In [ ]:

```
项目ID：28
项目名称：magnetW
小组成员：徐志臻 51215903003 王聪聪 51215903017
```

In [ ]:

```
# 项目的基本背景和发展历程介绍

## 技术类型

magnetW是一个磁力搜索聚合工具。 magnetW通过访问并解析各大磁力链接搜索网站的结果，实现磁力搜索结果

## 版本发布历史

GITHUB上有三个版本发布记录：

1. [3.0.0](https://github.com/xiandanin/magnetW/releases/tag/3.0.0)
    1. 发布软件
2. [3.1.0](https://github.com/xiandanin/magnetW/releases/tag/3.1.0)
    1. 更新了解析规则
    2. 增强了解析器
    3. 支持二级页面的源站
    4. 增加云解析功能
    5. 增加了一些菜单和设置
    6. 支持选择安装位置
3. [3.1.1](https://github.com/xiandanin/magnetW/releases/tag/3.1.1)
    1. 更新了一批规则
    2. 支持Socks5代理（#68 #79 ）
    3. 支持macOS仅关闭窗口不退出应用
    4. 支持自动分配和自定义端口
    5. 修复了解析器的一些问题

## 主要贡献者的构成（国家、区域和组织等）

1. [xiandanin](https://github.com/xiandanin)：项目的所有者，贡献了绝大多数代码
2. [JsonHive](https://github.com/xiandanin/magnetW/issues?q=is%3Apr+author%3AJsonHiv
3. [ggymm](https://github.com/xiandanin/magnetW/issues?q=is%3Apr+author%3Aggymm):提交
4.

## CI/CD 的使用

无
```

项目的历史轨迹分析

In [1]:

```
var openDigger = require('../src/open_digger');
```

In [229]:

```
function middle(args){
   args.sort() //排序
   if(args.length%2===0){
       return ((args[args.length/2]+args[args.length/2-1])/2);
   }else{
       return args[parseInt(args.length/2)];
  }
}

function average(nums) {
    return nums.reduce((a, b) => a + b) / nums.length;
}
```

In [88]:

```
query(
    `SELECT count(*),toMonth(created_at)
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'WatchEvent'
    GROUP BY toMonth(created_at)
    LIMIT ${limit}`
).then(res=>{console.log(res);})
```

```
[
  { 'count()': '89', 'toMonth(created_at)': 1 },
  { 'count()': '92', 'toMonth(created_at)': 2 },
  { 'count()': '108', 'toMonth(created_at)': 3 },
  { 'count()': '49', 'toMonth(created_at)': 4 }
]
```

In [151]:

```
ar startYear = 2018, endYear = 2022;
ar years = [];
ar all = [];
or (var y = startYear; y <= endYear; y++) years.push(y);
romise.all(years.map(year => {
   return openDigger.driver.clickhouse.query(`SELECT count(*) as star,toMonth(created
   FROM github_log.year${year}
   WHERE repo_name = 'xiandanin/magnetW' and type = 'WatchEvent'
   GROUP BY toMonth(created_at)
   LIMIT ${limit}`
.then(res=>{
       all.push.apply(all,res)
   });
))
```

Out[151]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [153]:

```
all.sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
```

