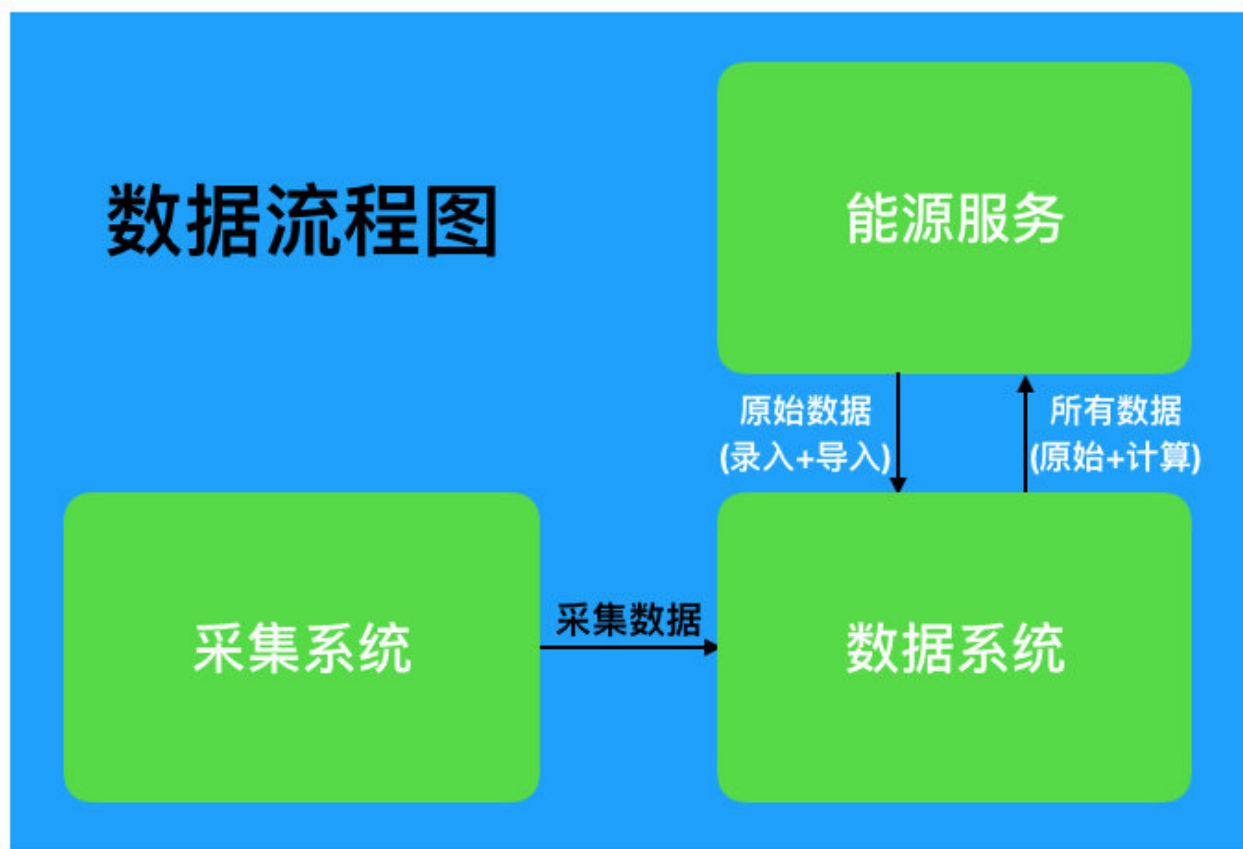


能源系统中存储着很多数据，如采集接入的能源数据，手工录入的能耗数据、产量数据，基于这些原始数据根据一定的规则计算得到的计算数据等。这些数据每天、每小时，甚至每几分钟都在更新，数据量大，计算规则复杂，又要能灵活的配置，适应各种业务场景。

