中国移动
China Mobile

基于PostgresSQL及其插件进行大数据敏捷开发

——一种低成本的大数据集成方案

湖北移动网优中心
2017.10.15

目录

1、LTE覆盖分析工具研究背景

集中覆盖规划存在的困难：

评估算法单一，算法还有待完善，问题点**精度低**

评估模型考虑要素比较少，现场拟合度低，模型**系统性差**

数据量极大，归一化处理耗时过长，**费时耗力**

智能化

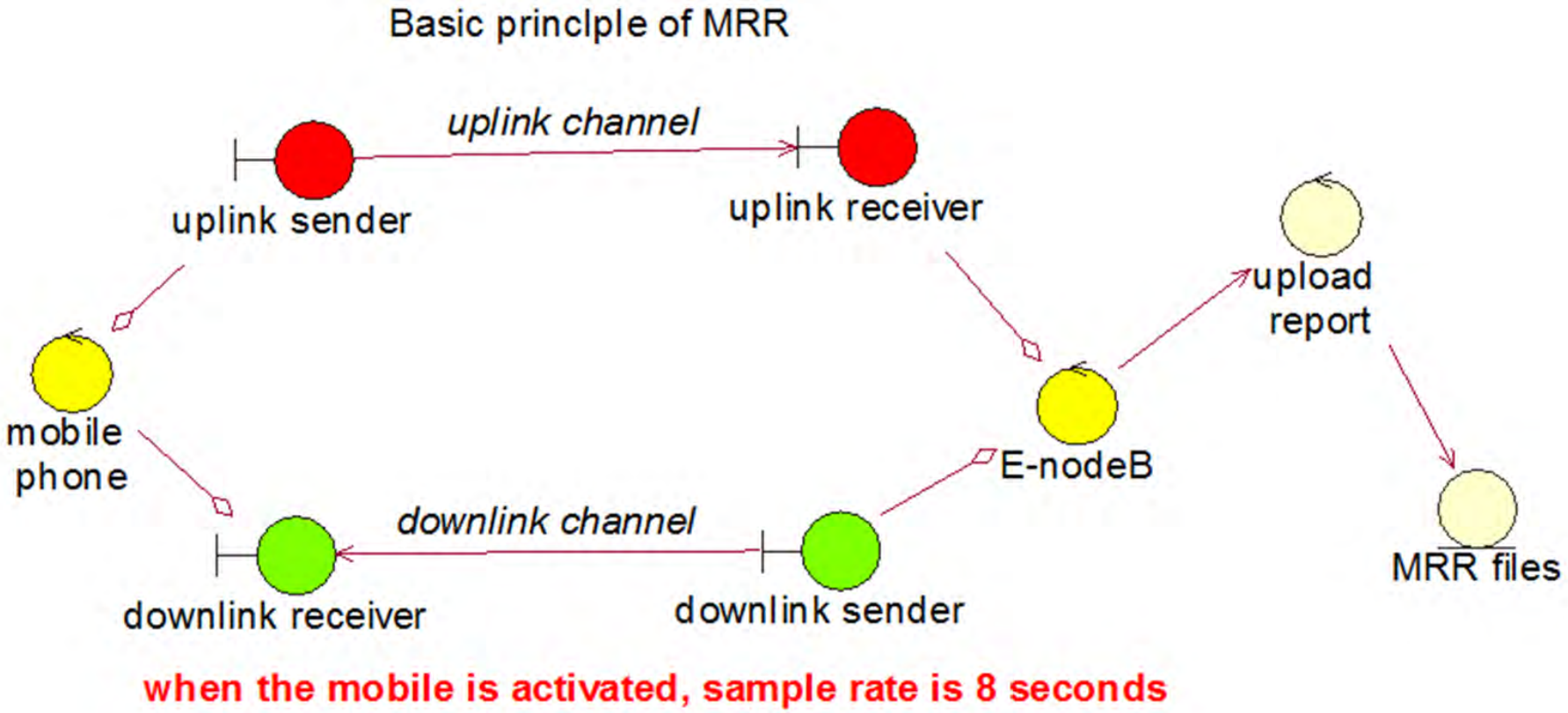
高效化

是LTE网络覆盖精准规划的重要演进方向。

为解决上述困难，湖北公司考虑建设**基于MRO大数据无线网覆盖评估工具**，实现覆盖规划**智能化、系统和高效化，提供快速、高效、精准的覆盖规划方案**。

2、什么是MRO大数据

MRO是一种手机测量原始数据。



当移动手机用户处于业务连接状态时，他将周期性的测量服务小区，和邻小区的信号强度和服务质量，精确评估出全网的覆盖质量。 以一个2000W的省级网络为例，1天的原始数据为**2T**， 记录数为**200亿**。

3、网优MRO大数据应用的难点

中国移动 China Mobile

旧有MRO方法进行覆盖评估面临以下问题：

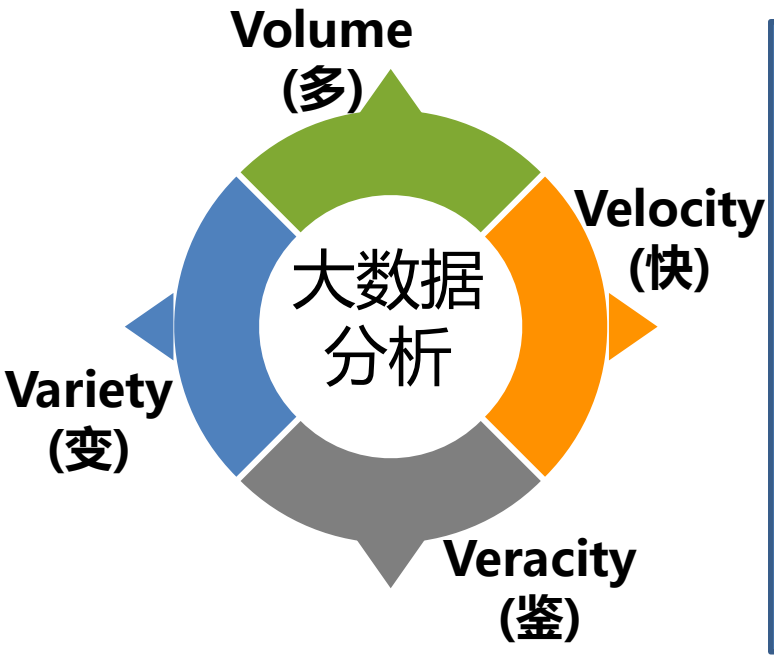
1、投入产出比低。以湖北为例，要完成约50000数量站点的MRO文件分析，需投入12台32核服务器资源，另外由于分析方法的限制，相关服务产出较少（2016的MRO相关应用的仅4个）

2、处理时延大。传统MRO数据处理框架处理时延大，3天MR原始数据，需7天才能解析完毕。

3、开发灵活性差。以往的MRO相关应用开发周期较长，不够灵活，平均约需3-6个月。

4、大数据应用的特点及MRO应用的难点

MRO分析是典型的大数据分析场景，需针对大数据的“4V”特点进行针对性解决。



1、Volume(多)：以湖北为例，4G站点数约5万，常态化开启后每日数据量为1.2T。

2、Velocity(快)：数据输入达100G/小时，数据输出达300G/小时。高速数据的处理需重点解决。

3、Variety(变)：数据来源及格式统一。无需考虑。

4、Veracity(鉴) ：MRO数据自OMC直采，可确保数据真实。无需考虑。

受限于MRO数据数据处理效率(Velocity)，基本上无法进行实时解析及应用，限制了MRO应用的场景。

6、大数据开发工作的困难与对策

中国移动 China Mobile

大数据敏捷研发体系

问题	对策	
数据挖掘效率低	MPP数据库	**•筛选适合的MPP数据库** 利用postgres-sql的稳定性和扩展性，选择GP数据库。
功能扩展复杂度大	微服务	**•基于事件驱动的微服务框架** 将一个系统功能解耦为多个微服务，采用事件驱动的机制进行服务粘合。
运维升级费时费力	私有云	**•基于弹性容器的私有云平台** 基于弹性容器搭建一个私有云平台，屏蔽底层的硬件细节，实现云化运维。

