1. 总体架构
   1. 软件服务程序组成

整个服务程序由三个功能模块组成：实时处理模块，预处理模块，后处理模块

* 1. 服务程序对外服务接口



图1 服务程序对外接口示意

* 1. 服务程序各javascript文件功能说明

1. realtime\_process：实时处理程序
2. follow\_process：预处理程序
3. statistics\_process：后处理统计分析程序
4. parse：RTCM包分解
5. rtcm3：RTCM包解析
6. calib：公共常量定义
7. comn：公共函数定义
8. ephemeris：卫星位置计算相关
9. pntpos：最小二乘定位及完好性计算相关
10. ionosphere：电离层延迟计算
11. troposphere：对流层延迟计算
12. nodepos：服务相关数据更新及定位结果对象构建
13. integrity：后处理计算相关
14. 实时处理服务

2.1 实时处理模块对外提供的接口函数是：parser\_pos=**function** (sta\_id,data)

/\*参数说明

输入：sta\_id：测站编号，由系统定义

data：RTCM数据流

输出：JSON格式定义：showJson=**function** () {  
 this.time=cmn.timenow();//在有UTC时间转换参数时输出定位结果的UTC时间，无UTC时间参数时 输出相应系统的定位时间，格式：YYYY-MM-DD hh:mm:ss.ppp  
 this.posR={};//定位结果，输出为javascript对象，对象属性为定位系统编号，GPS：0，GLO：1，BD：2，组合：3，（注意，输出结果不一定包含所有系统，具体值取决于观测数据情况），每个系统定位结果由*posR\_create*()定义。  
 this.satR=[];//实时卫星观测数据信息数组，每个数据元素是一颗卫星观测数据javascript对象，由*satR\_create*()定义

}

*posR\_create*=**function** () {  
 **this**.**stat**=0;//定位结果指示，0为定位结果不可用，大于0表示可用

**this**.**week**=0;//观测时间周，对应导航系统，GLO为0  
 **this**.**tow**=0;//观测时间周内秒，对应导航系统，GLO为0  
 **this**.**time**=0;//观测时间年月日形式，对应导航系统，GLO对应UTC时间（跳秒修正到20170101）  
 **this**.**Lat**=0;//定位结果：纬度  
 **this**.**Lon**=0;//经度  
 **this**.**H**=0;//高程  
 **this**.**posNum**=0;//参与定位卫星数量

**this**.**trackNum**=0;//跟踪卫星数  
 **this**.**exsats**=**""**;//RAIM最大可能故障卫星  
 **this**.**HDOP**=0;  
 **this**.**VDOP**=0;  
 **this**.**dH**=0;//实时定位水平误差  
 **this**.**dV**=0;//实时定位垂直误差  
 **this**.**Hc**=0;//到当前时刻定位结果95%概率水平定位精度  
 **this**.**Vc**=0; //到当前时刻定位结果95%概率垂直定位精度  
 **this**.**HPL**=0;//RAIM计算水平保护水平  
 **this**.**VPL**=0;//RAIM计算垂直保护水平  
 **this**.**navsys**=[]//参与定位的导航系统  
}

*satR\_create*=**function** () {  
 **this**.**sys**=0;//卫星所属导航系统  
 **this**.**sat**=0;//卫星编号，对应导航系统  
 **this**.**week**=0;//观测时间周，对应导航系统  
 **this**.**tow**=0;//观测时间秒，对应导航系统  
 **this**.**utc**=**""**;//观测时间年月日形式，时间值对应导航系统  
 **this**.**svh**=1;//卫星健康信息  
 **this**.**ELe**=0;//卫星仰角  
 **this**.**Azi**=0;//卫星方位角  
 **this**.**SNR**=[0,0];//信号载噪比，双频  
 **this**.**ura**=0;//卫星URA  
 **this**.**rura**=0;  
 **this**.**udre**=0;  
}

\*/

2.2 实时处理设置更新函数：procset=**function** (sta\_id)

参数说明：sta\_id //测站编号，在调用该函数前需要确定配置文件服务已经更新，如果定位配置文件有了更改，需通知定位处理程序更新配置信息。

2.3 实时处理删除配置函数：delconfig=**function** (sta\_id)；

当测站对应编号不在需要时，可调用此函数注销该站配置文件。

2.4 实时定位配置文件读取函数

**function** *getProopt*(sta\_id)，当前函数从本地文件中读取配置文件，如果从配置文件服务器取配置文件，只需要将该函数替换，保证返回的配置文件对象与从文件中读取的一致就可。

2.5 实时处理调用流程

实时定位处理，只需要直接调用入口函数就可以，返回结果为实时定位处理javascript对象，实时处理函数调用过程如下：



图2 实时定位处理调用流程

1. 预处理服务

预处理服务必须先调用初始化函数，然后在输入原始RTCM数据进行定位

3.1 预处理初始化函数：procinit=**function** (sta\_id,bt,et,len)

