openPlant®实时数据库

产品介绍

上海麦杰科技股份有限公司

2019/09

目录

1. 公司介绍.....1
1.1. 公司发展历史.....1
1.2. 商业模式.....2
1.3. 公司荣誉.....2
1.4. 成功案例.....2
1.4.1. 华能风电集控平台.....3
1.4.2. 雪链冷库智能监控平台.....4
1.4.3. Magus IoT 助力中广核设备云运维.....5
1.4.4. 智慧电厂解决方案.....6
1.4.5. 环保三位一体监控平台.....7
1.4.6. 空气质量预报预警解决方案.....8
2. openPlant®概述.....9
2.1. openPlant®发展历程及里程碑.....9
2.2. openPlant®的主要特征.....10
2.3. openPlant®软件组成.....11
3. openPlant®技术原理.....12
3.1. openPlant®系统架构.....12
3.2. openPlant®数据结构.....13
3.2.1. 测点数据结构.....13
3.2.2. 基本数据类型.....14
3.2.3. 默认点表结构.....14
3.2.4. 数值状态说明.....16
3.3. openPlant®数据接口技术.....16
3.4. openPlant®核心 I/O 策略.....17
3.4.1. 数据管理.....17
3.4.2. 数据流分析.....18
3.4.3. 核心调度策略.....19
3.4.4. 数据检索技术.....19
3.4.5. 数据访问效率.....19
3.4.6. 数据库安全技术.....20
3.5. openPlant®数据存储技术.....20
3.6. openPlant®数据压缩原理.....21
3.7. openPlant®与关系数据库性能对比.....22
3.8. openPlant®报警服务.....22
3.8.1. 报警类型 LC 字段.....22
3.8.2. 报警状态 AS 字段.....24
3.8.3. 报警颜色.....25
3.9. 计算服务.....25
4. openPlant®可靠性设计.....26
4.1. openPlant®交叉热备功能.....26
4.2. 交叉热备原理.....27
4.3. 交叉热备配置.....28
5. 附录.....29

29
5.1. 字段详细说明: 29
5.2. 计算公式配置详解 32
5.2.1. 支持的语法符号 32
5.2.2. 计算点示例 33
5.3. 系统函数库 33
5.4. openPlant®函数 36
5.5. 水和水蒸汽函数 40
5.6. 公司业绩表 46

1. 公司介绍

上海麦杰科技股份有限公司成立于 2000 年，是中国工业互联网的先行者和行业标准的起草成员，18 年来一直专注于物联网技术的研发和应用。

目前拥有 800 多家客户，遍布电力、石油、化工、冶金、环保、钢铁、装备制造、工程建筑等行业。麦杰的物联网系统已经为超过 4000 万台设备提供链接和数据分析服务。国内超过 1000 套，海外超过 80 套，每秒钟处理着来自工业现场超过 1 亿测点的数据！

1.1. 公司发展历史

上海麦杰科技股份有限公司发展历程：

2005 年，“珠海发电厂生产数字化管理系统”通过广东省科学技术厅科技成果鉴定

2008 年，获得国家科技部的创新基金资助

2008 年，企业级实时数据库系统获得上海市科技进步奖

2009 年，获得工信部《实时数据库研发及产业化》电子产业发展基金

2012 年，被评为“上海市规划布局内重点软件企业”

2013 年，获得国家发改委“物联网技术研发及产业化专项”

2013 年，获得“上海市优秀软件产品”称号，被评为“上海市明星软件企业”

2014 年，实时数据库系统被认定为“上海市高新技术成果转化项目”

2015 年，承担环保部示范研究项目“在线监测与主要污染物减排和排污收费管理平台。应用开发标准化研究”（主管单位为内蒙古环保厅），通过环保部验收

2016 年，参与编制“火电厂工况在线监控系统安装及验收技术指南”标准编制

2018 年，麦杰工业物联网云平台发布

1.2. 商业模式

上海麦杰科技股份有限公司的商业运作是软件产品直销同时配合分销，共同推广公司产品。

1.3. 公司荣誉

上海麦杰科技股份有限公司自成立以来，共获得多份荣誉：

获得国家级荣誉超过 10 份，获得国家发明专利超过 9 个。

获国家科技部的创新基金资助，获上海市科技小巨人培育企业称号，获上海市规划布局内重点软件企业

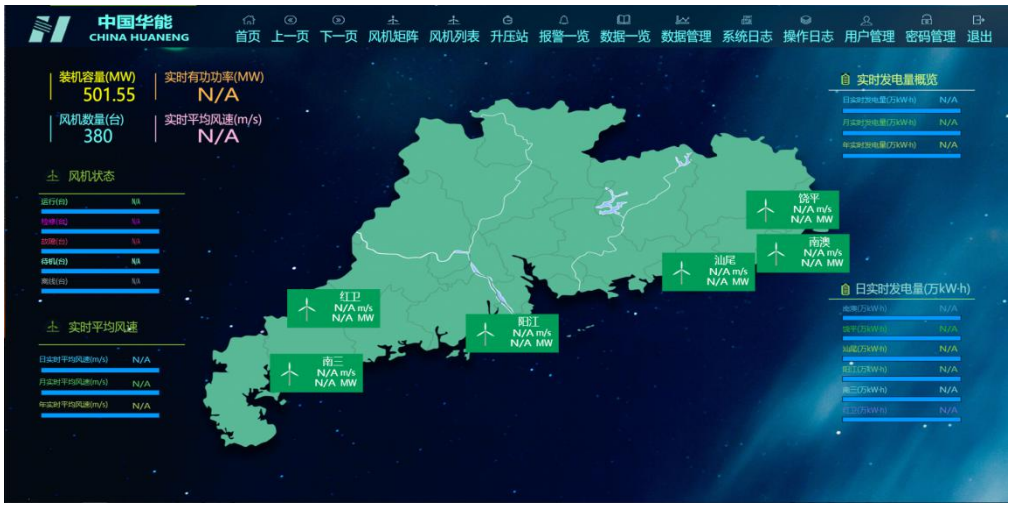
获国家发改委“物联网技术研发及产业化专项”：

1.4. 成功案例

上海麦杰科技股份有限公司 openPlant®实时数据库，已在电网、核电、火电、清洁能源、化工行业、环保领域、石油、风电以及冷链新能源等工业领域得到成功应用。公司同时成功开拓海外市场，电力领域已成功应用多套实时数据库。更多公司产品案例情况请见附录 5.6 公司业绩表。

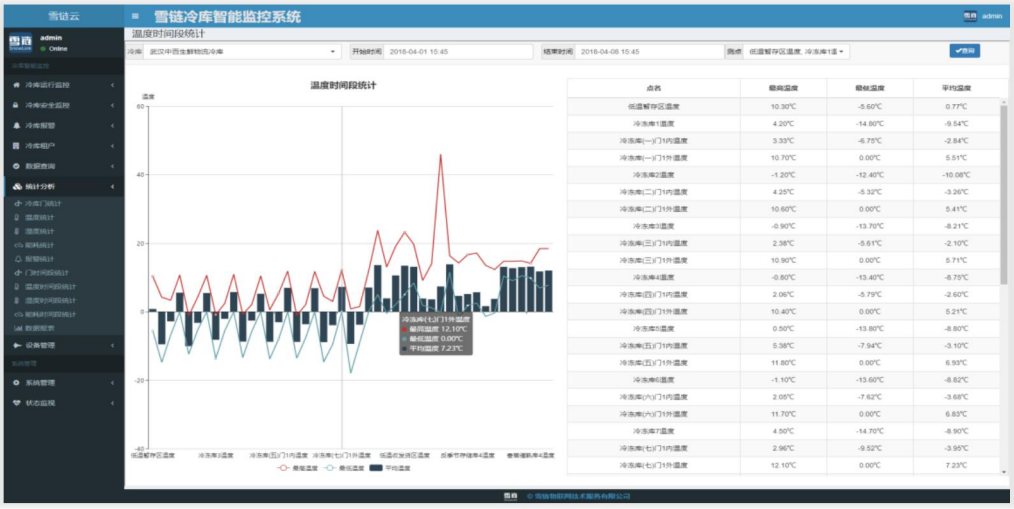
1.4.1. 华能风电集控平台

华能风电集控平台采用 Magus IoT 平台，通过平台强大的组态工具和安全集控技术，建立了 7 个区域中心（广东、吉林、河北、山西、山东、贵州、蒙西）的风电远程集控中心，实现了多风场的集中监测、远程控制和管理。



1.4.2. 雪链冷库智能监控平台

雪链冷库智能监控平台采用 Magus IoT 平台，通过 MagusBox 强大的设备联网技术，为冷库提供数字化智能管控服务，实现冷库的集中监测、集中维护、统一管理。

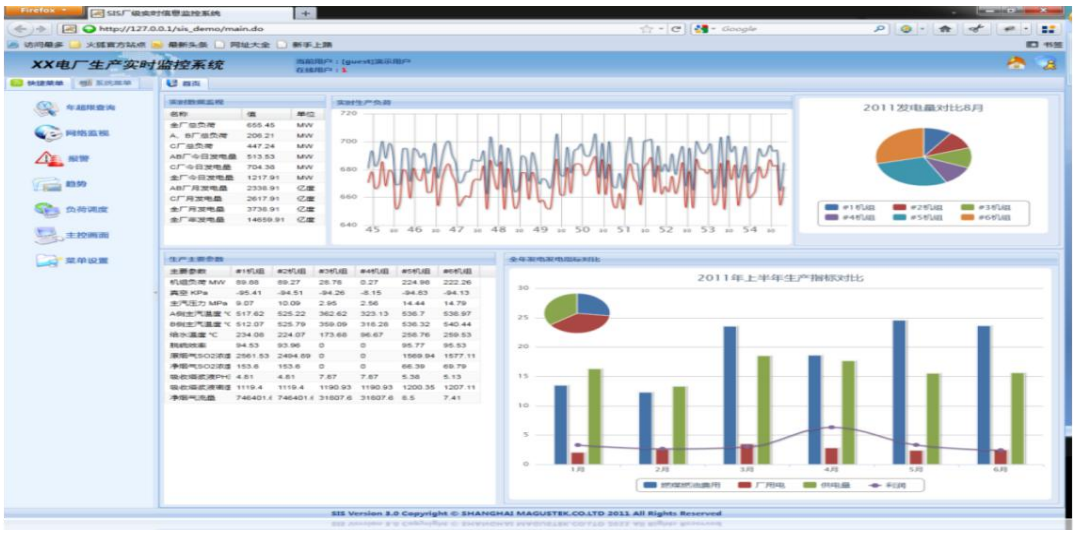


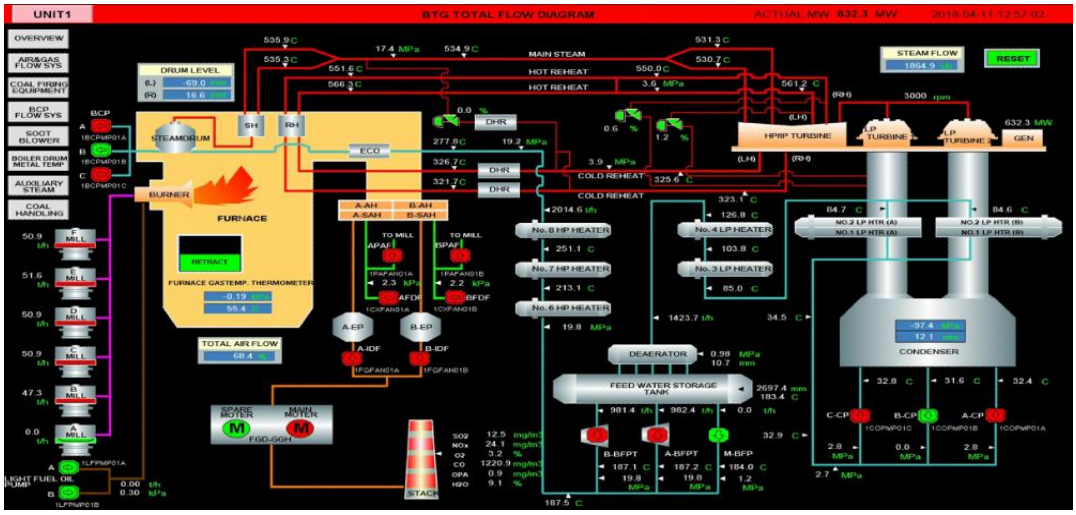
1.4.3. Magus IoT 助力中广核设备云运维

Magus IoT 助力中广核设备云运维系统，其中“DCS 的运维管理” 是中广核全产业链条的重要组成部分。运维平台整合核电 DCS 设备基础数据，贯通 ERP、SAP，以智能化的分析手段，提升运维深度，提高运维响应效率，减少控制系统安全隐患。