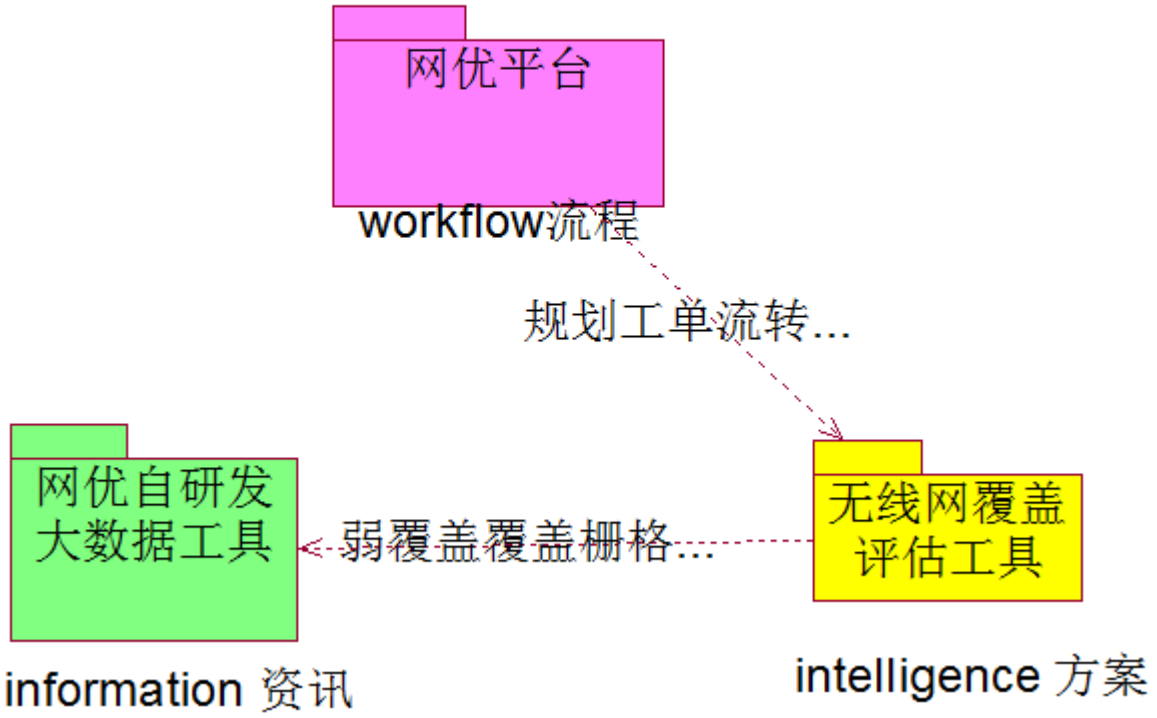
自主研发体系基于团队共创，通过人才激活培养，构建模块化自研发流程，利用低门槛开发工具等方法打造了自研发团队。