数据，是能源管理系统中最重要的部分，数据的准确性、可靠性、高效性、稳定性、实时性、灵活性、透明性，是数据系统设计的目标。

1. 数据系统的定位

在整个能源管理系统中，数据系统处于怎样的位置？



在整个系统中，和数据相关的有3个子系统：采集系统、数据系统 和 能源服务。

- 采集系统：负责各种数据对接到我们的数据系统，如采集器采集上来的、企业生产系统通过工业协议（modbus、bacnet、opc等）对接上来的、第三方系统对接上来的等。
- 数据系统：负责数据的建模、存储、计算、供应。
- 能源服务：负责将「用户」手工录入 或 文件导入的原始数据，存入「数据系统」，并将各种「业务模型」组装成「数据系统」提供的「通用模型」，做成「配置」存入「数据系统」，并将数据系统的数据，以合宜的方式展示给用户。

2. 要解决的问题

数据系统要解决哪些问题呢？

- 对数据统一建模：以目前接入的得力纺织为例，有采集器采集来的数据，有用户手工修正数据，有

导入的产品产量数据，有各种业务模型建模出的计算数据等。这些数据的来源不同、方式不同、格式不一、模型也是多种多样，要对这些数据进行统一建模，以统一的模型对外提供数据；

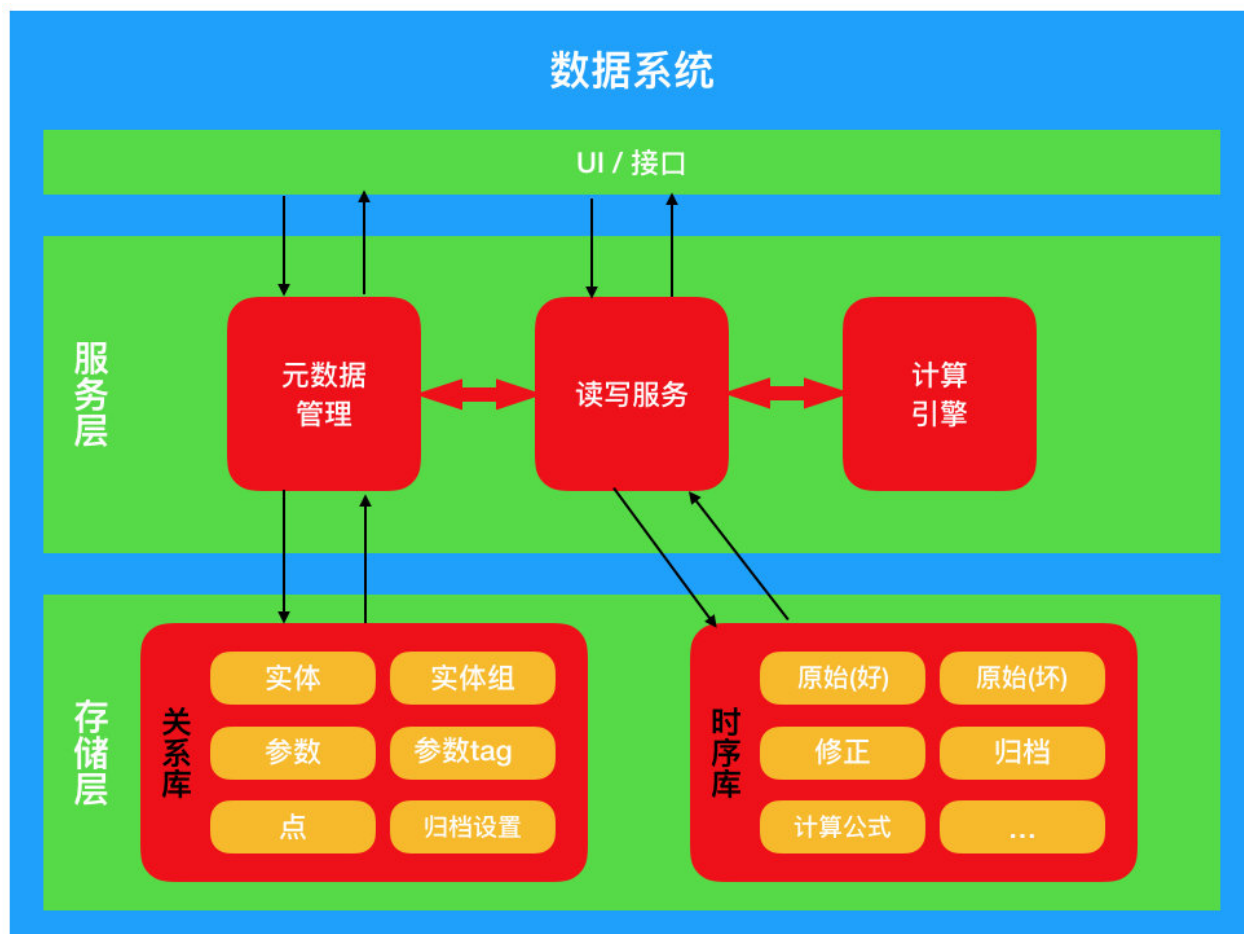
- 海量数据的存储：得力纺织有324个传感器、1万多个数据点位，每10分钟上来一批采集数据，还有页面按天录入的各种产品产量数据等，基于这些数据，计算得到各种用户关心的「计算数据」。这些数据还只是1家企业的，如果类似的企业接入100家，数据存储10年，数据量可想而知；
- 计算：原始数据用户通过一些方式自己也能拿到，但原始数据自身并没有多少参考价值。我们系统的核心，便是通过将「原始数据」按照一定的模型建模，将数据的价值挖掘出来，为企业提供有参考价值的数据。
- 供应：数据存起来不是目的，算出来也不是目的，其目的是将这些数据变成价值，展示给用户，能快速高效的查询任意数据，并保证数据的 真实、准确、稳定、实时。

