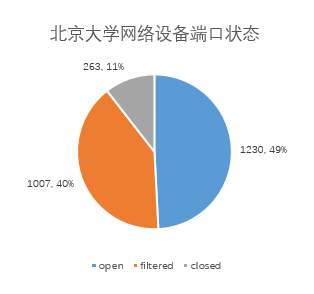
② 基于网络资源位置。我们根据服务器的位置，将网络资源的位置分为清华大学、北京大学、中国人民大学、复旦大学、上海交通大学等五类，并将展示的网络资源样本分为清华大学和北京大学两类。这种分类方式看似是预设性的，但是这种分类的原则却并不是局限的，在其他网络资源中基于位置进行分类同样是一种基本的分类标准；

③ 基于网络设备的端口空闲程度。为了评价网络设备的端口空闲程度，我们所衡量的指标为服务器所开放的端口数量，在对我们获取的资源进行分析之后，我们将边界值定为5个，即当开放的端口数大于或等于5个时，我们定义这个网络资源为忙碌资源，相应的，当开放的端口数小于5个时，我们定义这个网络资源为空闲资源；

④ 基于网络设备的服务名称。网络设备所提供的服务有各式各样的名称，仅我们搜集到的数据而言，网络设备的服务名称就达上百个。

## **3.2** 分类结果

3.2.1 基于端口状态

根据检所得到的资源信息来看，北京大学的端口开放数量不足总数量的一半，而清华大学的这一比例接近70%。

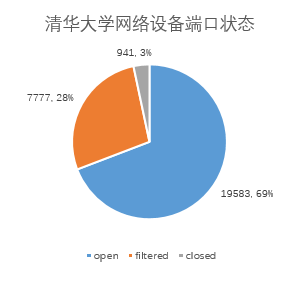
图2 北京大学端口状态情况

图3 清华大学端口状态情况

3.2.2 基于网络资源位置

如前章所展示的情况，清华大学服务器的网络端口数量超过28000个，而北京大学不足清华大学的十分之一，为2500个。

3.2.3 基于网络设备的端口空闲程度

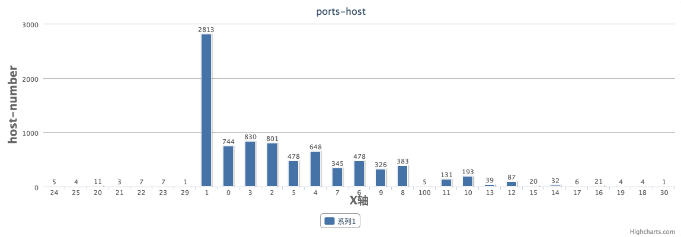
 根据我们的分类标准，空闲资源比例大于忙碌资源比例，按照服务器所开放的端口数量来看，开放1个端口的情况最为普遍，而开放端口数量最多的服务器所开放的端口数量达到100个。

图4 网络设备的端口空闲程度统计表

3.2.4 基于网络设备的服务名称

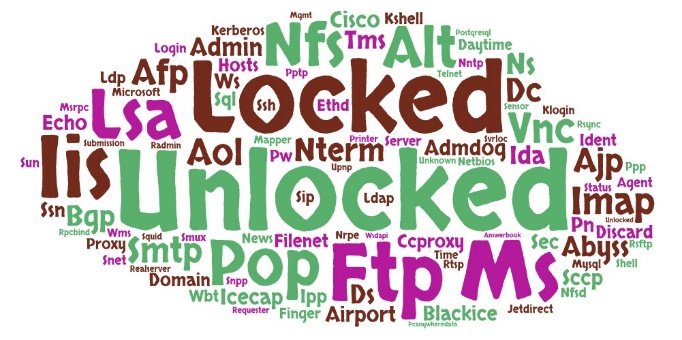
网络设备所提供的服务名称非常多样，在上百种被统计在内的服务名称中，使用最多的几种服务名称为unlocked, locked, Ftp, lis, Ms等。下图根据服务名称出现频次制成。