1.4.4. 智慧电厂解决方案

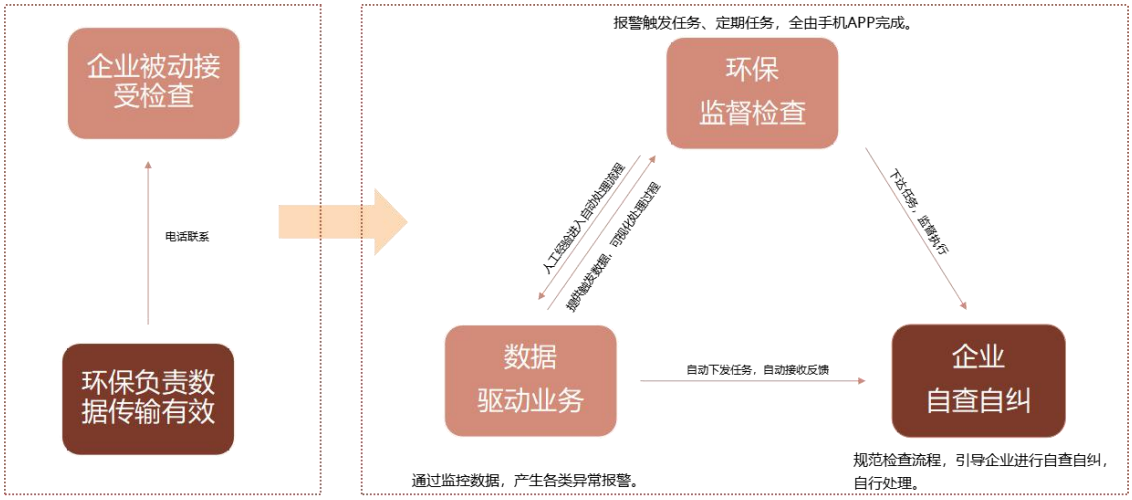
智慧电厂解决方案基于 Magus IoT 平台，利用物联网、云计算、大数据和机器学习等技术，实现电厂生产和经营管理的全方位监测和感知，并对感知数据进行融合、分析和深度挖掘，使生产过程长期处于安全、经济和环保的运行状态，让企业更智能、更高效、更具竞争力。

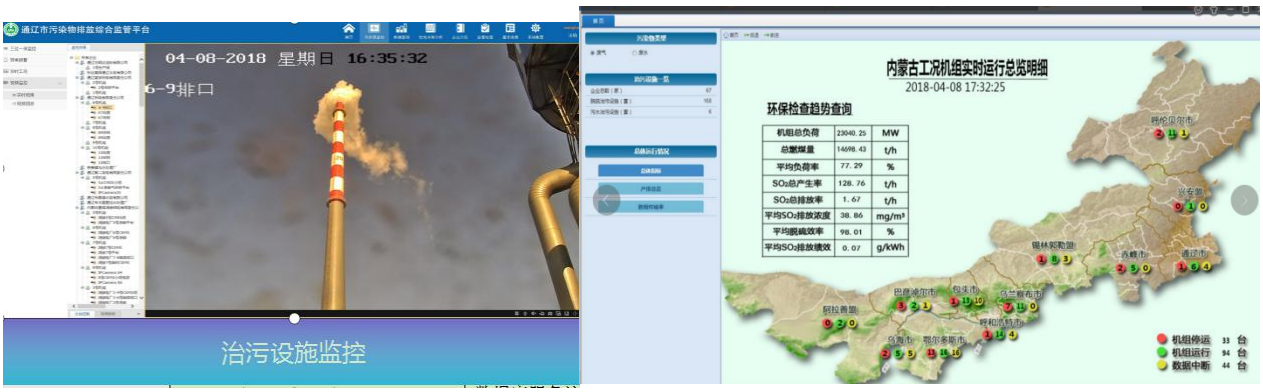




1.4.5. 环保三位一体监控平台

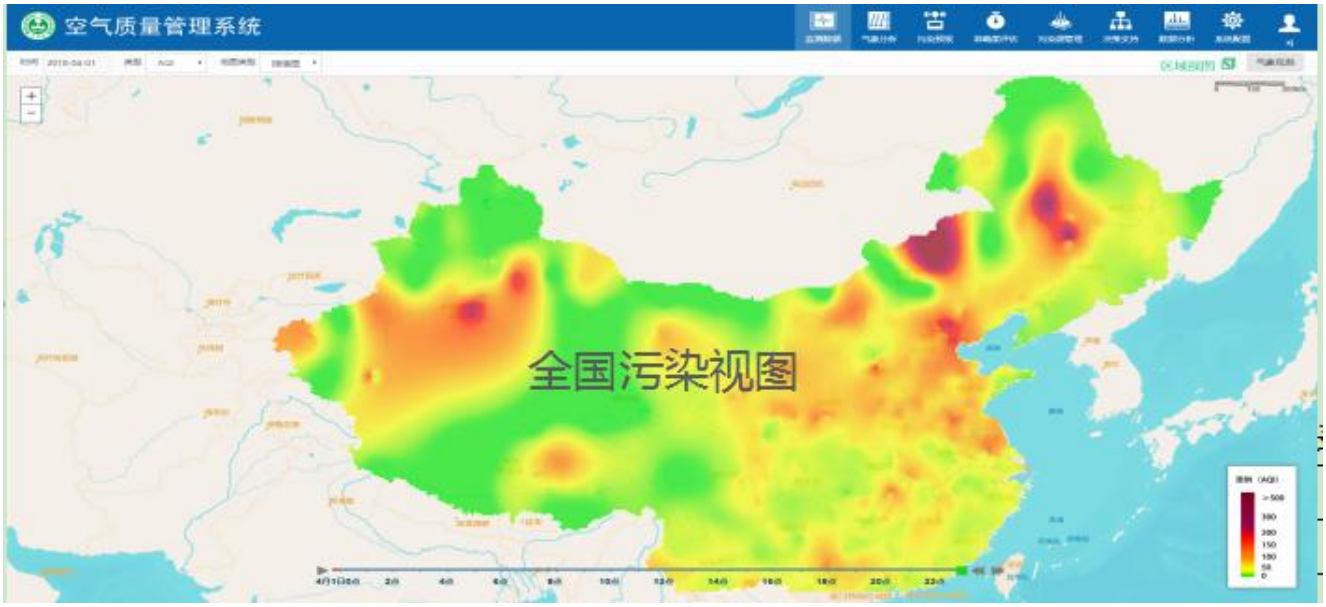
三位一体监控是集视频监控、工况监控、在线监控于一体的污染源综合监管平台，通过 Magus IoT 平台与环保管理的紧密结合，实现环境管理“系统化、科学化、法治化、精细化、信息化”的目标，提升环境监管能力。





1.4.6. 空气质量预报预警解决方案

空气质量预报预警解决方案平台，利用空气质量监测数据、气象预报数据和卫星遥感，凭借大数据分析技术和分布式计算处理能力，提供未来 3-7 天的高精度空气质量预报，实现对污染物来源和分布状况的精准预测，编制动态源清单，提供源解析服务，为改善空气质量提供决策支持。



2. openPlant®概述

openPlant®是实时数据库理论在数据中心平台中的实现。openPlant®在电力、化工、钢铁、石油、环保、交通、证券金融和物联网等领域有着非常广阔的应用前景。为企业提供高速、及时的实时数据服务，对快速变化的实时数据进行长期高效的历史存储，是前端数据层（现场总线、生产监测控制等）与管理系统之间连接的桥梁，同时也是流程模拟、先进控制、在线分析监测、过程改进优化、故障诊断分析等系统的数据平台。

openPlant®采用先进的技术和架构，可安全、稳定地实现与现场各监测仪器数据的接口，并能对采集的数据进行高效的数据压缩和长期的历史存储，同时提供方便易用的客户端应用和通用的对外数据访问接口（API/JDBC/OPC/WebAPI 等），使企业的管理和决策人员能及时、全面的了解当前的现场情况，也可方便的回顾历史，及时发现现场所存在的问题，提高决策准确度，增强企业的核心竞争力。

2.1. openPlant®发展历程及里程碑

2000 年 8 月，成功开发出实时数据库的雏形。

2000 年 10 月，成功运用于小型火电生产过程监视。

2001 年 8 月，openPlant® V1 版本发布。

2003 年 7 月，成功运用与大型火电厂级监控系统。

2005 年 7 月，openPlant® V2 版本发布。

2007 年 10 月，成功运用于核电领域。

2009 年 11 月，成功运用于环保领域。

2009 年 5 月，中标工信部”实时数据库研发及产业化”招标。

2010 年 6 月，openPlant® V3 版本发布。

2010 年 9 月，成功运用于石油生产领域。

2011 年 8 月，openPlant® 嵌入式版本发布。

2011 年 9 月，实时数据库有损/无损压缩多个专利。

2012 年 7 月，openPlant® 云平台发布。

2012 年 10 月，获得国家发改委”物联网技术研发及产业化专项”。

2013 年 5 月，openPlant® 提供国产芯片（龙芯）支持。

2013 年 11 月，openPlant® 分布式实时数据库发布。

2014 年 7 月，openPlant®安全版本发布，支持自定义映射策略等。

2015 年 7 月，openPlant® V4 版本发布。

2016 年 3 月，openPlant® 海量实时数据库发布。

2016 年 10 月，成功运用于内蒙古生态环境大数据分析领域。

2017 年 4 月，openPlant® 图形引擎全面支持各应用平台。

2017 年 7 月，openPlant® 系列应用全面支持国产操作系统平台。

2018 年 1 月，成功运用于冷链监控领域。

2018 年 9 月，麦杰工业物联网云平台发布。

2.2. openPlant®的主要特征

openPlant®采用先进的技术和架构，可安全、稳定地实现与现场数据采集、存储、分析。帮助企业实现全过程范围内的实时数据共享，为提高企业的管理水平、节约成本、增强企业的市场竞争能力提供了坚实的数据基础和科学依据。

openPlant®具有以下功能特点：

1. 采用标准的 C/C++语言编写，具有优异的跨平台性能，支持 UNIX、LINUX、WINDOWS 等各种主流操作系统；
2. openPlant®支持标准的 B/S（浏览器/服务器）和 C/S（客户/服务器）结构，采用分布式数据库技术，具有良好的开放性和扩展性；
3. 提供与国内外主流生产控制系统（DCS）、PLC 系统、SCADA 系统、各类智能仪器仪表的标准数据接口；
4. 所有采集点均可自由设定属性，如死区、压缩类型、高低限等，支持在线定义，无需重启服务进程；
5. 采用基于时间和空间维度的高效压缩算法，死区与线性压缩相结合，单点存储空间 < 2K 字节/天；
6. 数据的采样周期与各控制系统同步，记录的历史数据可保持控制系统的时间标签、数值和质量标记，数据精度与控制系统中的原始数据精度一致；