/\*参数说明

输入参数：sta\_id，测站编号

bt，预处理开始时间，年月日时分秒数组形式，北京时间，例如[2017,4,27,0,0,0],注意javascript·的月份从0开始。

et，预处理结束时间，数组形式，北京时间，例如[2017,4,28,23,59,59]，注意每天定时处理时前后两天时间设置不要有时间重复

len，预处理每次输入的RTCM最大字节长度。

输出：执行成功返回0，不成功返回大于0的数

\*/

注意，预处理中，也需要读取配置文件，读取方式与实时定位处理一致

3.2 预处理入口函数：parser\_pos=**function**(data)，输入参数为RTCM数据

3.3 预处理输出参数格式定义

输出参数由函数构建的javascript对象：**function** *logOutJson*() {  
 **this**.**time**={};//定位时间，相应导航系统对应的时间，格式采用服务程序定义的统一时间{“time”:0,”sec”:0},由自1970年1月1日后秒数定义  
 **this**.**posR**={};//定位结果，由**function** *posResult*()定义，每个定位成功的导航系统一个对象，与实时定位结果类似  
 **this**.**obsR**=**new** Array();//观测数据数组，每个元素是一颗卫星，由**function** *obscreate*()定义  
 **this**.**ephs**={**"eph"**:**new** Array(),**"ceph"**:**new** Array(),**"geph"**:**new** Array()};//广播星历数据，有更新时有数据，eph：GPS星历，ceph：BD星历，geph：GLO星历，  
 **this**.**alms**={**"alm"**:**new** Array(),**"calm"**:**new** Array()};//历书数据，alm：GPS历书，calm：BD历书  
 **this**.**ions**={**"ion"**:**new** Array(),**"cion"**:**new** Array()};//电离层模型参数，ion：GPS，cion：BD  
 **this**.**uras**={**"ura"**:**new** Array(),**"cura"**:**new** Array()};//星历精度  
 **this**.**utcs**={**"utc"**:**new** Array(),**"cutc"**:**new** Array()};//UTC时间转换参数  
 **this**.**udre**=**new** Array();  
 **this**.**rura**=**new** Array();  
}

**function** *posResult*() {  
 **this**.**stat**=0;*//定位结果状态* **this**.**week**= 0;*//定位时间GPS周* **this**.**tow**= 0;*//定位时间GPS周内秒* **this**.**time**= **""**;*//定位结果对应的年月日时间* **this**.**X**= 0;*//定位结果，ECEF坐标* **this**.**Y**= 0;  
 **this**.**Z**= 0;  
 **this**.**dX**= 0;*//定位误差，本地坐标系下水平东向* **this**.**dY**= 0;*//北向* **this**.**dZ**= 0;*//垂向* **this**.**Lat**= 0;*//定位结果纬度* **this**.**Lon**= 0;*//定位结果经度* **this**.**H**= 0;*//定位结果高程* **this**.**GDOP**= 0;*//几何精度GDOP* **this**.**PDOP**= 0;*//* **this**.**HDOP**= 0;  
 **this**.**VDOP**= 0;  
 **this**.**VPL**= 0;*//定位垂直保护水平* **this**.**HPL**= 0;*//定位水平保护水平* **this**.**posNum**= 0;*//定位卫星数* **this**.**trackNum**=0;//当前跟踪卫星数  
 **this**.**exsats**= **""**;*//定位排除的卫星* **this**.**minEl**= 0;*//最小卫星仰角* **this**.**navsys**=[];*//定位卫星系统*}

**function** *obscreate*() {  
 **this**.**sys**=0;*//卫星所属导航系统* **this**.**sat**=0;*//卫星PRN号* **this**.**week**=0;*//GPS周* **this**.**tow**=0;*//周内秒* **this**.**time**=**new** ca.gtime();*//观测数据GPS时间服务定义时间* **this**.**P**=[0,0,0];*//伪距1频点2频点* **this**.**L**=[0,0,0];*//载波相位* **this**.**D**=[0,0,0];*//多普勒* **this**.**S**=[0,0,0];*//载噪比* **this**.**Azi**=0;*//方位角* **this**.**Ele**=0; *//仰角*}

预处理的基本过程与实时定位处理一致，只是输出数据格式稍有不同，

1. 后处理服务

4.1后处理输入参数定义

是直接从预处理输出的定位结果取得数据统计定位性能，后处理统计前需从前端获得统计参数选项，为一个javascript对象，由statis\_create()定义：

**function** *statis\_create*() {  
 **this**.**id**=0;//测站编号  
 **this**.**bt**=cmn.timenow();//统计开始时间  
 **this**.**et**=cmn.timenow();//统计结束时间  
 **this**.**hist**={}; *//统计结果一些表示格式，每个对象对应导航系统，由hist\_create*()定义**this**.**option**={};*// 需要统计处理的选项，每个属性对应导航系统，statis\_option\_create*()定义}