3. 业务需求

下面是梳理出来的业务需求清单。

1. 所有的原始数据，要全部保存、不能修改、永远保存不删除；
2. 数据要有「可追溯性」，即这个数据「从哪儿来」，「计算数据」要有每一步的「计算规则」和「中间值」；
3. 对数据按照「业务概念」抽象出类型，称为「参数」，如A相电流，天用电量；
4. 数据分3类：瞬时值、期间值、持续值；
5. 瞬时值为数据采集当时的值，如电流，或基于瞬时值计算出反应当时信息的值，如负载率；
6. 期间值有小时、天、周、月、季、年；
7. 持续值是指在这段时间内，都使用这一个值，如单价；
8. 期间支持自定义时间段，如天支持8:00~8:00，8:00~16:00，月支持5.4~6.2；
9. 值要有单位，否则一个数值没有意义；
10. 同样是电流，可能会有mA、A、KA等多个单位；
11. 值在显示时，要有精度，保留到小数点后几位；
12. 每个点可以自己定义精度，如果没有定义，则用参数中定义的默认精度；
13. 参数可以根据各式各样的标签找到，如A相电流 可以根据「电流」标签找到；
14. 参数的标签可以在「参数管理」页面增加、删除；
15. 原始数据没有质量码的，全部认为是「好」的；
16. 原始数据有质量码的，只有「好」的能参与运算，不好的可以查看；
17. 数据计算模型有时效性，如部门增加线路、减少线路；
18. 数据要定期归档，要能够查归档时的数据值；
19. 当出现换表时，用户设置一个认为合理的小时值，跨换表时间的期间值要能正常计算；
20. 当数据出现异常时，用户可以手工修正，以修正值为事实上的数据；

4. 整体架构



数据系统整体分3层：UI/接口层、服务层、存储层。

- UI/接口层：提供页面需要的「HTTP服务」和 能源服务系统需要的「数据接口」
- 服务层：负责逻辑处理，包括元数据管理、读写服务 和 计算引擎
- 存储层：实际落地到磁盘中的结构

5. 模型设计

思路：

- 使用「实体」+「参数」+「时间粒度」定义一个「点」，「点」+「时间」定义一个「数值」，「数值」分为「好」和「不好」，「不好」的有「质量码」表示「错误信息」；
- 只存储原始数据，不存储计算数据；
- 通过「偏移量」处理换表；
- 通过配置有时效性的「计算规则」，支持「时效性」的计算数据；
- 通过配置「自定义期间」，支持用户个性化的「统计周期」；
- 「修正表」和「归档表」数据存储结构和原始数据类似；
- 优先读「修正表」保证 在有修正数据的情况下，使用修正值；
- 通过调用参数区别是否查归档数据