7. 快速访问历史数据。采用基于树和数据块的方法存储历史数据，用户检索几年的历史数据与检索几天前的历史数据几乎没有区别；
8. 安全认证管理。采用用户认证、功能授权、访问授权等安全措施，保护用户核心数据安全。
9. 完善的容灾策略。提供实时数据镜像、数据归档、自动故障切换等安全保护服务，保证系统无故障稳定运行。
10. 超大规模。支持千万级单实时/历史数据服务。系统采用云存储架构，可支持海量集群拓展，实现无容量瓶颈的扩容。
11. 图形仿真技术。拥有专业的图形转换工具，可将控制系统的图形转换为网络格式，用户在办公室内也能看到与控制现场一致的过程图形，实现控制室到桌面的延伸，大大提高工作效率。

2.3. openPlant®软件组成

本节主要介绍 openPlant®实时数据库安装后的缺省目录结构。

以在 Windows 下安装为例，数据库主目录为 /openplant，其下目录及文件如示：

目录	包含文件	功能描述
主目录	openPlant.exe	数据库主程序
	licrep.exe	数据库注册机器码获取工具
	Regsvc.bat,unreg.bat	数据库服务注册文件
	openPlantMonitor.exe	数据库服务监控程序
Config（配置文件目录）	config.xml	数据库主配置文件
	scheme.xml	数据库定义文件
	security.xml	数据库安全文件
	Master.db	数据库存储文件
	replicator.xml	数据库镜像配置文件
Data（历史数据目录）	database_yyyymmdd_hhmmss	历史数据文件
	alarm_yyyymmdd	数据库告警日志文件，文件按创建日期标识。
License（注册文件目录）	license.xml	数据库注册文件和安全证书
	license.cer	
	magus.cer	
Logs（日志文件目录）	openPlant.log	数据库当前日志文件
	db_yyyymmdd.log	数据库历史日志文件，文件按创建日期标识。

Web（图形趋势目录）	\diagram	图形展示文件
	\webapp	图形展示工具

3. openPlant®技术原理

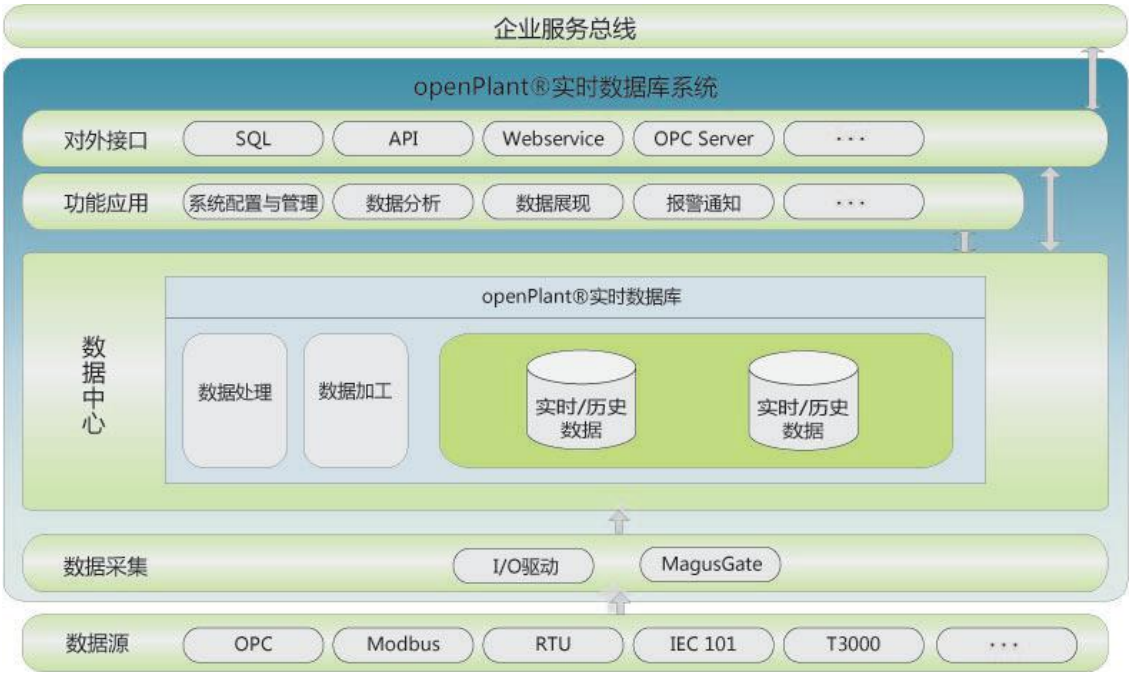
在整个 openPlant® 实时数据系统中，核心部分便是为客户提供的实时/历史数据服务。系统包含采集数据流收集、数据分析过滤、数据压缩整理、实时数据服务、历史数据归档、数据检索、权限检查、报警点配置、计算点计算等多个复杂的交叉步骤。

3.1. openPlant®系统架构

openPlant®实时数据库是用于大规模分布式过程数据的自动采集、存储和监视的企业级数据管理平台，包括 openPlant®数据接口、openPlant®实时/历史数据库核心模块、openPlant®分布式 Web 应用、openPlant®应用开发接口及工具组件；

openPlant®实时数据库采用标准 C++语言编写，对硬件平台的依赖性很小，具有良好的可移植性和优异的跨平台性能，支持主流的硬件平台包括 Unix 小型机和各类 PC 服务器，如 IBM、HP、DELL、SUN 等服务器厂商的硬件平台；

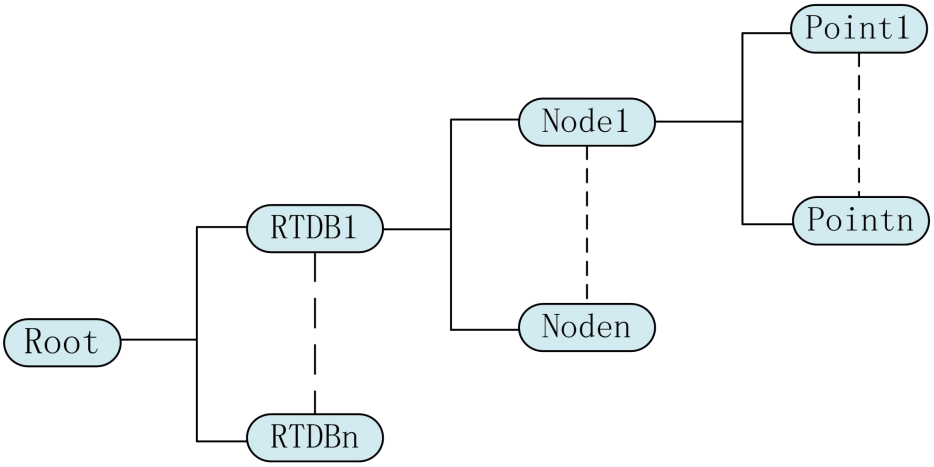
openPlant®实时数据库支持 Windows、Unix、Linux 等主流操作系统，在各平台之上都经过工程师的严格测试和现场的实际考验，可以实现上万小时的安全稳定运行。



3.2. openPlant®数据结构

3.2.1. 测点数据结构

实时数据库处理的主要对象为从现场各系统采集来的实时数据，为了管理这些数据，保证数据的唯一性，openPlant®实时数据库采用了以系统来划分的层次结构，整个系统采用树状结构，共分为 rtdb、node、point 三层，如下图所示：



其中：

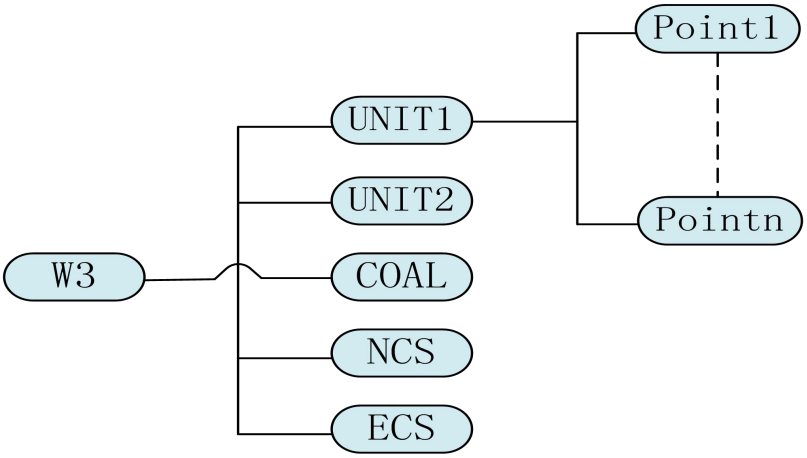
root　系统的根节点。

rtdb　可表示一个企业的数据库实例。数据库实例的缺省为 W3。

node　可以表示一个企业中的某个系统，如 DCS、ECS、NCS 等，或用户自定义的每个节点，如计算点的节点、手工输入点的节点等。

point　表示某个采集节点中的点。

例：一个只装有单一 openPlant®实时数据库的企业，该企业有 1、2 号机组 DCS，燃料、NCS、ECS 等控制系统，则该企业的数据结构如下：



如 1 号机组中有个点名为 TE0001，则该点在系统中的全局点名为 W3.UNIT1.TE0001

如 NCS 系统中有个点名为 AI0001，则该点在系统中的全局点名为 W3.NCS.AI0001

主要特点：

- 采用“[数据库名].[节点名].[点名]”的多维结构，对进入系统的所有采集点、手工输入点、计算点进行统一规划和属性定义，保留控制系统原有的点名，使采集数据在集团范围内得到统一管理并易于查询，为企业数据的集成应用提供便利。
- 内嵌数据库内核，支持标准的 SQL 语句，如 SELECT、DELETE、INSERT、UPDATE 等。通过它可以实现数据表的新建、删除、更新、查询、（多格式）导入/导出。
- 内置各种函数（如数学函数、水蒸汽焓/熵等），支持计算点在线定义和实时计算，无需编程即可获得所需的二次加工数据。
- 支持统计计算功能，可直接获取数据库点的统计值（最大值/最小值/平均值/累计值）。

3.2.2. 基本数据类型

实时数据库主要处理对象为现场数字仪表的数据和设备状态等，这些对象一般可分为数字量和模拟量两种，数字量如设备状态、开关信号、手自动信号等，模拟量如温度、流量、压力、液位参数等。

openPlant®实时数据库中主要包括两大类型的数据对象：现场采集点（一次点）和计算点（二次点）。一次点分为模拟量点（AX），数字量点（DX），短整数点（I2），长整数点（I4），双精度点（R8）。计算点一般均为 AX 数据类型，并包含计算公式定义的相关字段。

测点类型表：

测点类型	数据格式	说明	适用场合
AX	float	单精度模拟量	温度、压力、流量等
DX	boolean	逻辑开关量	开关状态
I2	short	短整型量（0-65536）	适合打包类测点,如:用于代表 16 个开关量的打包数据
I4	int	整型量（4 字节整数）	可用于存储 32 个开关量的打包数据
R8	double	双精度模拟量	如:电表数据、累计量等对数据精度要求较高的数值

3.2.3. 默认点表结构

名称	描述	类型	样式	变更影响
ID	点标识	int	静态	无法变更

ND	节点标识	int	静态	无法变更
RT	点的类型	byte	静态	无法变更
PN	点名	char	静态	无法变更
CT	修改时间	time	静态	无法变更
AN	点的别名	char	静态	无影响
PT	点的来源	byte	静态	无影响
ED	描述	char	静态	无影响
KR	特征字	char	静态	无影响
FQ	数据分辨率	short	静态	影响超时状态
CP	处理器	short	静态	无影响
HW	位置	int	静态	无影响
BP	通道号	char	静态	无影响
LC	报警类型	byte	静态	影响报警
AP	报警优先级	byte	静态	影响报警
AR	存档	byte	静态	影响历史存储
FL	标志位	short	静态	影响实时处理
ST	值为 1 时的描述	char	静态	无影响
RS	值为 0 时的描述	char	静态	无影响
EU	单位	char	静态	影响显示
FM	显示格式	short	静态	影响客户端显示
IV	初始值	float	静态	影响实时数据
TV	量程上限	float	静态	影响显示
BV	量程下限	float	静态	影响显示
LL	报警低限	float	静态	影响报警
HL	报警高限	float	静态	影响报警
ZL	报警低低限	float	静态	影响报警
ZH	报警高高限	float	静态	影响报警
DB	死区	float	静态	影响数据存储
DT	死区类型	byte	静态	影响数据存储
KZ	压缩类型	byte	静态	影响数据存储
TT	统计类型	byte	静态	保留字段
TP	统计时间周期	short	静态	保留字段
TO	统计时间偏移	short	静态	保留字段
KT	计算类型	byte	静态	影响计算点
EX	计算表达式	varchar	静态	影响计算点
TM	时间	time	动态	
AS	状态	short	动态	
AV	数值	float/byte/short/int/double	动态	