目录

1、基于MRR的覆盖分析工具

2017年3月，利用4周时间，开发基于MRR大数据的LTE覆盖分析工具。

基于MRR大数据的无线网覆盖分析工具



提供3个接口：

1）弱覆盖栅格评估

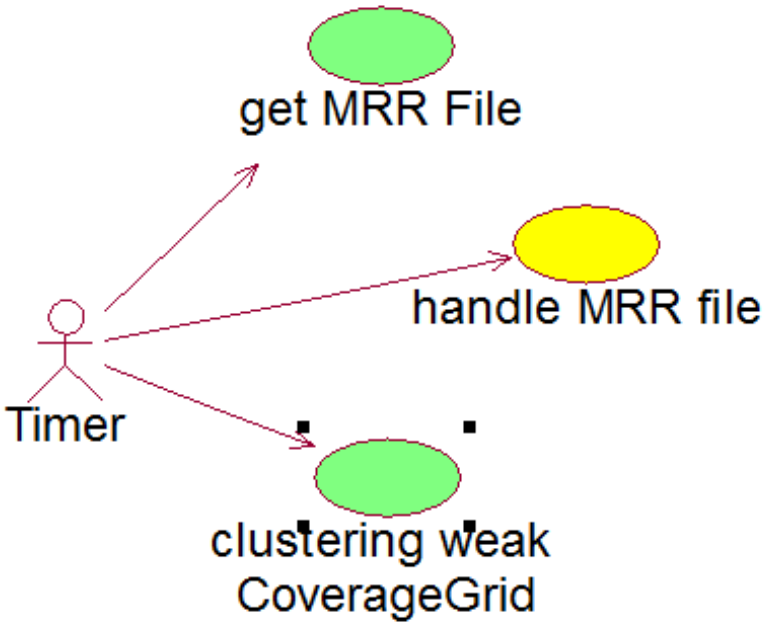
2）新增站点评估

3）环比覆盖情况评估

2、项目需求模型

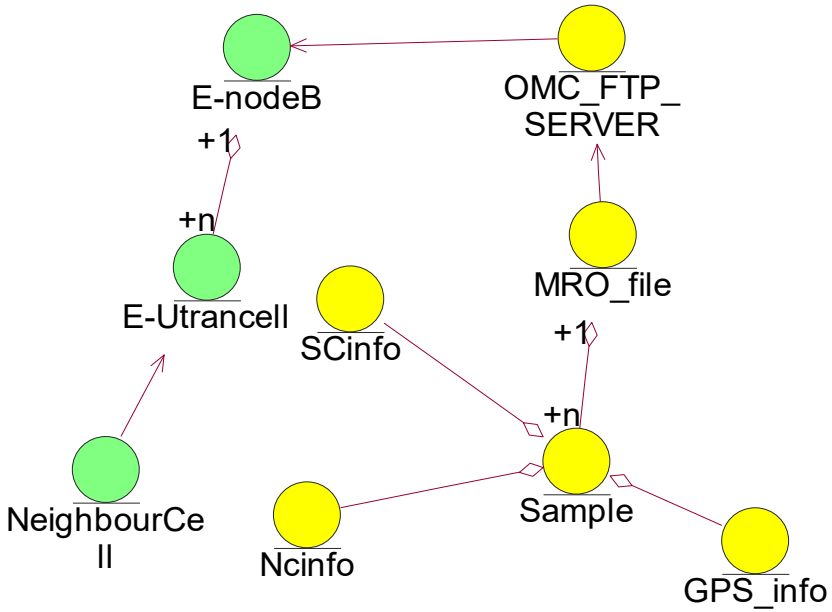
基于OOSE方法的的用例与领域对象

基于MRR的覆盖分析工具



用例模型

Domain object model of the project



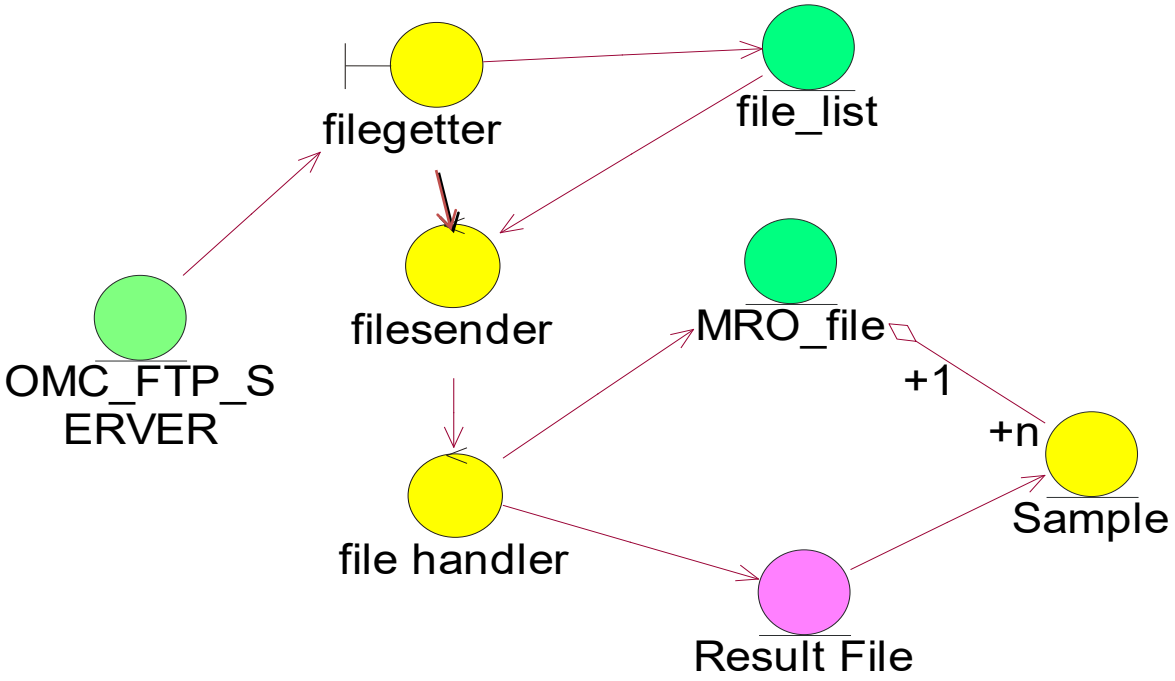
领域对象模型

从用例模型中可见，在三个用例中，hand MRR File 是最复杂的用例，每小时输入**压缩原始XML格式数据 0.1T ，输出数据0.3T，15亿条数据，** 是**关键用例。**