5.1 实体 (Entity)

实体定义：

一个实际存在的主体单位，是「点 (Point)」的「主体 (Owner)」。如采集线路、部门等。

因为有时效性，创建后普通用户不能删除，如确认是建错了，需要管理员删除。点也有类似的特性。

字段	名称	类型	描述	备注
id	ID	String	惟一标识，有可读性	
name	名称	String	客户可读的名称	
customId	终端客户ID	String	组织机构代码	可以放到group中
points	点集合	Set	该实体下面挂的所有「点」	
group	实体组	EntityGroup	该实体在哪个实体组中，用来获取自定义「统计周期」	

5.2 参数（Parameter）

参数定义：

参数是业务上描述数据的概念，一个孤立的「数值」是没有意义的，「数值」+「参数」组成的「数据」则可以「自解释」。

表示同一业务含义，单位不同的，有几个单位则建几个参数。

字段	名称	类型	描述	备注
id	ID	String	惟一标识，有可读性	
name	名称	String	客户可读的名称	
type	类型	enum	INSTANT：瞬时值， PERIOD：期间值， CONTINUOUS：持续值	
unit	单位	String	单位	
dataType	数据类型	enum	BOOLEAN：开关 DOUBLE：所有数值 STRING：字符串	可能会扩展
precision	精度	Integer	小数点后的位数	
periods	可用期间	Set	HOUR, DAY, WEEK, MONTH, SEASON, YEAR	辅助功能，不强制 如一个参数只有 月、季、年3个时间粒度
tags	所有标签	Set	该参数所有的标签	

5.3 点（Point）

点定义：

点是「实体（Entity）」下的一个「数据项」，是描述数据的最小单位，其由「实体（Entity）」、「参数（Parameter）」和「期间类型」（如有）组合定义。

点的另一个常用名称是「位号」，其意义是等价的，其加「时间」对应一个「数字」，但点上有「参数」，使得点上的「数字」变成有意义的「数据」。

字段	名称	类型	描述	备注
id	ID	String	惟一标识，有可读性	
name	名称	String	客户可读的名称	
type	类型	enum	RAW：原始数据， CALC：计算数据	
entity	实体	Entity	该点属于哪个实体	
parameter	参数	Parameter	该点关联的参数	
periodType	期间类型	enum	参数类型为PERIOD时有效	可以不在参数的periods集合中
precision	精度	Integer	小数点后的位数，如配置了，则用这个	

5.4 点数据（PointData）

点数据定义：

点数据，描述了一个具体的数值，是某个「点（Point）」在某个「时间」的「数值」为多少，其「质量码」是怎样的。

字段	名称	类型	描述	备注
entityId	实体ID	String		
parameterId	参数ID	String		
periodType	期间类型	PeriodType		
time	时间	LocalDateTime	时间精确到秒	
value	数值	T	根据具体配置，数据类型可变	
quality	质量码	enum		

5.5 计算公式（CalculateExpression）

字段	名称	类型	描述	备注
entityId	实体ID	String		
parameterId	参数ID	String		
periodType	期间类型	PeriodType		
time	时间	LocalDateTime	时间精确到秒	
expression	公式	String	公式，只描述一层	

6. 计算设计

「点」按照「类型(type)」分为「原始点」和「计算点」，「原始点」是指采集器采集、手工录入 和 第三方数据对接等方式过来的原始数据，「计算点」是指由「原始点」经过「公式(expression)」计算出来的。