更多字段详细说明请见附录 5.1《字段详细说明》。

3.2.4. 数值状态说明

实时数据库中所有实时、历史数据均包含动态数据时间、状态、值信息。其中 AS 信息中包含的数据质量信息表示为：

16 位状态字说明：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit 0　表示开关量 DX 的值
bit 1　报警位
bit 2　报警位
bit 3　报警位
bit 4　报警位
bit 5　报警未确认标记,测点产生过报警后，消失的未人工确认报警。0 表示无未确认报警，为 1 表示当前报警未确认
bit 6　报警抑制
bit 7　是否报警

bit 8　设备强制状态
bit 9　点的质量　0 表示质量好，1 表示质量坏
bit 10 + bit 11

bit 10	bit 11	描述
1	1	控制指令超时
0	1	控制下发
1	0	控制故障

bit 12　表示有控制指令
bit 13　设备挂牌
bit 14　是否初始值，为 1 时表示当前数据为数据库启动时第一个数据。
bit 15　是否超时，为 1 时表示在指定周期内实时数据没有更新。
其它位系统保留。

3.3. openPlant®数据接口技术

丰富、高效、稳定的 I/O 驱动是实时数据库实现自身价值的一个重要保障。openPlant®实时数据库系统，通过适配现场的各种接口，并经过核心处理，完成数据的采集、实时计算、报警计算以及其它应用，实时数据被不断存入磁盘历史存储，形成可追溯的历史信息，同时通过向应用层提供各种适配接口，支持各种开发语言和各种应用需求的访问。

✧　各种控制系统接口协议支持，openPlant® 实时数据库系统拥有超过 450 个的 I/O 驱动：

- WDPF、Ovation、Foxboro、INFI-90、Symphony、Teleperm XP、XDPS、EDPF、MAX1000、日立、三菱、东芝、浙大中控、和利时、南瑞、四方、Intouch、RSView、iFix、Wincc、MCGS 、KingView 、FameView 等；
- SCADA 、PLC、RTU、智能仪表等；
- 通用协议 OPC/DDE/ModBus/ProfiBus；

✧ 协议载体

- TCP/UDP
- HTTP 等

✧ 自定义数据采样频率，默认是 1 秒；

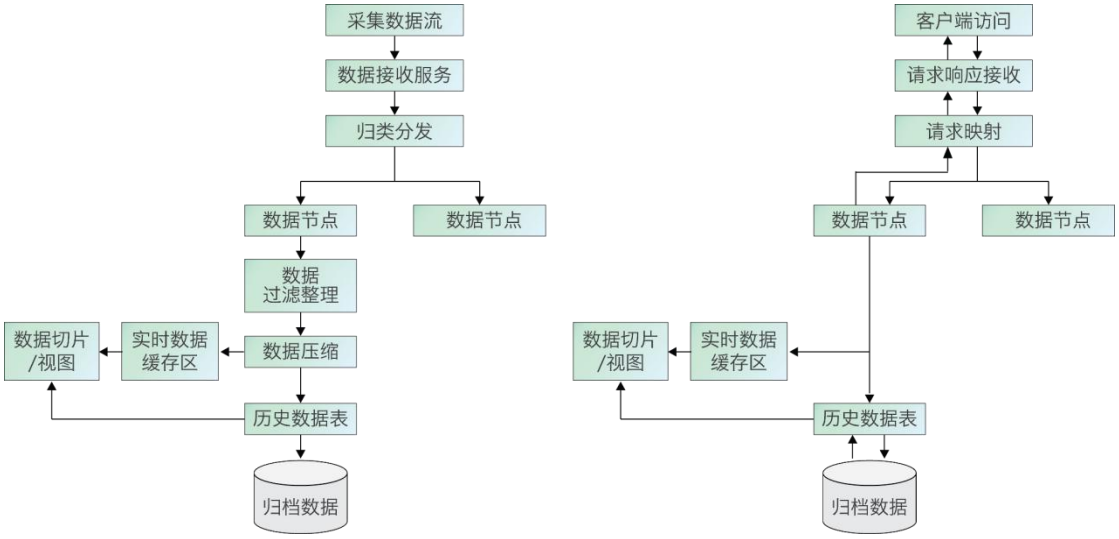
✧ 在线自动更新采样点配置，当实时数据库的配置发生改变，采集程序能得到同步更新；

✧ 监控数据链路状态，数据链路中断，能第一时间获得报警信息；

✧ 完善的故障恢复机制，接口支持断点续传,本地缓存的时间长度可配置；

3.4. openPlant®核心 I/O 策略

openPlant®实时数据库中数据流向如下所示，其中核心策略和处理机制体现在数据流向中：

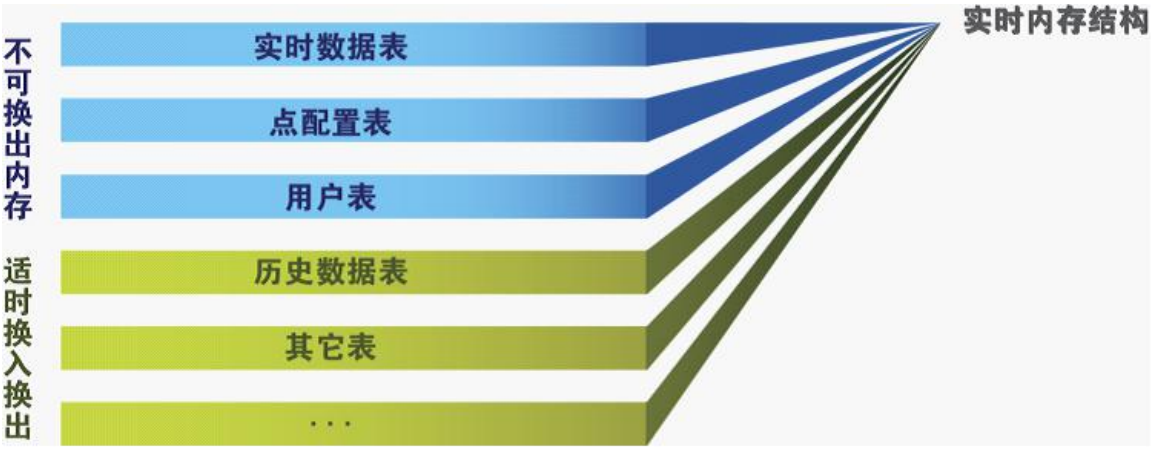


3.4.1. 数据管理

openPlant® 实时数据库数据结构中可分为：不可换出内存与适时换入换出两部分：

不可换出内存：在实时数据库运行过程中，其数据驻留在内存中，不进行释放内容。该类数据包括实时数据表、点配置表、用户表（含用户权限管控等相关数据）。

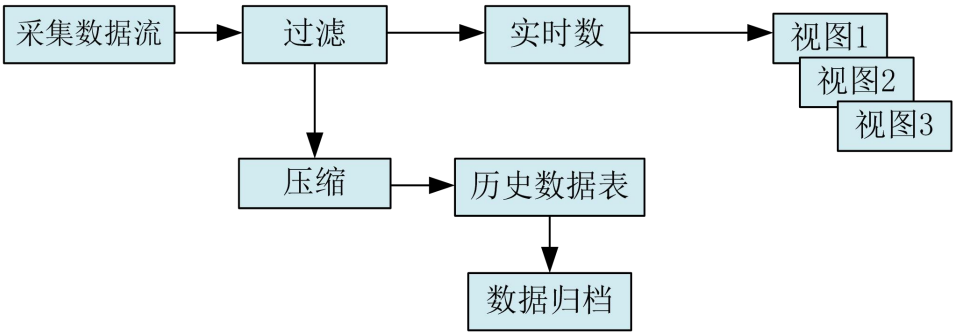
适时换入换出：在实时数据库运行过程中，系统会经常由内存与硬盘之间进行数据交换部分。包括系统历史数据表、系统日志信息等。该类数据会产生部分磁盘 I/O 读写操作。



3.4.2. 数据流分析

数据处理流程大致包含：数据流分析、数据过滤、实时数据更新、视图展示、数据压缩、历史数据表、数据归档等部分。

- 数据流分析：是指对现场设备数据、二次加工运算数据、其它软件程序等发送到 openPlant®要求写入的数据包进行数据分解。其中包含数据解包、数据解密、规范性检查等步序。
- 数据过滤：经由数据解析后产生的数据写入请求，将进行错误数据丢弃、合理数据过滤检查，从而完成数据请求分类，识别出实时数据与历史数据分支。
- 实时数据处理：对实时数据进行分析检查，完成实时数据视图更新。
- 视图展示：提供实时数据展示视图。
- 数据压缩：当过滤反馈为历史数据写入请求时，实时数据库将进行数据包压缩处理动作，并更新历史数据表。
- 历史数据表：提供历史数据视图。
- 数据归档：将更新的历史数据进行归档，写入磁盘操作。



3.4.3. 核心调度策略

实时数据库中存在线程池，对于需要处理读、写、采集等任务,实时数据库根据核心调度策略执行任务。

首先，判断哪些任务的优先级更高。实时数据库内部往往通过判断测点的更新周期来间接揣测任务的优先级。虽然往往可以让多个线程自己竞争，但如果某个测点的更新周期位 1 秒，而另一个的更新周期为 10 秒，那么，可想而知，应用对 1 秒更新的实时数据的实时性要求高于 10 秒的。因此，如果有 1 秒的读任务没有完成，则不执行 10 秒的，对于 CPU 数量小于等待线程数量的时候，特别适用。

另外，读实时值的任务优先级应该高于读历史值的任务。

读队列，任务可以被线程顺序执行，而如果低优先级的线程得以执行的时候，会检查一下是否还有更高优先级的队列中需要执行，如果有，则让出时间片。

对于写任务，往往可以和读任务并行，而写更重要，排在更高的优先级。

采集的优先级和读的优先级相比，采集往往是一个与读优先级的最高优先级相当的任务。

3.4.4. 数据检索技术

openPlant®实时数据库中，用来定位一个测点，是通过 ID 和名称。

数据库中 ID 和全点名都是唯一的，并且实现中 ID/名称是静态的，一旦指定或者自动分配，则不可变更。对于名称和 ID，采用哈希表的方式建立索引。

检索过程中，支持通过名称/ID/时间戳方式进行数据检索，支持条件聚合完成数据检索。时间戳支持绝对时间范围、指定时刻等基本的时序查询。在实时数据库中存储时，先将数据按时间分块存储，时间戳索引此时用于定位一个块。执行检索的时候，时间戳索引的查找方式是二分法查找存储块。

同时，通过测点数据结构可知，数据库对测点建立分组。无论是点表方式，还是查询历史过程建立的历史分组，都将用户访问的数据缩减为一个子集。查询分组索引的效率大大高于全局查找。

同时实时数据库完成 SQL 引擎优化，全面提升 SQL 查询及执行性能，引擎支持各类 SQL 语句和表达式。名称/ID/时间戳配合其他字段完成条件聚合，方便客户进行深度数据探索分析，更加完善了实时数据库的检索机制。