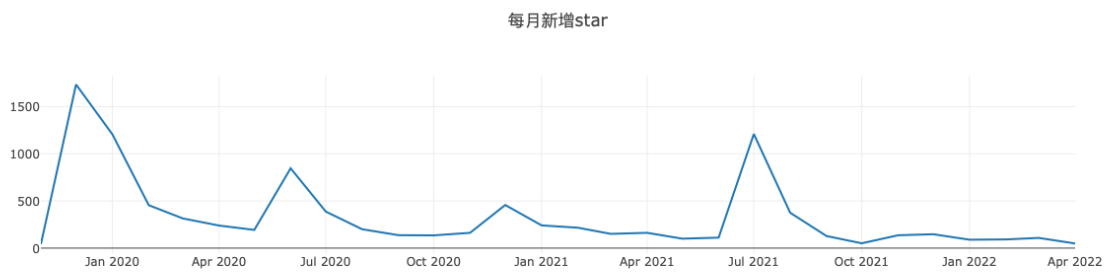
Out[153]:

```
[
  { star: '41', month: 11, year: 2019 },
  { star: '1735', month: 12, year: 2019 },
  { star: '1206', month: 1, year: 2020 },
  { star: '455', month: 2, year: 2020 },
  { star: '315', month: 3, year: 2020 },
  { star: '239', month: 4, year: 2020 },
  { star: '193', month: 5, year: 2020 },
  { star: '846', month: 6, year: 2020 },
  { star: '387', month: 7, year: 2020 },
  { star: '200', month: 8, year: 2020 },
  { star: '137', month: 9, year: 2020 },
  { star: '135', month: 10, year: 2020 },
  { star: '162', month: 11, year: 2020 },
  { star: '457', month: 12, year: 2020 },
  { star: '241', month: 1, year: 2021 },
  { star: '216', month: 2, year: 2021 },
  { star: '151', month: 3, year: 2021 },
  { star: '162', month: 4, year: 2021 },
  { star: '100', month: 5, year: 2021 },
  { star: '111', month: 6, year: 2021 },
  { star: '1210', month: 7, year: 2021 },
  { star: '375', month: 8, year: 2021 },
  { star: '128', month: 9, year: 2021 },
  { star: '51', month: 10, year: 2021 },
  { star: '136', month: 11, year: 2021 },
  { star: '147', month: 12, year: 2021 },
  { star: '89', month: 1, year: 2022 },
  { star: '92', month: 2, year: 2022 },
  { star: '108', month: 3, year: 2022 },
  { star: '49', month: 4, year: 2022 }
]
```

In [154]:

```
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.star),
        name: '新增star',mode: 'scatter'
    }], {title: `每月新增star`, xasix: { type: 'category' } })
```

Out[154]:



In [157]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(crea
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'ForkEvent'
    GROUP BY toMonth(created_at)
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

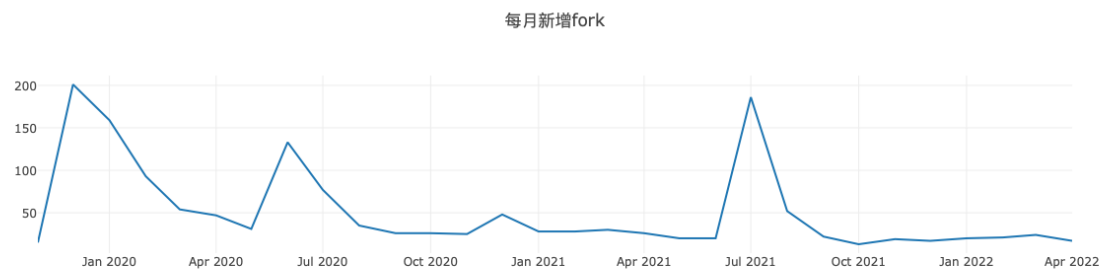
Out[157]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [159]:

```
all.sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),
        name: '新增fork',mode: 'scatter'
    }], {title: `每月新增fork`, xasix: { type: 'category' } })
```

Out[159]:



In [161]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(issu
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'IssuesEvent'
    GROUP BY toMonth(issue_created_at)
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

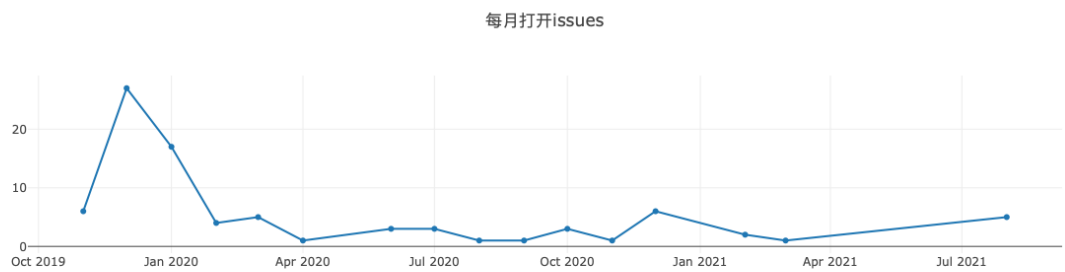
Out[161]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [163]:

```
all.sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),
        name: '打开issues',mode: 'scatter'
    }], {title: `每月打开issues`, xasix: { type: 'category' } })
```