图5 网络设备的服务名称频次图

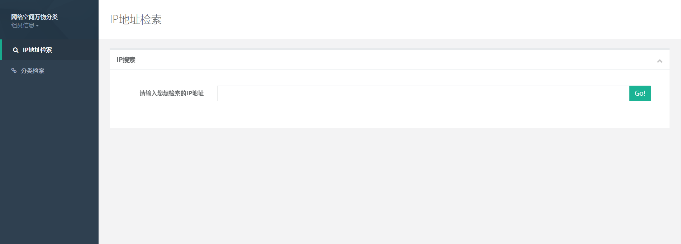
# **4** 前端展示及操作

本次研究搭建的展示前端的链接为：

[http://www.misscut.cn:8081](http://www.misscut.cn:8081/ip_search)。

## 4.1 IP地址检索

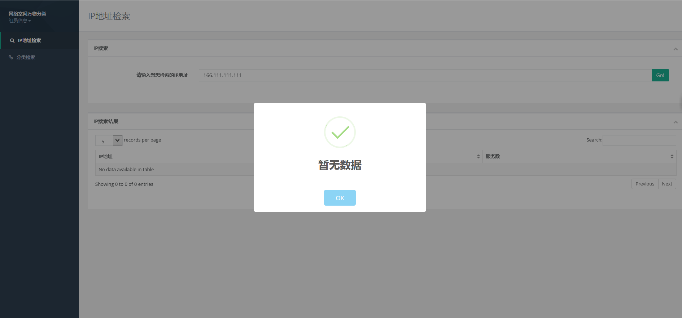
点击进入链接后，显示的为“IP地址检索”界面，即可在文本框中直接输入IP地址搜索器所在位置和拥有的端口数量。



检索成功的结果：



检索失败的结果：



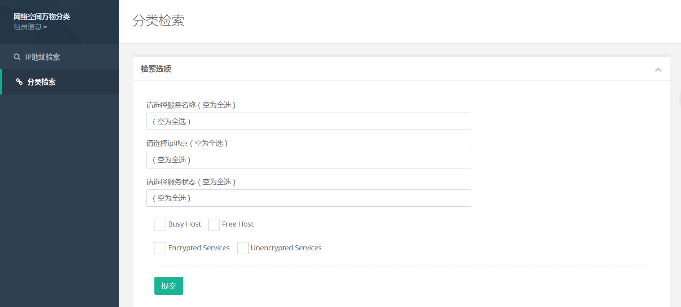
## 4.2 分类检索

分类检索界面可以针对服务器名称、IP地点、服务器状态等信息进行检索。

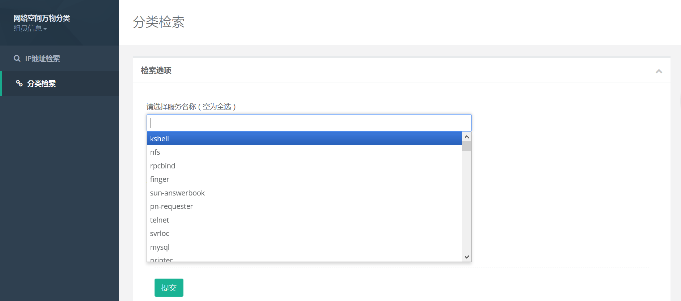
值得注意的是，在检索时，如果不加限制地检索清华大学的所有服务，会因为数据量较大、前端渲染表格时间长，因而展示效果较差。建议使用该分类检索的时候，在相对较小的范围内搜索。