6.0 计算的业务输入

- 按照有和业务建模相关的，有和业务建模无关的
 - 无关的：负载率
 - 相关的
 - 和建模相关的：如一个部门下面挂几条线路
 - 和时间相关的：如一个部门在5.1号是3条线路，5.1号之后是5条
- 瞬时值转为期间值的类型
 - 用量相关的
 - 尾-头
 - 子的加起来，如部门月用电量，要用天加起来，不能用几条线路加，原因是支持时效性
 - 最大值、最小值、平均值
 - 环比，同比
 - 跨期间的计算
 - 计算一个天值，用到月平均值
- 期间值的业务类型
 - 用量
 - 效率？
 - 环比？
 - 环比，做成一个方法，可以直接调用
- 计算的思路
 - 所有数据读出来，一次算
 - 效率高，性能好
 - 表现力有限
 - 每次算一个点，递归

- 效率差，算的慢
 - 表现力好
 - 如果能两种结合起来，就更好了
- 例子
 - 计算部门年用电量
 - 计算某月用电量 与 上月 环比

环比的问题：

环比不是一个参数，它是累积量都有的一个方法，做到DataPoint里面，增加一个方法即可。

6.1 基本思路

按照「期间类型(periodType)」来分，「计算点」有2类：「瞬时值(INSTANT)」和「期间值(PERIOD)」，「期间值」细分出来有时、日、周、月、季、年、班组等。

注：计算点没有「持续值(CONTINUOUS)」这一类型。

6.1.1 计算的要素

计算的要素有：结果值、公式、依赖值。

- 结果值：计算点 某一时间的值，包括「时间」和「值」；
- 公式：描述计算规则，有时效性，不同时间段可以用不同的公式；
- 依赖值：由「点」和「时间」两部分组成，「点」可以是「原始点」，也可以是「计算点」。

6.1.2 计算步骤

1. 根据「结果值」的「点」和「时间」，取出适用的「公式」；
2. 展开「公式」，使之只依赖「原始值」；
3. 计算出「结果值」。

6.1.3 时间处理

在整个计算中，「时间」是一个至关重要的因素，共有3种时间：结果值的时间、公式生效时间，依赖值的时间。

- 结果值的时间：要计算出来的结果的时间，为计算的参数；
- 公式生效时间：决定使用的公式，使用公式生效时间 \leq 结果值时间 的公式；
- 依赖值的时间：根据依赖值的类型，取不同的规则，下文详细描述。

6.1.4 依赖值的时间

依赖值的时间 有2种可能：

1. 等于依赖值的时间；
2. 采样时间：以「依赖值的时间」进行「采样」得出「采样值」的时间。

注：所谓「采样」，是指取「时间」小于等于「给定时间」的「最大时间」的值。

问题：目前采集时间是08:00:30，08:10:30，08:20:30，这样整点30秒，我们采样8:00:00，是否希望用08:00:30？

6.1.4.1 结果值为瞬时值

「瞬时值」只能由「瞬时值」和「持续值」计算出来：

- 瞬时值：以「结果值的时间」的「采样值」；
- 持续值：以「结果值的时间」的「采样值」。

6.1.4.2 结果值为期间值

「期间值」是计算的重头戏，使用的最多，计算规则也最复杂。

「期间值」可以由瞬时值、期间值 和 持续值 计算出来。

- 瞬时值：使用「内置函数」转化为「期间值」后，才能参与运算；
- 期间值：使用「结果值的时间」；
- 持续值：以「结果值的时间」的「采样值」。

6.1.5 自定义期间

为了支持「2019.5月份的时间为5.4~6.3号」这样的场景，在算5月份的「用电量」这样的「期间值」时，需要获取「5.4」和「6.3」这样2个时间参数。

6.2 公式表示

以下为样例：

- 某一小时(天周月季年班组)的用电量：minus(有功总电能的pointId)
- 某一班组一个月的用电量：sum(天用电量的pointId)
- 部门天用电量：minus(a线路的pointId) + minus(b线路的pointId)

6.3 支持的运算符和函数

6.3.1 运算符

- +：加
- -：减
- *：乘
- /：除
- %：取模
- m^n ：m的n次方

6.3.2 函数

支持所有Math类提供的静态函数，下面列举常用的一部分：

- abs: 取绝对值
- ceil: 向大取整
- floor: 向小取整
- cbrt: 立方根
- exp: e的参数次方
- log: 对数
- max: 最大值
- min: 最小值
- pow: 同^
- round: 四舍五入
- sqrt: 平方根