3.4.5. 数据访问效率

openPlant® 实时数据库对实时数据的处理完全在主内存中进行，减少了磁盘访问、文件 I/O 和缓同步等开销，使得对数据的访问速度提高到了极致。

- 支持超过 1000 万实时事件接入响应。

- 磁盘写入历史速度超过 100 万事件每秒。
- 检索历史速度超过 200 万事件每秒。
- 实时数据与控制系统同步，数据刷新<1 秒。
- 实时掌握现场情况，提高管理的实时性和准确性。
- 100 万实时数据访问耗时 < 400ms。
- 支持超过 2000 用户并发访问。

3.4.6. 数据库安全技术

openPlant® 数据库支持用户认证，并提供 security.xml 进行用户以及权限管理。

其中安全管理模块用于从控制台界面管理数据库的用户和组。

用户权限表：

操作描述	需要系统权限	或组权限
查阅组列表	SELECT	不可
创建组	INSERT,CREATE	不可
删除组	DELETE,DROP	不可
为测点设定组	UPDATE	UPDATE
查阅用户列表	SELECT	不可
创建用户	INSERT,CREATE	不可
删除用户	DELETE,DROP	不可
修改重置其他用户密码	UPDATE,ALTER	不可
为用户授权	UPDATE,GRANT	不可
查阅测点静态数据	SELECT	SELECT
创建测点	INSERT	INSERT
编辑测点静态数据	UPDATE	UPDATE
删除测点	DELETE	DELETE
读取测点实时历史数据	READ	READ
写入测点实时历史数据	WRITE	WRITE

3.5. openPlant®数据存储技术

实时数据库支持元数据管理，历史数据存储采取分片技术。

在实时数据库中，时序数据是最基本和主要的信息。鉴于每条时序数据都具有特定的时间戳，数据库执行数据存储的时候，便按时间戳进行分块存储，将时间戳作为索引来检索定位。时序数据需要执行存储时，

只要具有相同的时间戳，系统都会将其存储到同一个数据块中去。

openPlant® 的历史数据文件都存放在 data 目录下，数据文件根据生成日期命名。其中数据文件大小阈值不是固定的，数据库系统支持自定义配置。如果当日单个历史数据文件大小超出客户定义的阈值，则系统自动生成新数据块文件存储。这样做就可以满足当有新数据存入的时候，可以快速达到块数据的均衡。而且每个数据块的数据量与区间的大小也是没有关系的，同样大小的数据块有可能其中一个数据非常集中。

实时数据库历史存储参数，支持自定义：

pageSize	历史在线存储大小	4096
archivePeriod	历史归档周期，单位:秒	86400
archiveSize	历史归档大小，单位：MB	1500

实时数据映射存在 openplant/config/ snapshot.dat。在实时数据库运行过程中，其数据驻留在内存中不进行释放内容。该类数据包括实时数据表、点配置表、用户表（含用户权限管控等相关数据）。

存储技术采用平衡二叉树方式。

平衡二叉树（Balanced Binary Tree）又被称为 AVL 树（有别于 AVL 算法），且具有以下性质：它是一 棵空树或它的左右两个子树的高度差的绝对值不超过 1，并且左右两个子树都是一棵平衡二叉树。最小二叉平衡树的节点的公式如下 F(n)=F(n-1)+F(n-2)+1 这个类似于一个递归的数列，可以参考 Fibonacci 数列，1 是根节点，F(n-1)是左子树的节点数量，F(n-2)是右子树的节点数量。

3.6. openPlant®数据压缩原理

在实时数据库系统中，数据压缩一般分为有损压缩和无损压缩两种。

无损的一般通过各类近似霍夫曼编码的方法压缩数据，有损则是采用线性拟合的方法，数据压缩在传统意义上是为了减少磁盘空间。

openPlant® 实时数据库根据实时数据的特点，采用了基于时间和空间的二维压缩算法，对大量的实时数据进行原型压缩，同时结合高效的数据检索策略设计了实时数据库的压缩系统。这样，既可以较完整地保存大量的原始数据，又能有效地使用数据，确保这些数据能够快速地被访问。

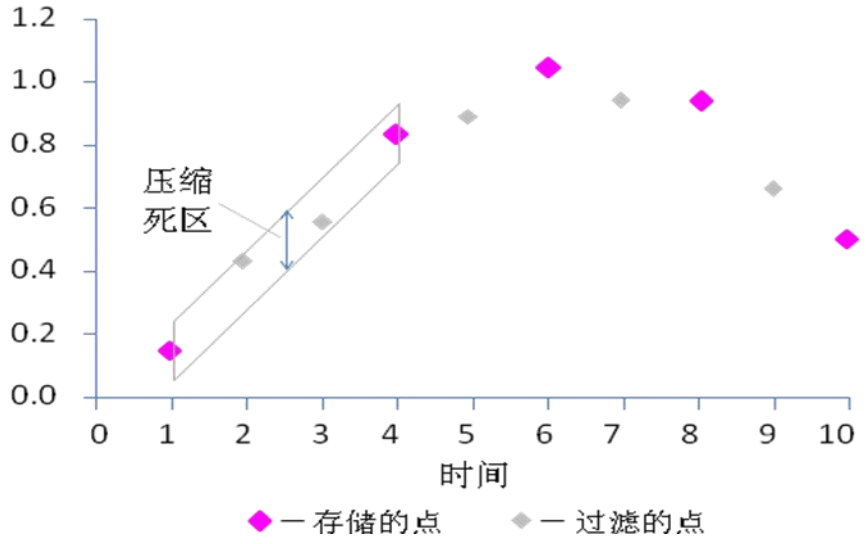
时间纬度的压缩：

- 定制的采样频率（1S/5S/10S/60S）；
- 例外报告：对不发生变化的点，或值的变化在定义的死区范围之内，或时间在最大时距之内，不存入历史表。
- 矢量线性压缩，保存关键点。

空间纬度的压缩：

- 以数据块为基本单位，结合平衡二叉树；

➢ 减少索引数据所占有的磁盘空间。



3.7. openPlant®与关系数据库性能对比

对比项目	关系数据库	openplant
每秒读写速度	5000	>1000000
实时数据查询	-	<1 秒
历史数据检索(5000 万条记录查询)	>1 小时	<1 分钟
磁盘存储占用	1	1/90
30 万个测点的每秒变化一次的历史数据数据一年存储所需	>6000GB	<500GB
支撑环境	多台高端服务器集群、磁盘阵列等	普通服务器
常规应用场景	业务及管理类数据记录,多应用于管理类系统	海量实时数据采集、存储，多用于工业流程监测、分析

3.8. openPlant®报警服务

openPlant® 实时数据库中提供数据报警服务。报警服务分为开关量及模拟量报警两类。其中 DX 类型点仅需要设置 LC 字段即可，AX/I2/I4/R8 类型根据需要设置 LL (报警低限)、ZL(报警低 2 限)、L3(报警低 3 限)、L4 (报警低 4 限)、HL(报警高限)、ZL(报警高 2 限)、H3(报警高 3 限)、H4(报警高 4 限)。报警高限和报警低限间的是正常范围。

AX/I2/I4/R8 报警共 8 个限值，且支持每个告警颜色单独设置，即报警支持 8 限 8 色。

3.8.1. 报警类型 LC 字段

LC 设定方式：若需要设定对应报警运算，通过将所选项目的代码 CODE 相加后设定即可。

模拟量(AX/I2/I4/R8)字段说明，报警可任意组合。

LC 字段		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
低限报警	LL								1
高限报警	HL							1	
低 2 限报警	ZL						1		
高 2 限报警	ZH					1			
低 3 限报警	L3				1				
高 3 限报警	H3			1					
低 4 限报警	L4		1						
高 4 限报警	H4	1							
识别码		128	64	32	16	8	4	2	1

例如：某需求中要对低限报警、高限报警、高 2 报警三项设定报警需求。

LC 设置计算公式为：1(低限报警) +2(高限报警)+ 8 (高 2 限报警) = 11(LC 设定值)

对于 AX/I2/I4/R8 点，根据 LC 配置的值，服务器可能产生如下的状态改变事件:

H4 报警，当 AV 值大于 H4 时报警，依赖于高限预警，如果高限不设置预警，就不会报警

L4 报警，当 AV 值小于 L4 时报警，依赖于低限预警，如果低限不设置预警，就不会报警

H3 报警，当 AV 值大于 H3 时报警，依赖于高限预警，如果高限不设置预警，就不会报警

L3 报警，当 AV 值小于 L3 时报警，依赖于低限预警，如果低限不设置预警，就不会报警

ZH 报警，当 AV 值大于 ZH 时报警，依赖于高限预警，如果高限不设置预警，就不会报警

ZL 报警，当 AV 值小于 ZL 时报警，依赖于低限预警，如果低限不设置预警，就不会报警

HL 预警，当 AV 值进入 HL-ZH 之间时发生预警

LL 预警，当 AV 值进入 LL-ZL 之间时发生预警

正常事件，当 AV 从非正常态进入正常态是也会发生状态改变事件。

开关量(DX):

名称	识别码
变为 0 报警	1
变为 1 报警	2
变化报警	3

DX 开关量点的报警，隔离事件根据 LC 字段配置的事件发生真值表：

LC 整型值	LC 二进制	报警事件

	第一位	第零位	
3	1	1	变化报警
2	1	0	变 1 报警
1	0	1	变 0 报警
0	0	0	不发生报警事件

3.8.2. 报警状态 AS 字段

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

第 7 位为 1 表示处于非正常状态，否则为 0 表示处于正常态，无论 AX 或者 DX 点。

对于 AX/I2/I4/R8 点，在第 7 位为 1 的情况下，第 2 位代表超低限，第 3 位代表超高限，第 4 位代表超 2 限(高 2，低 2)。第 4 位要与 2，3 位配合使用。第 5 位代表告警是否确认。

AX/I2/I4/R8 测点根据 AV 值的变化，AS 状态对应如下:

指标名称	字段	AS bit7	AS bit5	AS bit4	AS bit3	AS bit2	AS bit1
		是否告警	是否确认				
高 4 限报警	H4	1	0/1	1	1	0	1
高 3 限报警	H3	1	0/1	0	1	0	1
高 2 限报警	ZH	1	0/1	1	1	0	0
高限报警	HL	1	0/1	0	1	0	0
正常		0	1	上次报警级别			
		0	0	0	0	0	0
低限报警	LL	1	0/1	0	0	1	0
低 2 限报警	ZL	1	0/1	1	0	1	0
低 3 限报警	L3	1	0/1	0	0	1	1
低 4 限报警	L4	1	0/1	1	0	1	1

告警上报和告警确认的对应关系如下：

报警状态	报警(7)	报警未确认(5)	报警级别(4,3,2,1)
正常无报警	0	0	无
新产生报警	1	1	当前报警级别
已确认报警	1	0	当前报警级别
恢复的报警	0	1	上次报警级别

DX 点也可通过第七位与 AX 点等同的方式判断是否报警。如果第七位为 1 则认为此点进入了隔离状态和报警状态。另外如果此点的 LC 字段被配置为 3，即变 0 变 1 都报警，那么此点的 AS 的二进制末尾两位一定都为 1，即 TOGGLE 状态。

注意：节点中，LC=1 表示启用该节点的报警处理。若未设置，默认不进行判定。

3.8.3. 报警颜色

AX/I2/I4/R8 类型测点对应的 8 门限报警默认颜色如下：

告警名称	默认颜色
报警 LL 限颜色	0xFF0000
报警 ZL 限颜色	0xCC0000
报警 L3 限颜色	0x990000
报警 L4 限颜色	0x660000
报警 HL 限颜色	0xFF0000
报警 ZH 限颜色	0xCC0000
报警 H3 限颜色	0x990000
报警 H4 限颜色	0x660000