3、项目的分析模型

分析模型提供稳健的对象模型

The analysis model of the project MRO getter

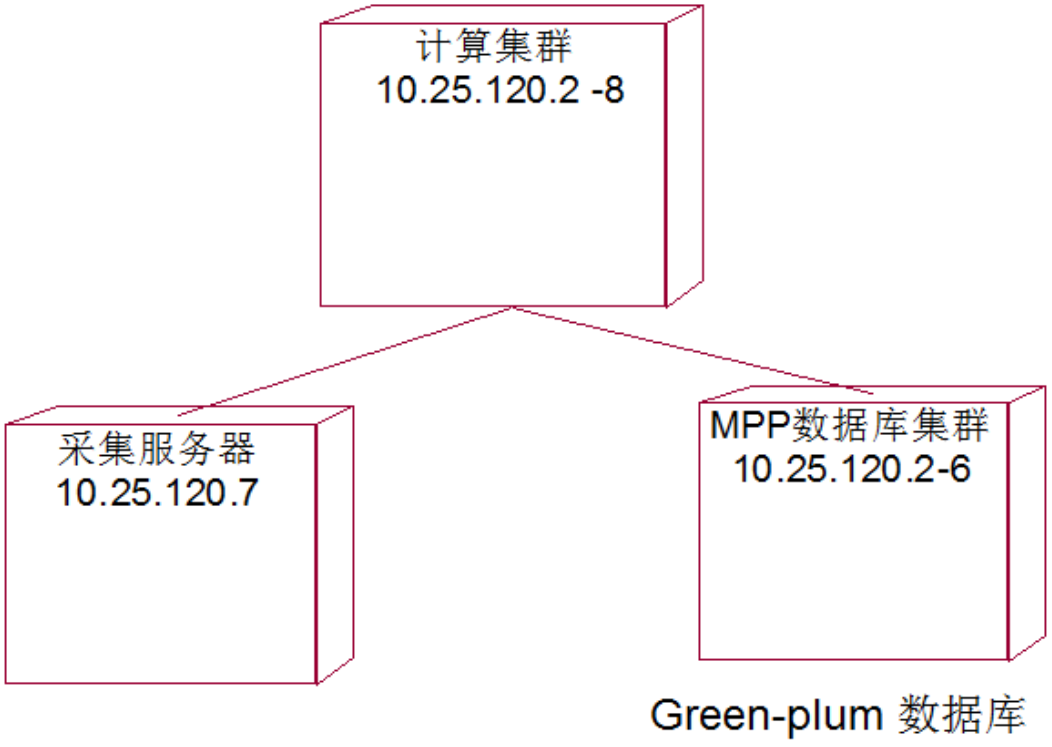


针对最核心的handle MRR file用例 ， 提供一个稳健（Robostness）的对象模型，可以保障处理逻辑的正确性。（不依赖运行环境的影响）

4、部署环境与挑战

利用有13台服务器，突破大数据velocity 难题。

- MRR覆盖分析工具部署环境



Centos 7

主要的挑战：

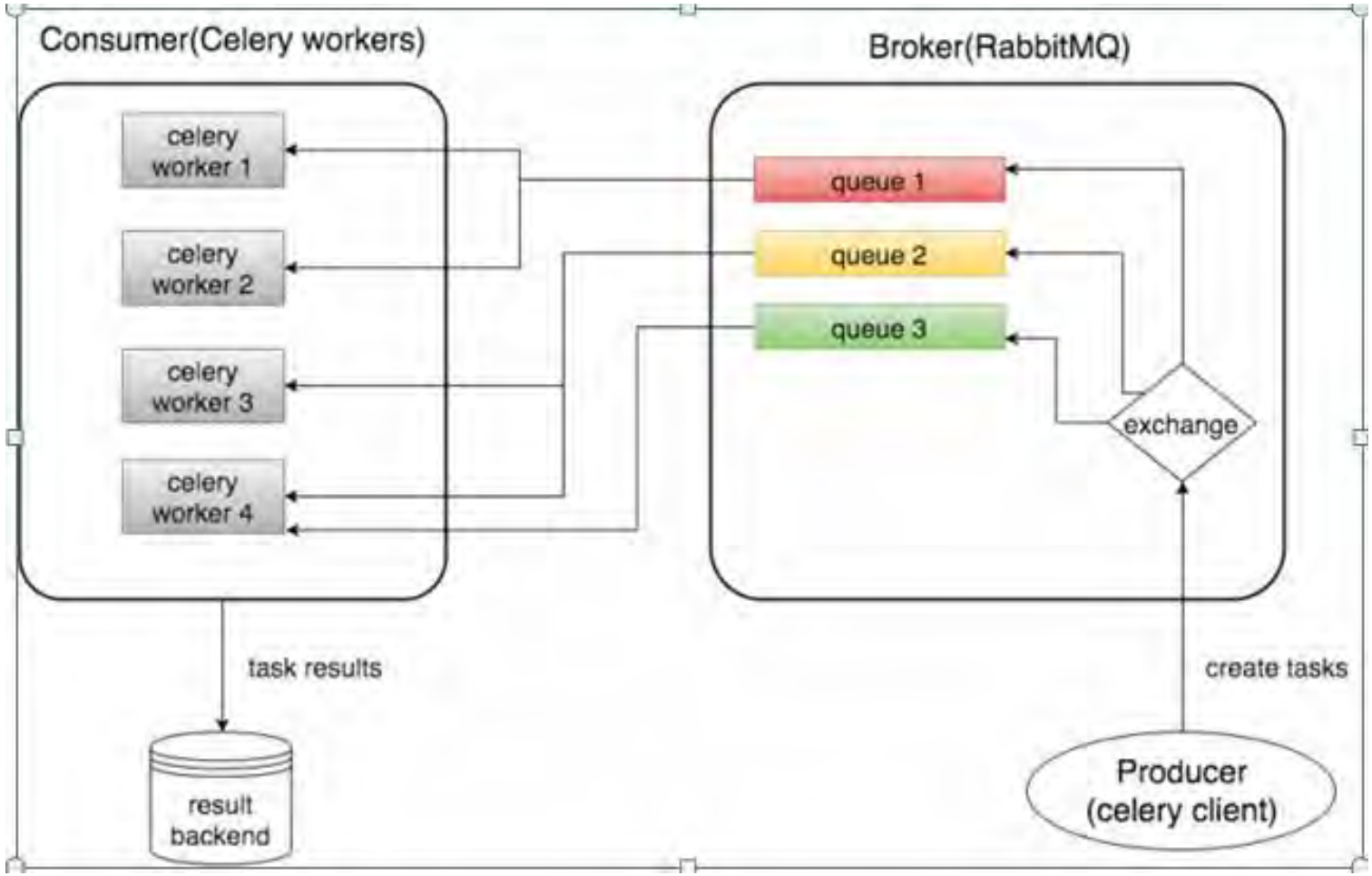
1）分布式计算效率

2）弹性扩展

3）大数据入库

5.1、对策一 基于事件驱动的微服务框架

利用consumer-producer模型，任务的生产者不必关心任务的消费者，获得一种多态（Polymorphism）能力。



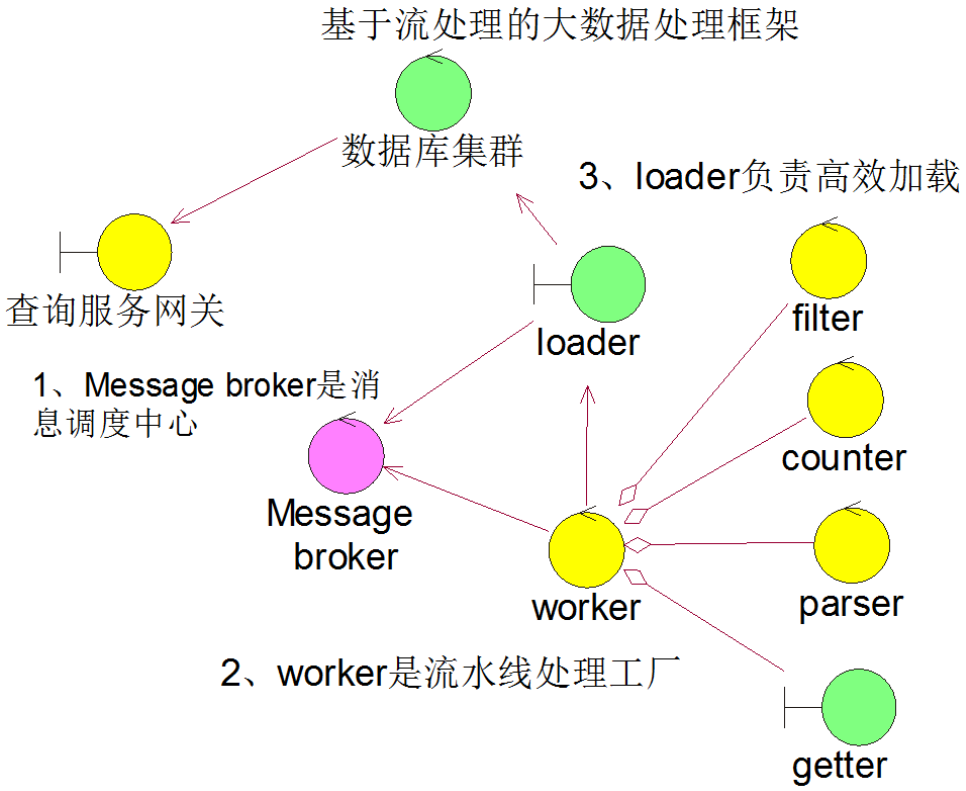
基于celery的事件驱动模型

5.2、基于celery实现高效分布式计算

中国移动 China Mobile

计算集群含7台48核、128G内存物理机。

Hostname	CPU Total/Sys/User (%)	Memory In Use (%)	Disk R (MB/s) | Skew	Disk W (MB/s) | Skew
WORKER01	95.20	30.78	38.40	101.2
WORKER02	79.72	27.37	57.97	107.2
WORKER03	90.49	25.20	48.40	191.2
WORKER04	60.24	24.68	38.86	97.05
MASTER	41.89	5.68	0	7.91



核心特点：

1）流处理框架

2）易于水平扩展

1.2万基站/1 机器

3）完美展现Velocity(快)