根据需要，可以扩展更多。

6.4 内置函数

在「支持的运算符」之外，提供部分「内置函数」，其中使用最多的可能是minus和sum。

6.4.1 minus尾-头

- 参数: 瞬时点
- 返回: 期间值

6.4.2 sum求和

一般来讲，求和与minus返回的结果相同，特例是班组。

- 参数: 瞬时点
- 返回: 期间值

7. 自定义期间

定义了1个接口：

- 参数: 实体ID，期间类型，时间
- 返回: begin, end

问题：

- 以什么为单位定义期间？分厂吗？模型如何定义？
- 班组的期间如何定义？
- 班组 是否为实体？如何配置班组？
- 页面如何配置出来？

问题：

所有电动机的负载率，只配一次，这种是固定的

8. Web页面

共分为4个菜单：参数管理、实体类型管理、实体管理 和 质量码管理。

8.1 参数管理

- 表格显示列：参数ID，参数名称，参数类型（瞬时、期间、持续），数据类型(布尔、数字、字符串)，单位，精度，公式，标签，操作
- 有全局搜索框、分页等
- 有导入、导出、增加 按钮
- 点参数名称列，进入参数详情
- 参数详情页，显示参数的具体信息
- 标签列，可以增加、删除标签
- 点标签，则根据该标签过滤参数
- 操作列，有修改、删除 2个功能

8.2 实体类型管理

- 表格显示列：实体类型ID，实体类型名称，参数数量
- 点实体类型名称，进入该实体类型详情页
- 表格显示该实体类型的参数

8.3 实体管理

共分为3层：实体层，实体参数层，实体参数详情层

8.3.1 实体列表

- 表格显示列：实体ID，实体名称，客户ID
- 有全局搜索框、分页等
- 点实体名称，进入实体参数

8.3.2 实体参数

- 显示该实体挂的所有参数
- 表格显示列：参数ID，参数名称，参数类型，单位等
- 点参数，进入实体参数详情页

8.3.3 实体参数详情

- 根据该参数有的期间类型，显示多标签页
- 如果参数类型为 瞬时，则只有一个标签页，如果为期间值，则显示小时、天、月、年等标签页
- 每个标签页下面，是近一段时间数据表格
- 表格列名：点ID，时间，值，操作
- 操作列：公式、数据追踪