3.9. 计算服务

openPlant®实时数据库中，提供了标准的数据二次计算引擎。可方便用户不进行任何二次开发即可实现简单数据二次加工运算，获得实时结果。

计算服务中，仅识别在 PT=CALC 类型的测点。若对应测点中 EX 字段未设定任何公式，系统将不对该点进行任何计算处理。

实时数据库支持水和水蒸气函数、openPlant® 函数和行业函数扩展包，同时支持 Lua 语言，计算公式中支持各类 Lua 语句和 Lua 标准库。

启动计算条件：

- ✧ EX 字段已设定，并设定内容满足语法规范。
- ✧ 表达式中所包含的测点中存在某测点数据发生变化时。
- ✧ 计算类型 KT 可配置为变化计算或每秒计算。

另外：

- ✧ EX 字段默认长度 248。 若计算公式太长导致字段丢失，需要修改 schem.xml 文件将 EX 字段长度修

改为 1023、或 1024。

✧ 计算点状态显示规则，表达式中所包含的测点中:

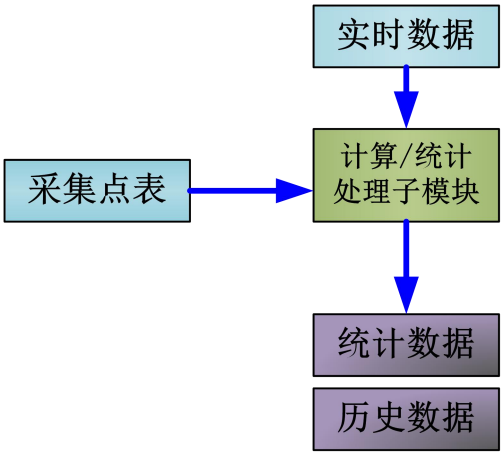
◆ 全部测点状态为 GOOD 时，该计算点状态为 GOOD。

◆ 某测点状态为 Timeout 时，该计算点为 Bad。

◆ 全部测点状态为 Timeout 时，该计算点为 Timeout。

数据计算引擎支持

实时数据库运行过程中，针对计算测点，系统会根据计算公式加载实时数据，再根据编写的各种算法公式，执行实时计算并获得结果，结果呈现为该计算测点的实时值。



1. **支持的语法符号:**

   算术符号: -, +, *, /, %, ^

   逻辑符号: &, |, ~, !, &&, ||

   比较符号: ==, >, <, >=, <=, !=

   嵌套符号: (,), {, }

   关键字: and, or, not, if, else

2. **水和水蒸气函数**
3. **行业函数扩展包**
4. **openPlant®函数**
5. **行业函数扩展包**
6. **数学与三角函数**
7. **Lua 逻辑语句和标准函数库。**

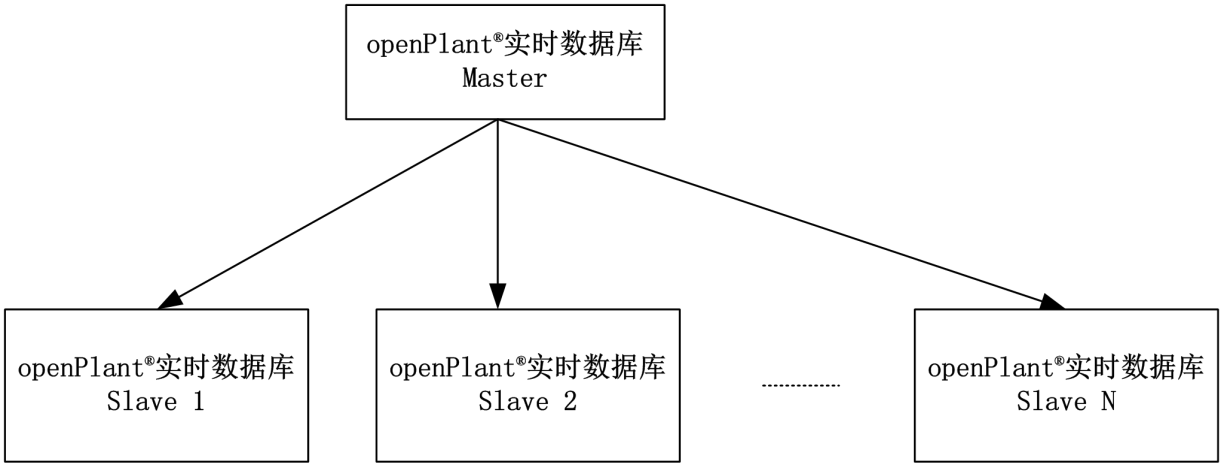
   详细计算公式和函数介绍请见附录 5.2、5.3、5.4 和 5.5。

4. openPlant®可靠性设计

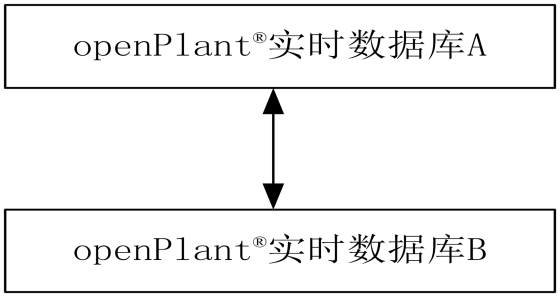
本节介绍 openPlant®实时数据库的可靠性功能，可满足客户多种数据备份需求。

4.1. openPlant®交叉热备功能

openPlant®实时数据库交叉备份功能，即支持不同数据库间实时数据、历史数据和配置信息等备份，数据库备份支持 1:N,N:1。



交叉互备功能另外重要的一点，实时数据库支持互相备份。



4.2. 交叉热备原理

1)实时数据库 A 和实时数据库 B 安装完成后，根据备份需要，配置待备份信息。

2)双方数据库启动后，系统根据配置的备份信息，决定 Master 数据库和 Slave 数据库。

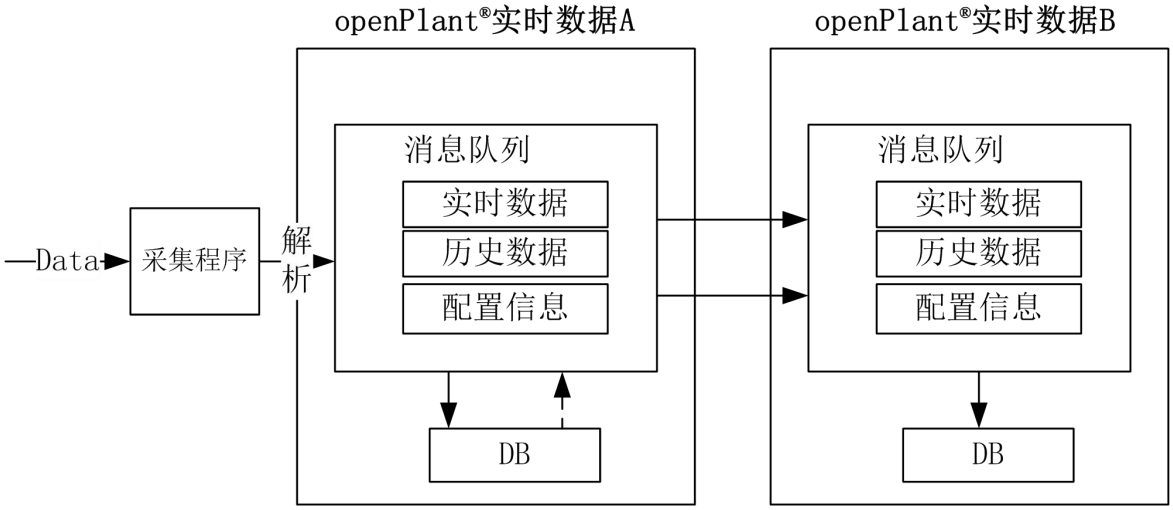
备份功能启动。

3)随后，采集程序获取到数据源，经过解析后上传至 Master 数据库。该数据类型包括实时数据、历史数据和配置信息。信息被保存在 Master 数据库的消息队列中。因为备份功能正常，该部分分信息将同步至 Slave 数据库。

4)如果通讯中断，备份无法执行。则备份失败的数据会被记录在 DB 中，同时给备份失败数据打时间标签。该通讯中断过程中，数据库会每 60s 自动检测连接是否恢复。待通讯恢复后，DB 数据会被更新至对方服务器。

5)需要注意的是，当数据库互相备份时，如果通讯中断，则数据库 A 和数据库 B 分别存在 DB 数据。此时将比较数据的时间标签，将较新时间标签的数据同步到对方数据库。

数据库交叉互备的工作原理图：



4.3. 交叉热备配置

openPlant®实时数据库备份功能，交叉热备配置文件为 replicator.xml，该文件放置在 openPlant\config 目录下。

文件内配置参数包括：

参数	意义
IP=""	Slave 数据库 IP，例 IP="192.168.4.115"。
PO=""	Slave 数据库端口 port，例 PO="8200" 。
US=""	Slave 数据库用户名， US="sis"。
PW=""	Slave 数据库密码，PW="openplant"。
PN=""	Master 数据库源对象，W3 或者点表或者测点。
TN=""	Slave 数据库备份对象，W3 或者点表或者测点。

数据库支持的备份映射关系如下：

映射关系	功能解析

数据库->数据库	W3-->W3	Master 数据库整体备份到 Slave 数据库
节点->数据库	W3.NODE-->W3	Master 数据库点表 NODE 和点表下所有测点信息，将备份到 Slave 数据库。
节点->节点	W3.NODE-->W3.NODE	Master 数据库点表 NODE 和点表下所有测点信息，将备份到 Slave 数据库 NODE 点表
	W3.NODE-->W3.UNIT	Master 数据库点表 NODE 和点表下所有测点信息，将备份到 Slave 数据库 UNIT 点表
点->数据库	W3.NODE.POINT-->W3	Master 数据库测点 W3.NODE.POINT 备份至 Slave 数据库
点->节点	W3.NODE.POINT-W3.NODE	Master 数据库测点 W3.NODE.POINT 备份至 Slave 数据库点表 NODE 下对应测点
		如果测点不存在，则备份开始时 Slave 数据库会先建测点 POINT
点->点	W3.NODE.POINT-->W3.NODE.POINT	Master 数据库测点 W3.NODE.POINT 备份至 Slave 数据库点表 NODE 下测点 POINT
	W3.NODE.POINT--> W3.NODE.TEST	Master 数据库测点 W3.NODE.POINT 备份至 Slave 数据库点表 NODE 下测点 TEST。
	W3.NODE.POINT--> W3.UNIT.TEST	Master 数据库测点 W3.NODE.POINT 备份至 Slave 数据库点表 UNIT 下测点 TEST。

5. 附录

5.1. 字段详细说明:

动态字段介绍:

TM 时间

动态字段，测点的时间，指测点数据的更新时间。

AS 状态

动态字段，测点的状态。

AV 数值

动态字段，测点的数值。不同类型测点的 AV 字段

测点类型	描述	AV 字段类型
AX	模拟量点	float
DX	开关量点	byte
I2	短整数点	short
I4	长整数点	int
R8	双精度点	double

静态字段介绍

ID 系统标识号

系统标识号 ID 在创建对象时由用户指定或者系统自动分配，设定后不允许更改，如果需要更改必须删除对象后再重建。

系统标识号全局唯一，是用于操作对象的静态属性、实时和历史数据的关键索引。

ND 节点标识号

ND 是测点所属父节点的系统标识号。

PT 点的来源

点的来源有 2 种：采集点(DAS)和计算点(CALC)。

采集点是由外部程序负责采集或计算，然后更新实时数据的测点。

计算点是由内部数据库计算模块根据公式自动计算实时数据的测点。

测点类型中仅允许设定为采集点或计算点两种类别,对应类型值为：采集点=DAS、计算点=CALC。

RT 点的类型

点的类型分为模拟量点（AX），数字量点（DX），短整数点（I2），长整数点（I4），双精度点（R8）。

点的类型设定后不允许更改。

PN 点的名称

测点在数据库中名称，同一节点下必须保持唯一。

点名和点系统标识号都是操作测点数据的索引。

点名合法字符包括：字母、数字、下划线_、冒号：

AN 点的别名

点的别名是测点在外部采集系统中的名称，如果测点在外部系统的名称中包含各种特殊符号如：+ - * / \ ^ % ! >< = ( ) [ ] { } | # @ ` ” 、, ; ? ~ $ & 在将测点导入到 openPlant®时需要将特殊符号转换为_，但是在从外部系统采集数据时一般是根据测点在外部系统中的名称，设立别名的目的主要为方便采集程序使用。