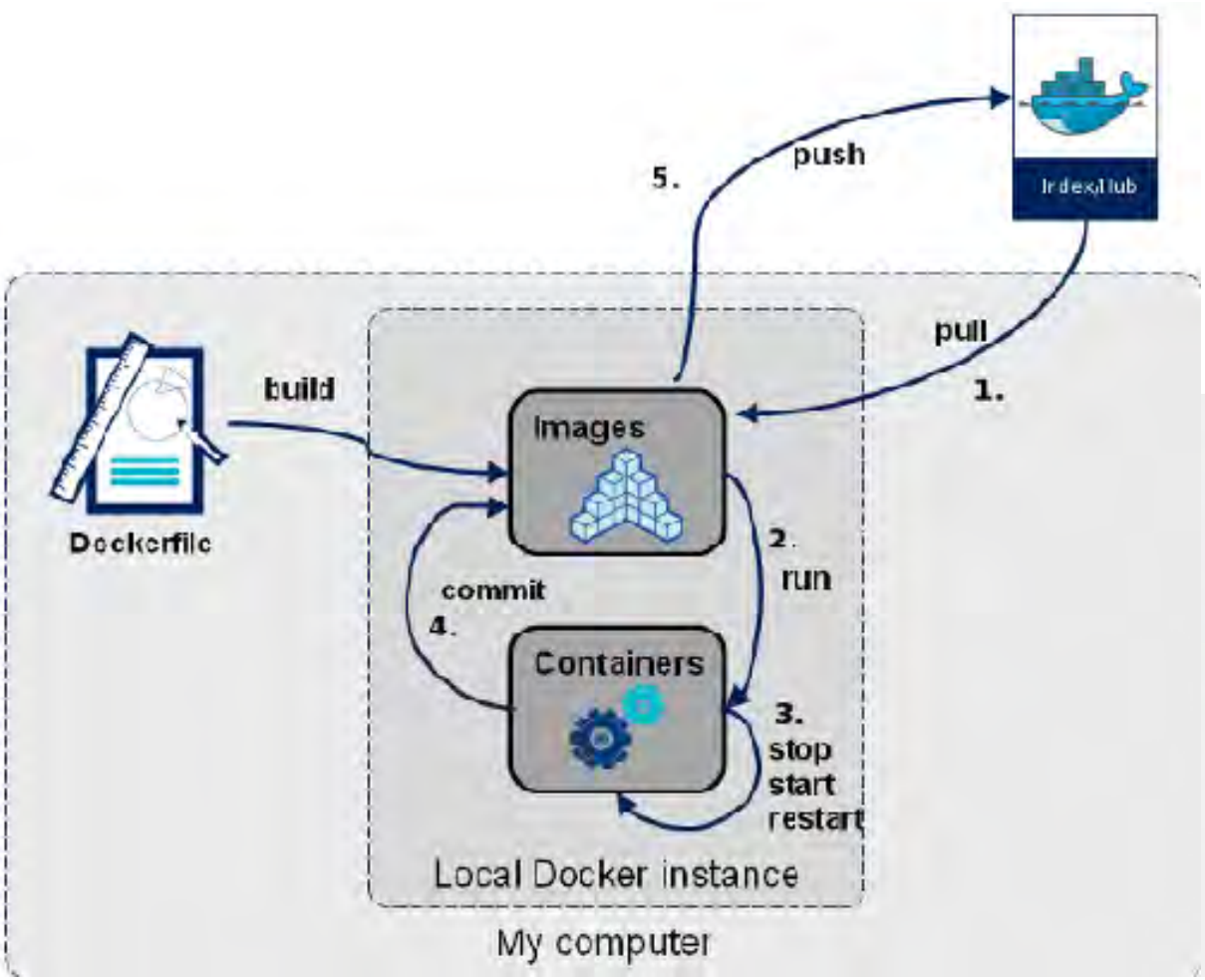
每小时 In 20G，out 60G / 1机器

每秒　1.8万条记录/1CPU

每秒　150万条记录/1集群

6.1、对策二：利用弹性计算平台

大数据多机环境下，应用程序发布，升级是巨大挑战，必须引入轻量化的弹性计算平台：Dockerswarm。



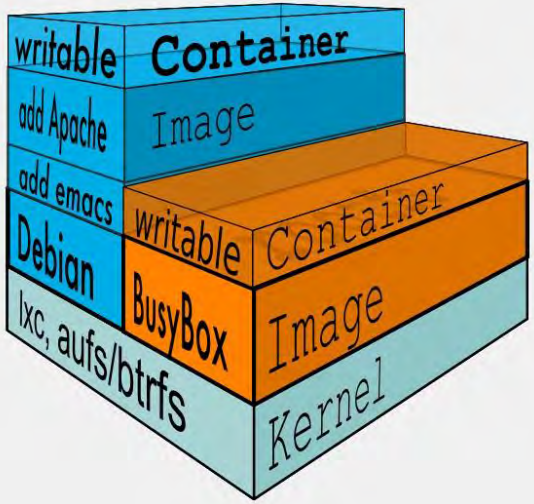
docker运行示意图

主要的优势：

1）节省计算资源

2）本地全栈开发

3）集群扩展方便



Build：创建image

6.2、利用docker swarm 实现一键运维

在管理节点上运行docker stack deploy-c docker-compose.yml 就能够完成集群全部机器的配置工作。

```
services:
  mro:
    # replace username/repo:tag with your name and image
    image: 10.25.226.2:5000/mro
    volumes:
      - "/gpdir/mro:/gpdir/mro"
    deploy:
      #  deploy container nums
      replicas: 18
      placement:
        constraints: [node.role == worker]
      restart_policy:
        condition: on-failure
    depends_on:
      - rabbit
    networks:
      - webnet
```