8.4 质量码管理

- 表格显示列：质量码、质量码名称、质量码描述
- 有全局搜索框、分页等

9. 常用接口

本节介绍常用的接口，只提供索引，不详细描述，具体用法参考 源码 和 其中的注释。

9.1 参数

参数相关接口，定义在 `ParameterService` 中。

9.1.1 导入参数csvImport

用于导入参数，可重复导入

9.1.2 导出参数csvExport

用于导出参数，全量导出

9.1.3 根据ID查询参数findByld

用于根据ID查参数的详情

9.1.4 根据标签名查询参数findByTag

用于根据标签 查询参数，如 根据「电流」查出「A相电流」，「B相电流」，「C相电流」。

9.2 实体和点

实体相关的接口，定义在 `EntityService`，点相关的接口定义在 `PointService`。

9.2.1 保存实体(包括下面的点)EntityService.insertOrUpdate

大B端组好一个实体和它的点之后，发到数据系统，修改后可重复发。

9.2.2 查询实体下面的点PointService.findByEntityld

用来查找实体下已经有的点

9.3 计算公式

公式相关的接口，定义在 `ExpressionService` 中。

9.3.1 保存公式write

在大B端配好一个公式，保存到数据系统

9.3.2 查某个点 一段时间所有的公式read

略

9.3.3 查某个点适用的公式sample

略

9.4 数据的读写

数据读写相关的接口，定义在 `PointDataService` 中，包括原始数据 和 计算数据。

9.4.1 写一个点的值write

可以是数据搬运，对接，手工录入等各种。

9.4.2 批量写write

上一个接口的批量版

9.4.3 写修正值revise

用户发现一个小时值不对时，手工设一个修正值

9.4.4 写偏移量offset

换表时，用偏移量来解决读数改变的问题

9.4.5 读某个点一段时间的值read

如果是原始值，直接从库中读出来，如果是计算值，则根据公式实时计算出来

9.4.6 读某个点，某个时间的值read

如果是原始值，直接从库中读出来，如果是计算值，则根据公式实时计算出来

9.4.7 读某个点，某个时间的采样值sample

如果是原始值，直接从库中读出来，时间小于等于传入时间的值，如果是计算值，则根据公式实时计算出来

9.4.8 读某个点，一段时间的间隔采样值sample

一段时间，隔若干秒取一个值，用于页面画图

10 计算详解

计算是个复杂的话题，在 `6. 计算设计` 中宏观的描述了思路，步骤 和 时间处理，本章节详细描述实现细节。

10.1 公式规则

10.1.1 样例

- 计算某个电动机的瞬时功率，功率 = 电流 x 电压：{电流的ID} * {电压的ID}
- 计算某个部门的天用电量，在某段时间，是A，B，C3条线路之和：minus([A线路正向有功]) + minus([B线路正向有功]) + minus([C线路正向有功])
- 在上一题的基础上，计算该部门年用电量，5.1之前是2条线路，5.1之后是3条线路：sum([该部门天用电量])
- 某部门一天的电费：{该部门天用电量} * {电价}
- 某部门一年的电费，10.1之前和之后不一样：sum([该部门日电费])

10.1.2 规则概述

- 有2种特殊标记：{} 和 []
- {} 用来标记 持续量，同级的 期间量，计算瞬时量时标记瞬时量，特点是，不需要跨时间维度
- [] 用来标记 电表读数 到 电量，天电量 到 月电量这类，需要跨 类型，跨时间维度的计算
- 支持函数调用，函数名后面带()，其中放函数的参数

10.1.3 支持的函数

- minus
- sum
- max
- min

更多 内置函数，见 <https://github.com/killme2008/aviator/wiki/内置函数>，如果不够用，可以自定义函数。

10.2 瞬时值计算

在采样 或 读单个点的值 或 间隔采样时，发现是 瞬时值的计算点，则执行该规则。

1. 创建根执行结点
2. 解析公式
 1. 读出该时间点适用的公式
 2. 解析公式，得到直接依赖的点集合
 3. 如果依赖的点是原始点，加入缓存
 4. 如果依赖的点是计算点，跳到2继续递归执行
3. 读2.3中缓存的所有原始值
 1. 缓存的原始值，均为目标值时间的采样值
4. 计算
 1. 按照2中的依赖和公式，由下到上执行计算
 2. 计算过程
 1. 读取3中缓存中目标时间的采样值
 2. 调用公式执行引擎计算

10.3 期间值计算

期间值的计算最为复杂，以读一个点的为例。

1. 创建根执行结点

2. 解析公式

1. 如果是元素节点

1. 读出该时间点适用的公式
 2. 解析公式，得到直接依赖的元素节点
 3. 解析公式，得到直接依赖的数组节点
 4. 以上2条中，发现是原始值，加入缓存
 5. 如果依赖的点是计算点，跳到2继续递归执行
 2. 如果是数组节点，且其中的元素是计算值，跳到2递归执行

3. 读出计算值的自定义期间，得到开始时间和结束时间

4. 读原始值

1. 读瞬时值

1. 读出一段时间的累积量，向前后多读一个小时
 2. 读出这段时间的偏移量
 3. 根据以上2点，组装偏移量并放入缓存

2. 读期间值

1. 读出3中的时间段的期间值
 2. 加入缓存

3. 读持续值

1. 读出3中时间段的值
 2. 以开始时间采样，得到开始时间之前的第1个值
 3. 缓存 以上2点读取的值

5. 计算

1. 按照2中的依赖和公式，由下到上执行计算

2. 计算过程

1. 读瞬时值，3中时间段的偏移量，取和
 2. 读期间值，时间严格匹配，缓存中有就有，没有就没有
 3. 持续值，读缓存中的采样值
 4. 调用公式执行引擎计算