*输入参数对象的定义实例如函数***function** *satis\_init*(para) {  
 para.**id**=0;  
 para.**bt**=cmn.epoch2time([2017,4,28,11,45,0]);  
 para.**et**=cmn.epoch2time([2017,4,28,13,10,30]);  
 para.**hist**[ca.**SYS\_GPS**]=**new** *hist\_create*();  
 para.**hist**[ca.**SYS\_GLO**]=**new** *hist\_create*();  
 para.**hist**[ca.**SYS\_CMP**]=**new** *hist\_create*();  
 para.**hist**[ca.**SYS\_ALL**]=**new** *hist\_create*();  
 para.**option**[ca.**SYS\_GPS**]=**new** *statis\_option\_create*();  
 para.**option**[ca.**SYS\_GLO**]=**new** *statis\_option\_create*();  
 para.**option**[ca.**SYS\_CMP**]=**new** *statis\_option\_create*();  
 para.**option**[ca.**SYS\_ALL**]=**new** *statis\_option\_create*();  
 *option\_init*(para.**option**[ca.**SYS\_GPS**]);  
 *option\_init*(para.**option**[ca.**SYS\_GLO**]);  
 *option\_init*(para.**option**[ca.**SYS\_CMP**]);  
 *option\_init*(para.**option**[ca.**SYS\_ALL**]);  
}

**function** *hist\_create*() {  
 **this**.**section**=0.1;//误差，DOP值统计图横坐标间隔，默认值0.1  
 **this**.**vert\_axis**=1;*//误差，DOP值统计图纵坐标，0表示百分比，1表示点数* **this**.**lastlen**=86400;//超限时间片统计最长时间长度，默认24小时  
 **this**.**failure**=opt.alertTime;//连续性最大持续时间，默认10秒  
 **this**.**HAL**=opt.HAL;//完好性水平告警限  
 **this**.**Her**=opt.HAL;//水平误差  
}

**function** *statis\_option\_create*() {  
 **this**.**sat\_hist**=0;//标示是否统计卫星数，0表示不统计，1表示统计  
 **this**.**err\_hist**=0;//统计定位误差（水平，垂直）  
 **this**.**dop\_hist**=0;//统计DOP值（HDOP，VDOP）  
 **this**.**PL\_hist**=0;//统计完好性（HPL,VPL）  
 **this**.**acc95**=0;//计算95%定位精度（水平，垂直）  
 **this**.**slice**=**new function** () {//表示是否统计误差，DOP值区间  
 **this**.**sat\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};//flag，是否统计标示，0不统计该项，1统计该项；**extre\_min，**最小值；**extre\_max**最大值。目前此项目前不要求，不用处理，只是备用  
 **this**.**her\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 **this**.**ver\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 **this**.**hdop\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 **this**.**vdop\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 **this**.**hpl\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 **this**.**vpl\_num**={**"flag"**:0,**"extre\_min"**:0,**"extre\_max"**:0};  
 };  
 **this**.**up\_slice**=**new function** () {//误差，完好性等超限时间统计  
 **this**.**sat\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};//flag，表示是否统计，0不统计，1统计，**up\_min，最小误差，up\_len，最少持续时间**  
 **this**.**her\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 **this**.**ver\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 **this**.**hdop\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 **this**.**vdop\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 **this**.**hpl\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 **this**.**vpl\_num**={**"flag"**:0,**"up\_min"**:0,**"up\_len"**:0};  
 };  
}

4.2 后处理数据输入

输入接口函数statistic\_data=**function** (data)

data为输入参数，其数据来源于预处理后保存的定位结果数据，其数据格式定义如下：输入定位数据格式，为javascript对象，定义格式如下{**"time"**:log.time,**"data"**:log.posR}，其中log.time是定位时间，格式为服务程序内部定义格式，取预处理输出结果的时间，log.posR是定位结果数据，posR就是预处理输出的定位结果对象。

4.3 后处理结束，统计结果返回函数statistic\_get=**function (**)，返回结果为javascript对象，结构由函数*sta\_result*()定义，输出结果每一项与输入参数每一项一一对应

**function** *sta\_result*() {  
 **this**.**sat\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**herr\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**verr\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**hdop\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**vdop\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**hpl\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**vpl\_hist**={**"X"**:[],**"Y"**:[]};  
 **this**.**acc95\_h**=**new** *acc95\_create*();  
 **this**.**acc95\_v**=**new** *acc95\_create*();  
 **this**.**integrity**=[];//完好性百分比，表示不超限定位结果占所有数据百分比  
 **this**.**continuity**=1.0;//非精密进近连续性，输出为小数，1表示100%  
 **this**.**availability**=1.0;// 非精密进近可用性,输出为小数  
 **this**.**slice**=**new function** () {  
 **this**.**sat\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**her\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**ver\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**hdop\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**vdop\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**hpl\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**vpl\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 };  
 **this**.**up\_slice**=**new function** () {  
 **this**.**sat\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**her\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**ver\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**hdop\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**vdop\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**hpl\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 **this**.**vpl\_num**={**"flag"**:0,**"X"**:[],**"Y"**:[]};  
 };  
}

1. 测试实例

实时定位处理测试程序文件：test\_real，

预处理测试程序文件：test\_follow

后处理测试程序文件：post\_file