ED 描述

测点的描述。

KR 特征字

测点分类或分组名称。

FQ 数据分辨率

测点分辨率，以秒为单位，缺省为 1。

例如：某性能计算的测点每分钟才产生一个数据，就可以设置分辨率为 60 秒。

CP 处理器（之前的版本可能为 SR 控制器）

测点在采集系统中的控制器号。

HW 位置

测点在采集系统中的硬件位置。

BP 通道号

测点在采集系统中的通道号。

LC 报警类型

测点在采集系统中的报警类型。

AP 报警优先级

测点在采集系统中的报警优先级。

AR 归档

测点实时数据是否进行历史存储。

FL 标志位

系统保留。

CT 修改时间

测点静态属性的最后修改时间。

EU 单位

模拟量测点的工程单位。

FM 显示格式

模拟量测点的小数点显示位数,缺省值为2。

IV 初始值

测点的初始值。

TV 量程上限

模拟量测点的量程上限。

BV 量程下限

模拟量测点的量程下限。

LL 报警低限

模拟量测点的报警低限，当测点实时值小于它，测点状态设置为低限报警。

HL 报警高限

模拟量测点的报警高限，当测点实时值大于它，测点状态设置为高限报警。

ZL 报警低低限

模拟量测点的报警低低限。

ZH 报警高高限

模拟量测点的报警高高限。

DB 死区

测点的压缩死区，缺省为量程0.2 % 。

DT 死区类型

死区类型分PCT和ENG两种。

PCT - 按量程上下限的百分比。

ENG - 按工程量绝对值。

KZ 压缩类型

压缩类型分为：例外报告，线性压缩，无压缩。

例外报告：如果值的变化量超出死区范围，存入归档，否则抛弃。

线性压缩：值的变化量超出线性变化的死区带，存入归档，否则抛弃。

TT 统计类型

系统保留。

TP 统计时间周期

系统保留。

TO 统计时间偏移

系统保留。

KT 计算类型

变化计算：PT=CALC时起作用，随计算公式中数据源变化而变化。

每秒计算：PT=CALC时起作用，根据计算公式每秒计算。

EX 计算公式

5.2. 计算公式配置详解

5.2.1. 支持的语法符号

算术符号：-, +, *, /, %, ^,

逻辑符号：&, |, ~, !, &&, ||

比较符号：==, >, <, >=, <=, ~=

嵌套符号：(,), {, }

关键字：and, or, not, if, else

其他运算符：#,..

函数：openPlant®函数,水和水蒸汽函数,Lua 语句和标准库

数据库测点：W3.Node.Point

5.2.2. 计算点示例

5.2.2.1. 主要语句类型示例

```
1.  return op.value("w3.SYS.VOLTOTAL")- op.value("w3.SYS.VOLFREE")
2.  return op.value("w3.SYS.VOLTOTAL")+ op.value("w3.SYS.VOLFREE")
3.  return op.value("w3.SYS.VOLFREE") / op.value("w3.SYS.VOLTOTAL") * 100
4.  return 55
```

5.2.2.2. If-else 逻辑判断语句

```
if(op.value("W3.NODE.AX")>op.value("W3.NODE.I2")) then
    v=op.value("W3.NODE.AX")
else
    v=8
end
return v
```

5.2.2.3. for 循环

```
sum=0
for i = 1, 100  do
    sum=sum+i
    return sum+op.value("W3.AAA.AA3")
end
```

5.3. 系统函数库

函数	参数	说明
math 库		
math.abs(x)	x	返回 x 的绝对值
math.acos(x)	x	返回 x 的反余弦值（用弧度表示）
math.asin (x)	x	返回 x 的反正弦值（用弧度表示）
math.atan (y [, x])	y,x	返回 y/x 的反正切值（用弧度表示）

math.ceil (x)	x	返回不小于 x 的最小整数值
math.cos (x)	x	返回 x 的余弦（假定参数是弧度）
math.deg (x)	x	将角 x 从弧度转换为角度
math.exp (x)	x	返回 ex 的值 （e 为自然对数的底）
math.floor (x)	x	返回不大于 x 的最大整数值
math.fmod (x, y)	x,y	返回 x 除以 y，将商向零圆整后的余数。(integer/float)
math.huge		浮点数 HUGE_VAL， 这个数比任何数字值都大。
math.log (x [, base])	x,base	返回以指定底的 x 的对数
math.max (x, ・・・)	x,y…	返回参数中最大的值， 大小由 Lua 操作
math.maxinteger		最大值的整数
math.min (x, ・・・)	x,y…	返回参数中最小的值， 大小由 Lua 操作
math.mininteger		最小值的整数
math.modf (x)	x	返回 x 的整数部分和小数部分
math.pi		π 的值
math.rad (x)	x	将角 x 从角度转换为弧度
math.random ([m [, n]])	m,n	返回伪随机值
math.randomseed (x)	x	把 x 设为伪随机数发生器的“种子”： 相同的种子产生相同的随机数列
math.sin (x)	x	返回 x 的正弦值
math.sqrt (x)	x	返回 x 的平方根
math.tan (x)	x	返回 x 的正切值（假定参数是弧度）。
math.tointeger (x)	x	如果 x 可以转换为一个整数， 返回该整数
math.type (x)	x	返回参数的类型
math.ult (m, n)	m,n	如果整数 m 和 n 以无符号整数形式比较， m 在 n 之下，返回布尔真否则返回假。
string 库		
string.byte (s [, i [, j]])	s[i]	返回字符 s[i]， s[i+1]， ... ，s[j] 的内部数字编码
string.char (・・・)		返回和参数数量相同长度的字符串
string.dump (function [, strip])	fuction	返回包含有以二进制方式表示的（一个 二进制代码块 ）指定函数的字符串
string.find (s, pattern [, init [, plain]])	s,pattern	查找第一个字符串 s 中匹配到的 pattern
string.format (formatstring, ・・・)	formatstring	返回不定数量参数的格式化版本
string.gmatch (s, pattern)	s,pattern	返回一个迭代器函数
string.gsub (s, pattern, repl [, n])	s,pattern,repl	将字符串 s 中，所有的 pattern 都替换成 repl
string.len (s)	s	接收一个字符串，返回其长度
string.lower (s)	s	接收一个字符串，将其中的大写字符都转为小写后返回其副本
string.match (s, pattern [, init])	s,pattern	在字符串 s 中找到第一个能用 pattern 匹配到的部分
string.pack (fmt, v1, v2, ・・・)	fmt,v1,v2	返回一个打包了 v1, v2 等值的二进制字符串

string.packsize (fmt)	fmt	返回以指定格式用 string.pack 打包的字符串的长度
string.rep (s, n [, sep])	s,n	返回 n 个字符串 s 以字符串 sep 为分割符连在一起的字符串
string.reverse (s)	s	返回字符串 s 的翻转串
string.sub (s, i [, j])	s,i	返回 s 的子串， 该子串从 i 开始到 j 为止； i 和 j 都可以为负数。
string.unpack (fmt, s [, pos])	fmt,s	返回以格式 fmt 打包在字符串 s 中的值。
string.upper (s)	s	接收一个字符串，将其中的小写字符都转为大写后返回其副本
table 库		
table.concat (list [, sep [, i [, j]]])	list	提供一个列表，其所有元素都是字符串或数字
table.insert (list, [pos,] value)	list,pos,value	在 list 的位置 pos 处插入元素 value
table.move (a1, f, e, t [,a2])	a1,a2	将元素从表 a1 移到表 a2
table.pack (•••)		返回用所有参数以键 1,2, 等填充的新表
table.remove (list [, pos])	list pos	移除 list 中 pos 位置上的元素,并返回这个被移除的值
table.sort (list [, comp])	list	在表内从 list[1] 到 list[#list] 原地 对其间元素按指定次序排序
table.unpack (list [, i [, j]])	list	返回列表中的元素
io 库		
io.close ([file])	file	关闭文件
io.flush ()		等价于 io.output():flush()。
io.input ([file])	file	用文件名调用它时，（以文本模式）来打开该名字的文件， 并将文件句柄设为默认输入文件
io.lines ([filename •••])	filename	以读模式打开指定的文件名并返回一个迭代函数
io.open (filename [, mode])	filename,mode	这个函数用字符串 mode 指定的模式打开一个文件。
io.popen (prog [, mode])	prog	用一个分离进程开启程序 prog,
io.read (•••)		等价于 io.input():read(•••)。
io.tmpfile ()		返回一个临时文件的句柄
io.type (obj)	obj	检查 obj 是否是合法的文件句柄
io.write (•••)		等价于 io.output():write(•••)。
os 库		
os.clock ()		返回程序使用的按秒计 CPU 时间的近似值。
os.date ([format [, time]])	formate	返回一个包含日期及时刻的字符串或表
os.difftime (t2, t1)	t2,t1	返回以秒计算的时刻 t1 到 t2 的差值
os.execute ([command])	command	调用系统解释器执行 command
os.exit ([code [, close]])	code,close	调用 ISO C 函数 exit 终止宿主程序
os.getenv (varname)	varname	返回进程环境变量 varname 的值,
os.remove (filename)	filename	删除指定名字的文件
os.rename (oldname, newname)	oldname,newname	将名字为 oldname 的文件或目录更名为 newname
os.setlocale (locale [,	locale	设置程序的当前区域

category])		
os.time ([table])	table	返回给定表或者当前的时刻
os.tmpname ()		返回一个可用于临时文件的文件名字符串。

5.4. openPlant®函数

分类	函数	功能说明	调用示例	计算公式 EX 示例
数学	sinh(x)	双曲正弦	op.sinh(x)	return op.sinh(x)
	cosh(x)	双曲余弦	op.cosh(x)	return op.cosh(x)
	tanh(x)	双曲正切	op.tanh(x)	return op.tanh(x)
测点	value(tag)	测点数值	op.value(tag);	return op.value("TAG") 或者 op.value(id)
	status(tag)	测点状态	op.dynamic(tag)	ds = op.status("TAG") return ds:alarm()
	time(tag)	测点更新时间	op.time(tag)	return op.time("TAG")
	get(tag, key)	测点字段属性	op.get(key, "AV")	return op.get("TAG", "AV")
	static(tag)	静态对象	op.static(tag)	info = op.static("TAG") return info.tv
	dynamic(tag)	动态对象	op.dynamic(tag)	dyn = op.dynamic("TAG") return dyn.av
	rate(tag, interval)	数值变化率	op.rate(tag, interval)	return op.rate("TAG", 1)
	acc(tag, period, offset)	周期累计值	op.acc(tag, period, offset)	return op.acc("TAG", '1d')
状态	good()	质量好	ds:good()	ds = op.status("TAG") return ds:alarm()
	forced()	手工强制	ds:forced()	ds = op.status("TAG") return ds:forced()
	bad()	质量坏	ds:bad()	ds = op.status("TAG") return ds:bad()
	timeout()	超时	ds:timeout()	ds = op.status("TAG") return ds:timeout()
	alarm()	报警状态	ds:alarm()	ds = op.status("TAG") return ds:alarm()
	level()	报警级别	ds:level()	ds = op.status("TAG") return ds:level()
	inhibit()	报警抑制	ds:inhibit()	ds = op.status("TAG") return ds:inhibit()
	unack()	报警未确认	ds:unack()	ds = op.status("TAG") return ds:unack()
快照	snapshot(tags, time)	已知 tags，time，获取数据快照的时间状态和值	op.snapshot(tags, time)， 结果集：rs={id,time,status,value}}	t= op.date("2017-04-20 10:34:00") rs=op.snapshot("W3.NODE.TEST",t) return rs.value 支持 rs.value,rs.status,rs.time,rs.id。 注意:时间参数需要早于最后更新时间。