启动的虚容器数量

依赖的前置作业组件

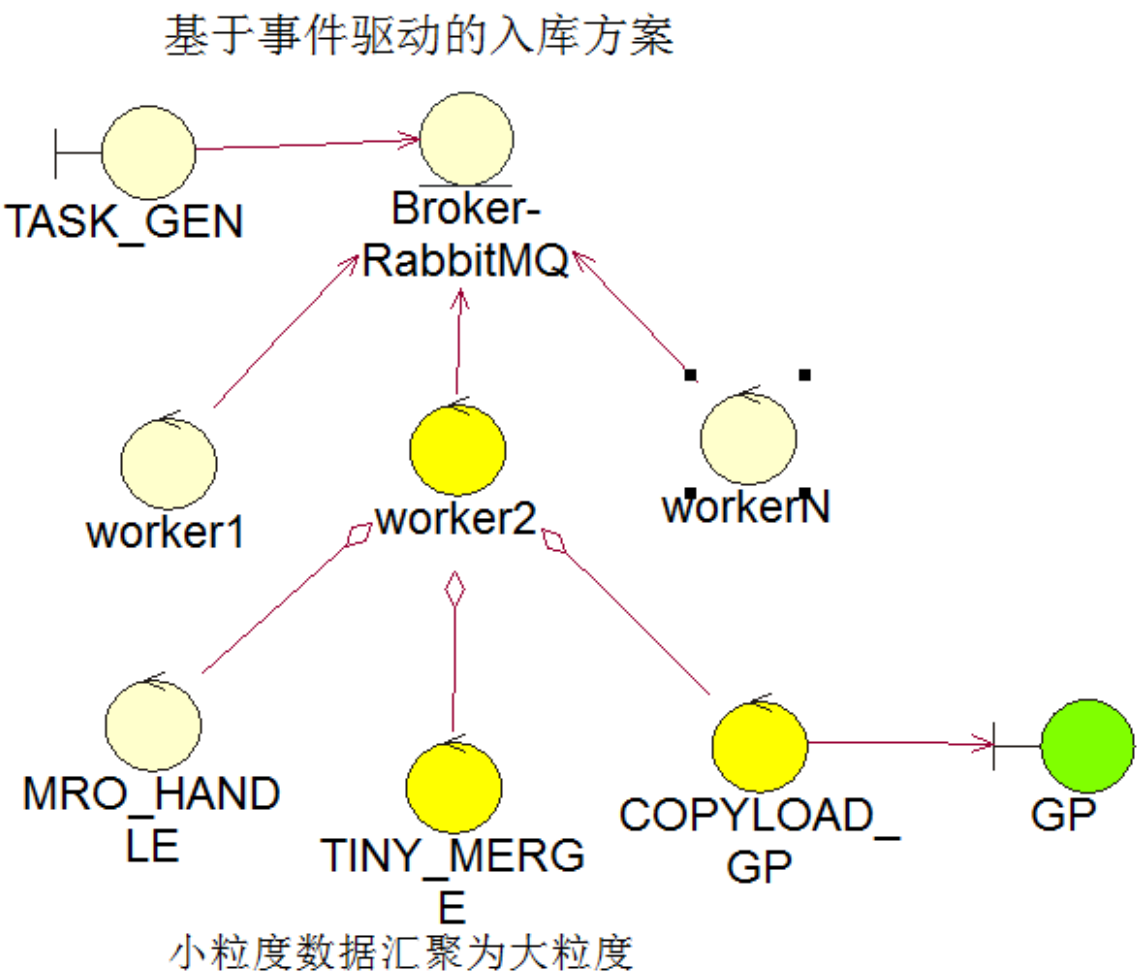
Compose file

Manager节点

Worker节点

7.1、对策三 利用事件驱动实时入库

利用18个弹性容器解决数据处理效率的问题，进一步利用事件驱动的方案实现数据实时入库。



入库优化策略：

1）每个容器实时入库

2）将小粒度数据汇聚

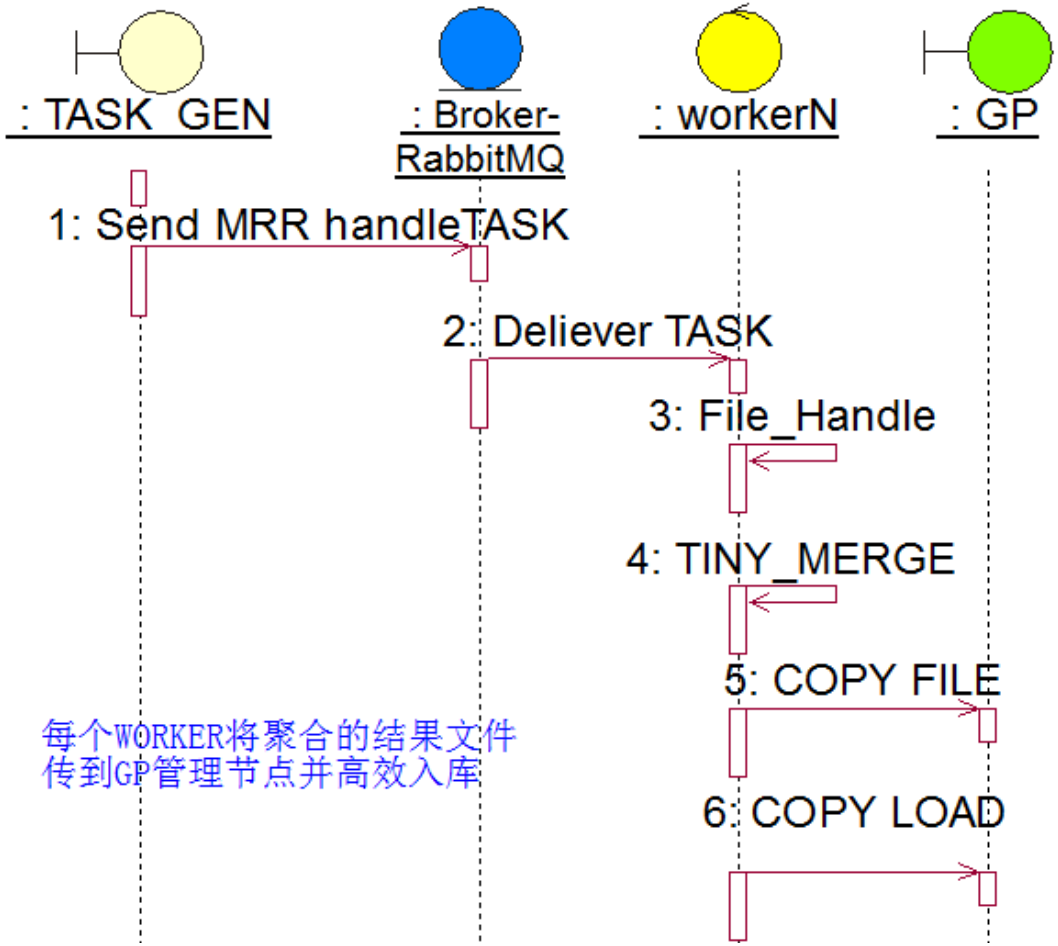
3）采用COPY模式

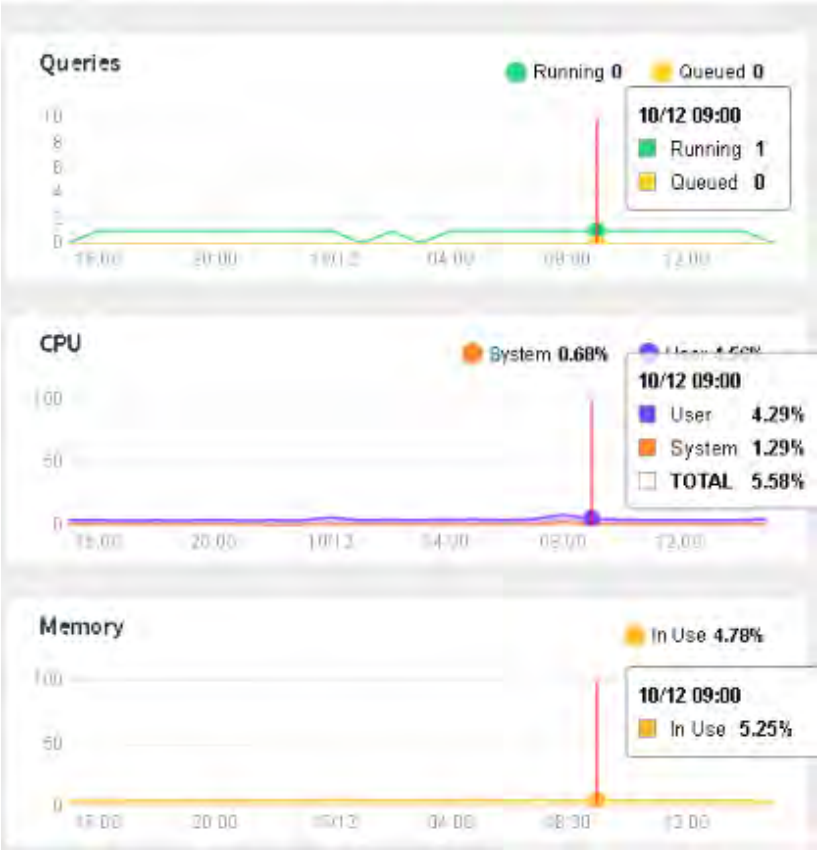
每小时入库：

15亿条数据，0.3 T

7.2、多个容器独立数据入库

多个虚容器独立入库，极大的降低了GP集群的处理负荷。

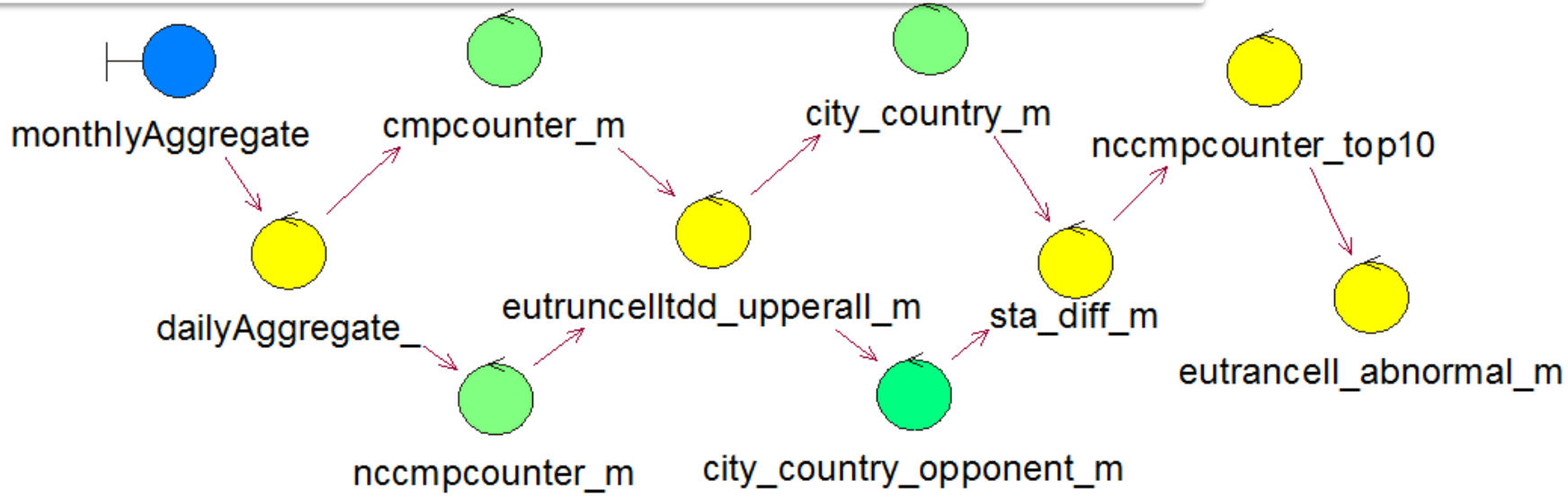




利用GCC监控

8、主要功能1：周期性汇总运算

利用celery任务调度框架，实现灵活汇总任务。




```
@app.task
def dailyAggregate(starttime):
    cmpcounter_chain = chain(cmpcounter_d.s(starttime),eutruncelltdd_upperall_d.s(starttime))
    return  group(cmpcounter_chain,nccmpcounter_d.s(starttime))

@app.task
def monthlyAggregate(starttime):
    mid_group = group(cmpcounter_m.s(starttime), nccmpcounter_m.s(starttime))
    cmpupper_chain = chain( eutruncelltdd_upperall_m.s(starttime),
                            group(city_country_m.s(starttime),
                                  city_country_opponent_m.s(starttime)),
                            sta_diff_m.s(starttime)
                            )
    abnormal_chain = chain(nccmpcounter_top10.s(starttime),eutrancell_abnormal_m.s(starttime))
    return  chain(dailyAggregate(starttime), mid_group, cmpupper_chain, abnormal_chain)
```