Out[163]:



In [164]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(issu
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'IssuesEvent'
    GROUP BY toMonth(issue_closed_at)
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

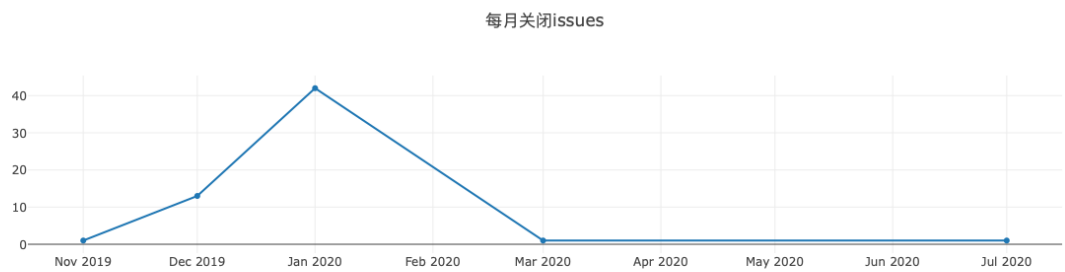
Out[164]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [174]:

```
all = all.filter(x=>x.year>2015).sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),
        name: '关闭issues',mode: 'scatter'
    }], {title: `每月关闭issues`, xasix: { type: 'category' } })
```

Out[174]:



In [179]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(crea
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'PullRequestEvent'
    GROUP BY toMonth(created_at)
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

Out[179]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [181]:

```
all = all.filter(x=>x.year>2015).sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),
        name: '打开pr',mode: 'scatter'
    }], {title: `每月打开PR`, xasix: { type: 'category' } })
```

Out[181]:

In [182]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(pull
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'PullRequestEvent'
    GROUP BY toMonth(pull_merged_at)
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

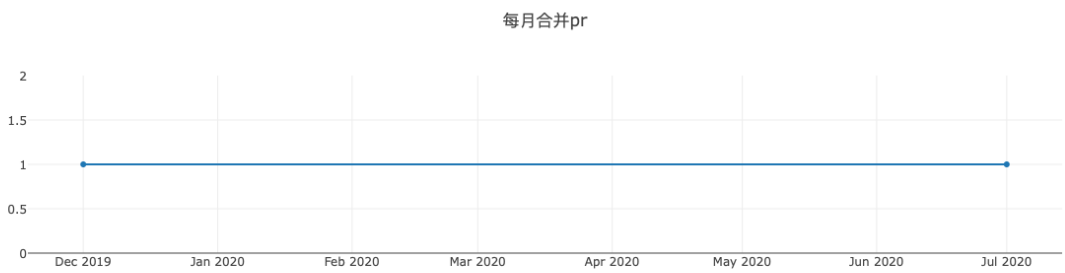
Out[182]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [184]:

```
all = all.filter(x=>x.year>2015).sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),
        name: '打开pr',mode: 'scatter'
    }], {title: `每月合并pr`, xasix: { type: 'category' } })
```

Out[184]:



In [185]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT count(*) as count,toMonth(crea
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW'
    GROUP BY toMonth(created_at),actor_id
    LIMIT ${limit}`
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

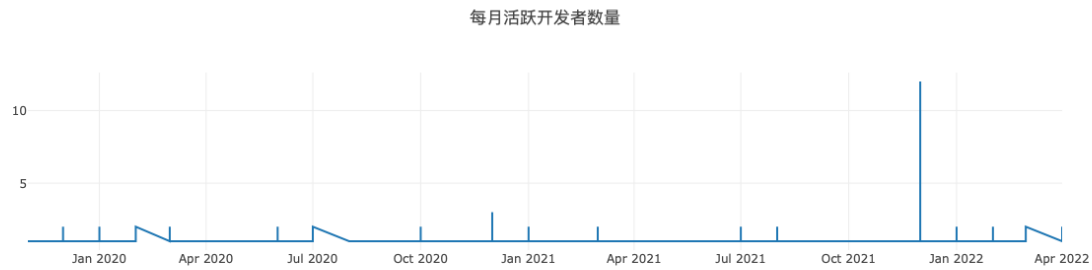
Out[185]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [187]:

```
all = all.filter(x=>x.year>2015).sort((a,b)=>{
    return 12*a.year+a.month-12*b.year-b.month
})
openDigger.render.plotly([{
        x: all.map(d => d.year.toString()+'-'+d.month.toString()),
        y: all.map(d => d.count),mode: 'scatter'
    }], {title: `每月活跃开发者数量`, xasix: { type: 'category' } })
```