除此以外，分类检索中服务器忙碌状态和是否加密两项需要至少选择一项方能完成检索。

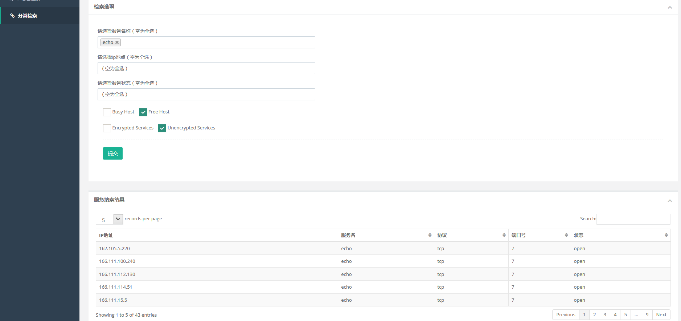
分类检索界面为：



按服务器名称选择检索：



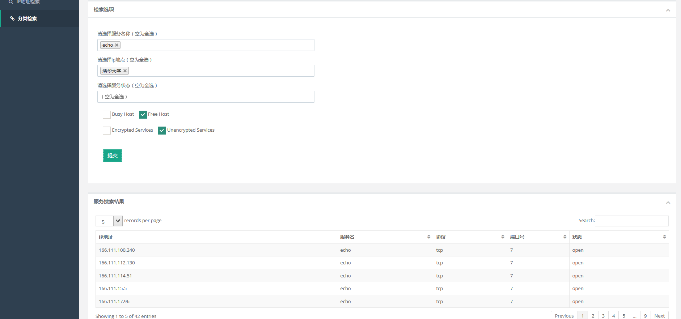
按服务器名称检索得到的结果：



按IP地点选择搜索：



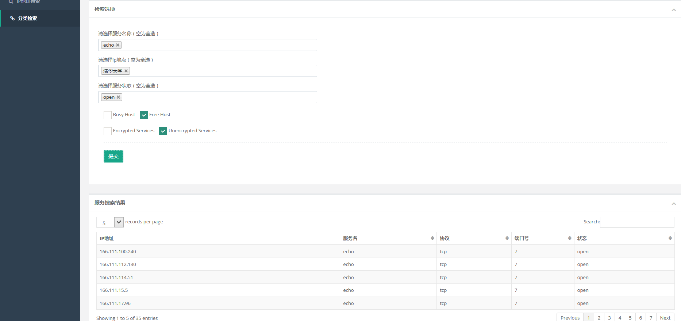
按IP地点检索得到的结果：



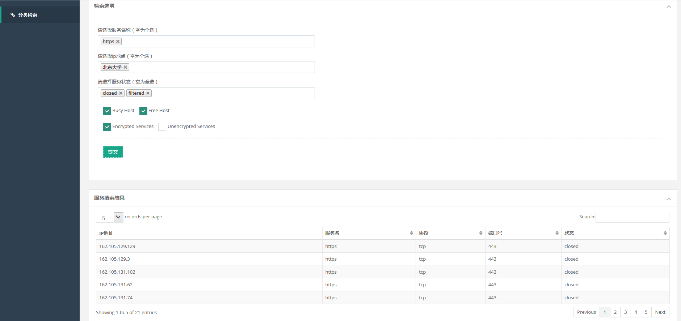
按服务器状态选择检索：



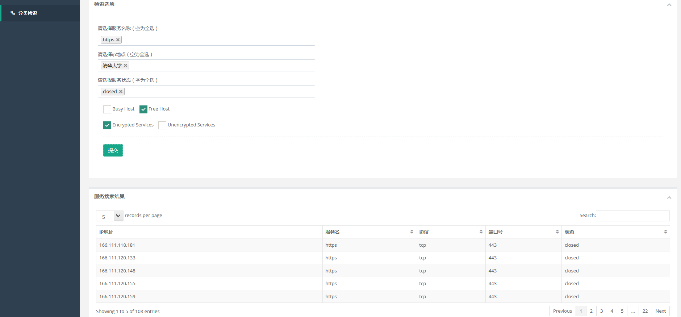
按服务器状态检索得到的结果：



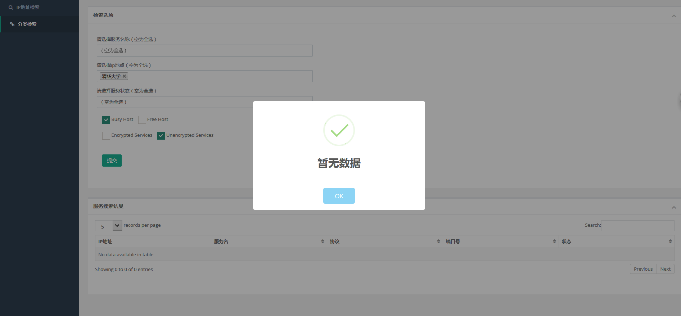
一种相对全面的检索示意：



另一种相对全面的检索示意：



一种暂无数据结果的检索示意：



# **5** 总结

在本次研究中，我们利用nmap手段，基于以高校服务器数据为代表的网络空间资源，提出了符合正交性等要求的四种网络空间资源分类方法，并将获取的服务器信息资源和分类情况的结果以每日更新的方式展现在前端。

参 考 文 献

[1] Gordon Fyodor Lyon. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, 2009.

[2] Alois Ferscha,  J Johnson. Performance prototyping of parallel applications in N-MAP. international conference on algorithms and architectures for parallel processing, 1996.

[3] Alois Ferscha, and [James Johnson](https://www.aminer.cn/profile/James%20Johnson/53f494bbdabfaeb4be77b501). N-MAP: an environment for the performance oriented development process of efficient distributed programs. Future Generation Comp. Syst., pp. 571-584, 2000.

[4] Robert M. Colomb. Information spaces: the architecture of cyberspace. Information. Spaces: the architecture of cyberspace, 2002.

[5] Jia Guo, and Ing-Ray Chen. A Classification of Trust Computation Models for Service-Oriented Internet of Things Systems. SCC New York, 2015.