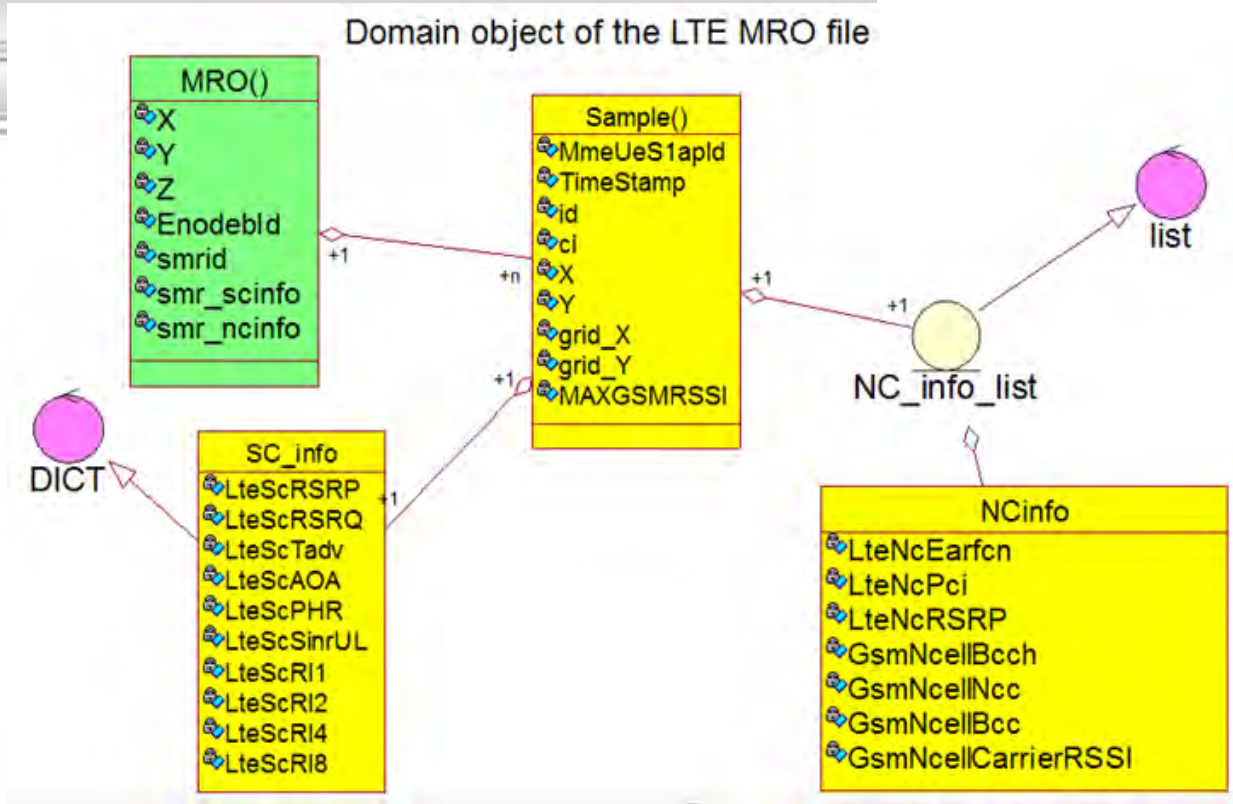
8、主要功能2：自定义数据挖掘

集群数据库（GreenPlum），轻松实现百亿级数据的自定义挖掘。

SQL编辑器 | 图形化查询构造器

前的查询

```
insert into cmpcounter_d
select
'2017-10-12'::timestamp,
enbid,
eci,
sum(rsrp_avg_cmcc),
sum(rsrp_count_cmcc),
sum(rsrp_avg_chun_nume),
sum(rsrp_avg_chte_nume),
sum(rsrp_count_chun),
sum(rsrp_count_chte),
sum(rsrp_weak_cmcc),
sum(rsrp_weak_chun_110),
sum(rsrp_weak_chte_110),
sum(rsrp_weak_chun_113),
sum(rsrp_weak_chte_113),
sum(rsrp_avg_inter_cmcc_nume),
sum(rsrp_count_inter_cmcc),
sum(rsrp_weak_inter_cmcc),
null
from cmpcounter where starttime >= '2017-10-12' and starttime < '2017-10-13'
group by enbid,eci ;
```