Out[187]:



In [207]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT issue_closed_at,issue_created_
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'IssuesEvent' and toYear(issue_
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

Out[207]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [230]:

```
console.log(middle(all.map(x=>Math.floor((new Date(x.issue_closed_at)-new Date(x.iss
console.log(average(all.map(x=>(new Date(x.issue_closed_at)-new Date(x.issue_created
```

```
13
23.838158068783066
```

In [231]:

```
var startYear = 2018, endYear = 2022;
var years = [];
var all = [];
for (var y = startYear; y <= endYear; y++) years.push(y);
Promise.all(years.map(year => {
    return openDigger.driver.clickhouse.query(`SELECT pull_merged_at,created_at
    FROM github_log.year${year}
    WHERE repo_name = 'xiandanin/magnetW' and type = 'PullRequestEvent' and toYear(c
).then(res=>{
        all.push.apply(all,res)
    });
}))
```

Out[231]:

```
[ undefined, undefined, undefined, undefined, undefined ]
```

In [232]:

```
console.log(middle(all.map(x=>Math.floor((new Date(x.pull_merged_at)-new Date(x.crea
consol
e.log(average(all.map(x=>(new Date(x.pull_merged_at)-new Date(x.created_at))/(24*360
#pr创建时间在哪
```

```
-1
-0.00001157407407407 4073
```

In [6]:

```
[
  { type: 'WatchEvent', 'count()': '1776' },
  { type: 'CreateEvent', 'count()': '11' },
  { type: 'PushEvent', 'count()': '49' },
  { type: 'IssuesEvent', 'count()': '33' },
  { type: 'GollumEvent', 'count()': '59' },
  { type: 'PullRequestEvent', 'count()': '9' },
  { type: 'ReleaseEvent', 'count()': '3' },
  { type: 'ForkEvent', 'count()': '216' },
  { type: 'DeleteEvent', 'count()': '75' },
  { type: 'IssueCommentEvent', 'count()': '79' },
  { type: 'CommitCommentEvent', 'count()': '1' }
]
```

```
[
  { repo_name: 'YTVanced/VancedManager', cnt: '2884' },
  { repo_name: 'dnSpy/dnSpy', cnt: '1379' },
  { repo_name: 'netchx/netch', cnt: '1308' },
  { repo_name: 'vmware/clarity', cnt: '1098' },
  { repo_name: 'h2y/Shadowrocket-ADBlock-Rules', cnt: '995' },
  { repo_name: 'GitSquared/edex-ui', cnt: '990' },
  { repo_name: 'helm/charts', cnt: '796' },
  { repo_name: 'PowerShellMafia/PowerSploit', cnt: '746' },
  { repo_name: 'DeviaVir/zenbot', cnt: '645' },
  { repo_name: 'CoatiSoftware/Sourcetrail', cnt: '586' }
]
```

In [ ]:

```
# 可能归档的原因
1. 项目的README上有注明：本应用开源且免费，仅用于爬虫技术交流学习，搜索结果均来自源站，亦不承担任何
项目被人非法使用，出于法律风险考虑终止项目开发。
2.从统计数据上看，参与到该项目的活跃贡献者始终增长较少，且项目后期新增Stars数量也迅速变少，因此开发
持续关注的情况下终止开发。
# 项目归档后可能产生的影响
对开发者来说，终止项目避免了用户不当使用给其带来的法律风险。对用户来说则失去了磁力搜索聚合工具持续优
# 对开源项目如何可持续发展
需要有持续性的商业模式，支持个人开发者，直接从用户或粉丝获得收入，向受捐助者（开源贡献者）承诺少量经
有针对保护个人开发者的法律协议，帮助个人开发者解决相关法律风险问题。
```