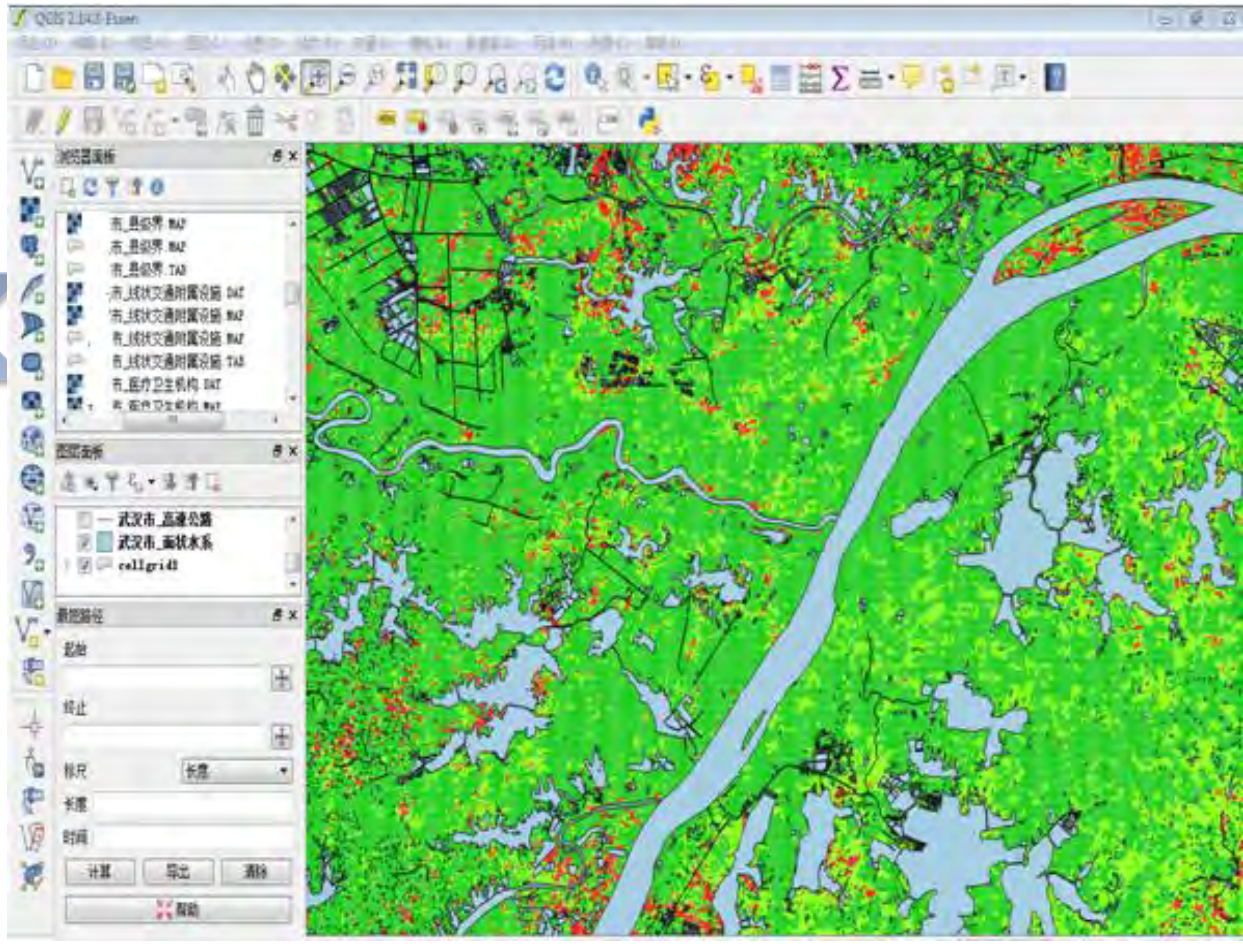
领域对象模型

8、主要功能3：自定义查看栅格级覆盖情况

利用QGIS强大的空间分析功能，可直接装载数据集群中的空间数据，进行自定义查看分析。

结果展示界面
（QGIS实现）

1. **采样点级：70亿（某城市）。**
2. **小区栅格级：1000万**
3. **栅格级：１００万：**

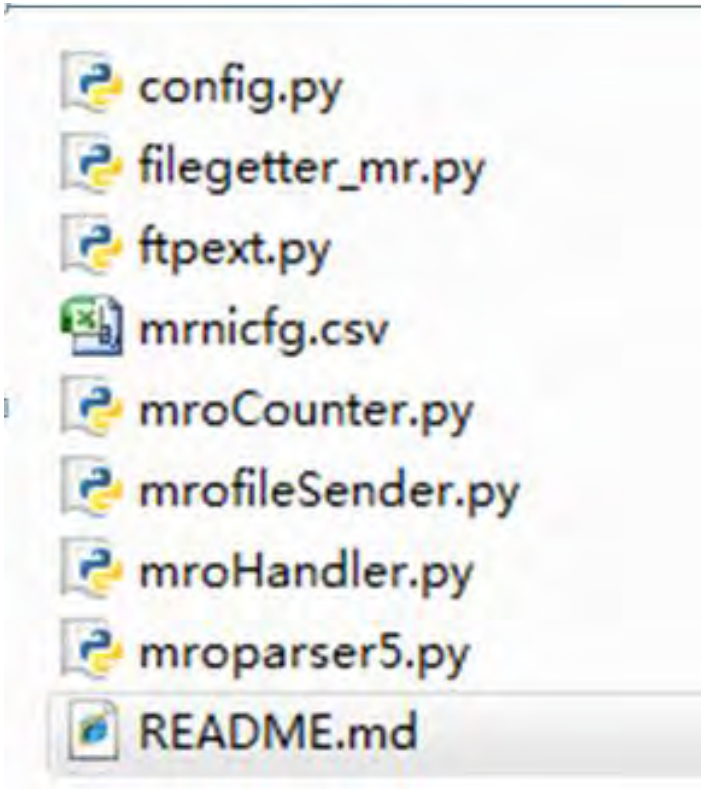


某城市100*100栅格覆盖专题图

目录

1、项目实施总结

目前该项目已经运行半年，经过了3次迭代升级。



低成本解决方案：

1）累计开发45人天；

2）每周运维0.2 人天 ；

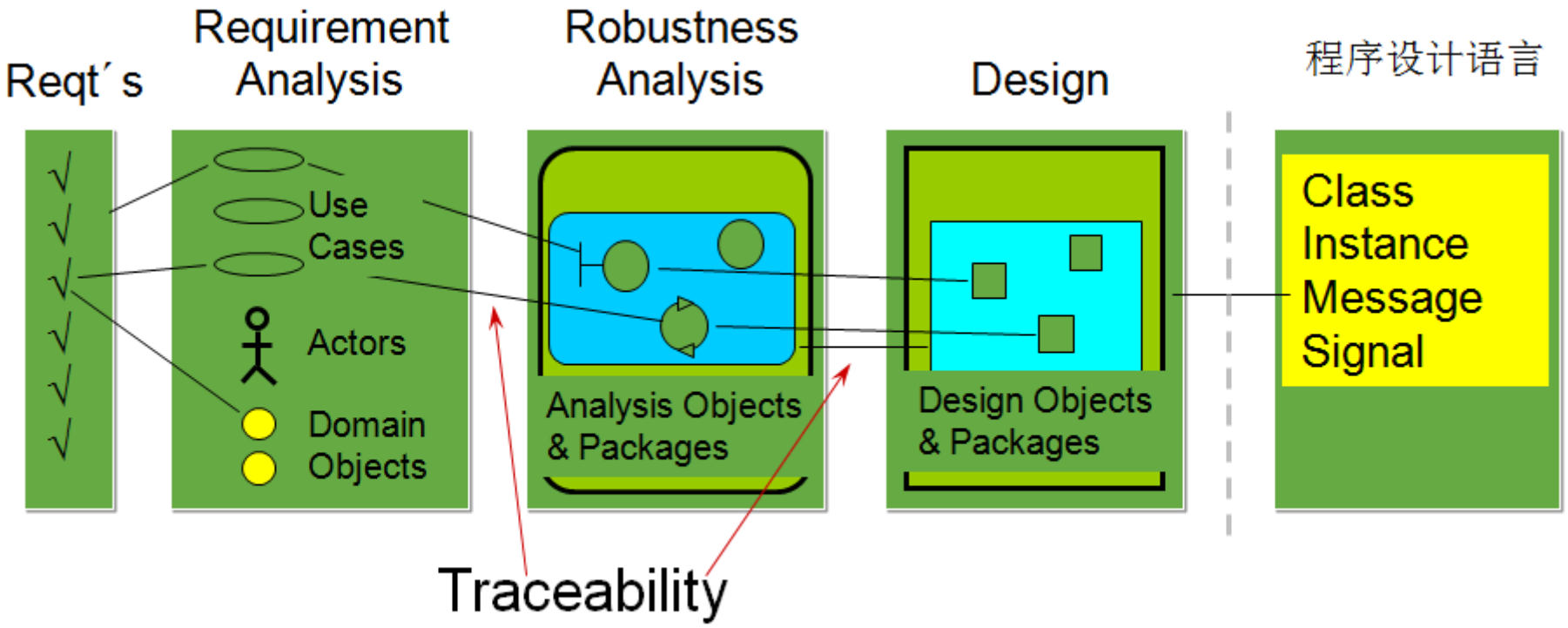
（fabaric）

3）节省开发费用100W；

低成本开发的关键： **系统的软件工程方法+完备的组件库。**

2、关键点1 ：系统的软件工程方法

OOSE的的核心是以需求为起点，构建一个稳健的组件模型，不依赖部署环境限制（ inversion-of-control ）。



可追踪性来克服The Babel Tower问题

OOSE： object oriented software Engineering

3、关键点2：优质的组件库

中国移动 China Mobile

湖北开发工具筛选标准：易于学习、开源扩展、主流高效
借用Postgres-SQL强大的插件能力：实现低成本的大数据敏捷开发

原型	项目	组织
QGIS Trademark **GIS展现：**查看PG图层	 KETTLE **配置式开发：**ETL、整合、一般处理	git **版本管理：**多人共享，分工合作
PostGIS **GIS分析：**通过SQL实现	PYTHON **代码编程：**数学分析、特殊处理	JIRA **缺陷管理：**问题跟踪、持续改善
Pivotal Greenplum **数据仓库：**大数据存储，以PostgreSQL为基础。	Celery **并发调度：**任务编排、消息处理	

4、 Postgres-SQL插件的特点

中国移动 China Mobile

PG的特点是“内核稳定，扩展丰富，生态完整”，非常适合进行低成本的大数据敏捷开发“。

扩展丰富

- PostGIS
- HS-store
- Green-plum

生态完整

Django

Celery+SQLALchemy

Dowker Swarm

5、大数据研究展望

借助PG及其强大的插件，真正发现数据的价值。

1、storage：存的起

2、DATA：看得见

3、information：看得懂。

4、**intelligence：用得出。**

目前攻克“Velocity”难题实现“看得懂”，下一步我们计划**突破“Veracity”难题，实现“用得出”。**

衷心感谢PG社区各位专家一直以来的关心和帮助！

谢谢聆听,敬请指正！