历史表	archive(tags, begin, end)	已知 tags（测点名）begin（开始时间),end(结束时间),通过 op.archive()获取历史数据	函数 op.archive(tags, begin, end)), 结果集：rs={error,count,{id,time,status,value}}	t1=op.date("2017-04-20 10:10:00") t2=op.date("2017-04-20 10:31:00") 示例 1： rs = op.archive("W3.NODE.TEST", t1,t2) return rs.count 示例 2： rs = op.archive("W3.NODE.TEST", t1,t2) return rs[700].value 示例 3： points={"W3.EE.QQ","W3.EE.DD"} rs = op.archive(points,t1,t2) return rs[1][8].value 支持 rs.error,rs.value,rs.status,rs.time,rs.id,rs.count.
	plot(tags, begin, end, interval)	已知 tags（测点名）begin（开始时间）,end(结束时间) interval（间隔）通过 op.plot()获取绘图值	函数 op.plot(tags, begin, end,interval), 结果集：rs={error,count,{id,time,status,value}}	t1=op.date("2017-04-20 10:10:00") t2=op.date("2017-04-20 10:31:00") rs = op.plot("W3.NODE.TEST", t1,t2,5) return rs[400].value 其他返回和示例参考 op.archive()。
	span(tags, begin, end, interval)	已知 keys（测点名）begin（开始时间）,end(结束时间) interval（间隔）通过 opplot 获取等间距值	函数 op.span(tags, begin, end,interval), 结果集：rs={error,count,{id,time,status,value}}	t1=op.date("2017-04-20 10:10:00") t2=op.date("2017-04-20 10:31:00") rs = op.span("W3.NODE.TEST", t1,t2,5) return rs[100].value 其他返回和示例参考 op.archive()。
	stat(tags, begin, end)	已知 tags(点名)，begin（开始时间），end（结束时间），通过 op.stat()获取指定时间段内所有统计值:累计/最大/最小/平均/方差	op.stat(),结果集：rs={error,id,time,status, flow,max,min,maxtime,mintime,sum,avg,mean,stdev}	t1=op.date("2017-04-20 10:10:00") t2=op.date("2017-04-20 10:31:00") rs = op.stat("W3.NODE.TEST", t1,t2) return rs.flow 支持返回结果 rs.max, rs.min,rs.maxtime,rs.mintime,rs.sum,rs.avg,rs.mean, rs.stdev,rs.status,rs.time,rs.id,rs.error,rs.flow,rs.max。 其他示例参考 op.archive()。
	max(tags, begin, end)	已知 tags(点名)，begin（开始时间），end（结束时间），通过 op.max()获取指定时间段内的最大值	op.max(), 结果集：rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") 示例 1： rs=op.max("W3.NODE.TEST",t1,t2) return rs.value 示例 2： gp={"W3.EE.QQ","W3.EE.DD"} rs=op.max(gp,t1,t2) return rs[1].value 支持 rs.error,rs.value,rs.status,rs.time,rs.id.

	min(tags, begin, end)	已知 tags(点名), begin（开始时间），end（结束时间），通过 op.min()获取指定时间段内的最小值	op.min(), 结果集: rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") rs=op.min("W3.NODE.TEST",t1,t2) return rs.value 其他示例和返回参考 op.max()。
	avg(tags, begin, end)	已知 tags(点名), begin（开始时间），end（结束时间），通过 op.avg()获取指定时间段内的平均值	op.min(), 结果集: rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") rs=op.avg("W3.NODE.TEST",t1,t2) return rs.value 其他示例和返回参考 op.max()。
	mean(tags, begin, end)	已知 tags(点名), begin（开始时间），end（结束时间），通过 op.mean()获取指定时间段内的算术平均值	op.mean(), 结果集: rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") rs=op.mean("W3.NODE.TEST",t1,t2) return rs.value 其他示例和返回参考 op.max()。
	sum(tags, begin, end)	已知 tags(点名), begin（开始时间），end（结束时间），通过 op.sum()获取指定时间段内的算数和	op.sum(), 结果集: rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") rs=op.sum("W3.NODE.TEST",t1,t2) return rs.value 其他示例和返回参考 op.max()。
	flow(tags, begin, end)	已知 tags(点名), begin（开始时间），end（结束时间），通过 op.flow()获取指定时间段内的流量值	op.flow(), 结果集: rs={error,id,time,status,value}	t1=op.date("2017-04-20 10:00:00") t2=op.date("2017-04-20 10:30:00") rs=op.flow("W3.NODE.TEST",t1,t2) return rs.value 其他示例和返回参考 op.max()。
系统	cacheq	历史归档队列数量	op.cacheq()	return op.cacheq("")
	event	实时事件队列数量	op.event()	return op.event()
	dbload	数据库瞬时负荷	op.dbload()	return op.dbload()
	dbmem	数据库使用内存	op.dbmem()	return op.dbmem()
	load	系统负荷	op.load()	return op.load()
	memfree	可用内存	op.memfree()	return op.memfree()
	memtotal	系统内存	op.memtotal()	return op.memtotal()
	volfree	数据库使用空间	op.volfree()	return op.volfree()
	voltotal	数据库磁盘空间	op.voltotal()	return op.voltotal()
	counter	计数器	op.counter()	return op.counter()
	uptime	数据库运行时间	op.uptime()	return op.uptime()
	session	活动会话数	op.session()	return op.session()
	session_peak	峰值会话数	op.session_peak()	return op.session_peak()
	calc_time	周期计算的测点执行计算耗时(毫秒)	op.calc_time()	return op.calc_time()
	thread	线程数量	op.thread()	return op.thread()

	idle	空闲线程数量	op.idle()	return op.idle()
	ping	ping 指定 IP，返回 0 或 1	op.ping()	return op.ping("192.168.2.230")
时间	now	通过 op.now(),获取系统当前时间,测点类型需要为 R8	op.now()	return op.now()
	today	通过 op.today()，获取当天日期	op.today()	return op.today()
	date	获取指定时间对应的时间戳，参数格式("2015-08-26 00:00:00")	op.date(y,m,d,h,m,s) ｜ date("2015-08-26 00:00:00")	格式：op.date(y,m,d,h,m,s) return op.date("2015-08-26 00:00:00")
	bday	已知 dt（时间），获取对应的日期	op.bday(dt)	当前日期：2017-04-06 示例：dt=op.today() today=op.bday(dt) return today 结果：1491408000，即 2017-04-06 00:00:00
	bmonth	已知 dt（时间），获取当月第一天的日期时间戳	op.bmonth(dt)	当前日期：2017-04-06 示例：dt=op.today() month=op.bmonth(dt) return month 结果：1490976000，即 2017-04-01 00:00:00
	bnextmonth	已知 dt（时间），获取下一个月第一天的日期时间戳	op.bnextmonth(dt)	当前日期：2017-04-06 示例：dt=op.today() month=op.bnextmonth(dt) return month 结果：1493568000，即 2017-05-01 00:00:00
	timeadd	已知 h（时间格式），1(相加数字)，dt（时间）输出结果为:在 dt 的时间上加 1 小时	op.timeadd("h", 1, dt)	当前时间 2017-04-06 11:30:40，获取当前时间，小时加 2。 示例：dt=op.now() hour=op.timeadd("h", 2, dt) return hour 结果：1491456640，即 2017-04-06 13:30:40
	timediff	已知 h（时间格式),t1(时间),t2(时间)，输出结果为 t1 和 t2 之间相差小时数	op.timediff("h", t1, t2)	示例： t1=op.date("2017-04-06 10:30:40") t2=op.date("2017-04-06 13:30:40") hour=op.timediff("h", t1, t2) return hour 结果：-3
	year	调用 op.year()获取条件日期对应的年份	op.year()	示例： dt=op.now() year=op.year(dt) return year 结果：返回当前日期对应的年份
	month	调用 op.month()获取条件对应的月份	op.month()	示例： dt=op.now() month=op.month(dt) return month 结果：返回当前日期对应的月份

	day	调用 op.day()获取当前日期对应月份的天数	op.day()	示例： dt=op.now() day=op.day(dt) return day 结果：返回当前月份对应的天数
	hour	调用 op.hour()获取输入时间对应的小时	op.hour()	dt=op.now() hour=op.hour(dt) return hour 结果：返回当前时间对应的小时（24 小时制）
	minute	调用 op.minute()获取输入时间对应的分钟	op.minute()	dt=op.now() minute=op.minute(dt) return minute 结果：返回当前小时对应的分钟
	second	调用 op.second()获取输入时间对应的秒	op.second()	dt=op.now() second=op.second(dt) return second 结果：返回当前分钟对应的秒
	format	获取具体时间格式，可调用 op.format(),时间格式参照参数中所写	op.format("yyyy-MM-dd HH:mm:ss")，时间字符串	示例： dt=op.now() format=op.format(dt,"yyyy-MM-dd HH:mm:ss") return format 结果：该属性返回结果为字符串，不可直接在测点计算公式中使用

5.5. 水和水蒸汽函数

名称	参数	函数说明
if97.p2t	(P)	已知压力(P),返回对应的饱和温度(T)
ifc67.p2t	(P)	已知压力(P),返回对应的饱和温度(T)
if97.p2hl	(P)	已知压力(P),返回对应的饱和水比焓(HL)
ifc67.p2hl	(P)	已知压力(P),返回对应的饱和水比焓(HL)
if97.p2hg	(P)	已知压力(P),返回对应的饱和汽比焓(HG)
ifc67.p2hg	(P)	已知压力(P),返回对应的饱和汽比焓(HG)
if97.p2sl	(P)	已知压力(P),返回对应的饱和水比熵(SL)
ifc67.p2sl	(P)	已知压力(P),返回对应的饱和水比熵(SL)
if97.p2sg	(P)	已知压力(P),返回对应的饱和汽比熵(SG)
ifc67.p2sg	(P)	已知压力(P),返回对应的饱和汽比熵(SG)
if97.p2vl	(P)	已知压力(P),返回对应的饱和水比容(VL)
ifc67.p2vl	(P)	已知压力(P),返回对应的饱和水比容(VL)
if97.p2vg	(P)	已知压力(P),返回对应的饱和汽比容(VG)
ifc67.p2vg	(P)	已知压力(P),返回对应的饱和汽比容(VG)
if97.p2cpl	(P)	已知压力(P),返回对应的饱和水定压比热容(CpL)
ifc67.p2cpl	(P)	已知压力(P),返回对应的饱和水定压比热容(CpL)
if97.p2cpg	(P)	已知压力(P),返回对应的饱和汽定压比热容(CpG)
ifc67.p2cpg	(P)	已知压力(P),返回对应的饱和汽定压比热容(CpG)
if97.p2cvl	(P)	已知压力(P),返回对应的饱和水定容比热容(CvL)
ifc67.p2cvl	(P)	已知压力(P),返回对应的饱和水定容比热容(CvL)
if97.p2cvg	(P)	已知压力(P),返回对应的饱和汽定容比热容(CvG)
ifc67.p2cvg	(P)	已知压力(P),返回对应的饱和汽定容比热容(CvG)
if97.p2ul	(P)	已知压力(P),返回对应的饱和水运动粘度(UL)
ifc67.p2ul	(P)	已知压力(P),返回对应的饱和水运动粘度(UL)

if97.p2ug	(P)	已知压力(P),返回对应的饱和汽运动粘度(UG)
ifc67.p2ug	(P)	已知压力(P),返回对应的饱和汽运动粘度(UG)
if97.pt2h	(P,T)	已知压力(P)和温度(T),返回对应的比焓(H)
ifc67.pt2h	(P,T)	已知压力(P)和温度(T),返回对应的比焓(H)
if97.pt2s	(P,T)	已知压力(P)和温度(T),返回对应的比熵(S)
ifc67.pt2s	(P,T)	已知压力(P)和温度(T),返回对应的比熵(S)
if97.pt2v	(P,T)	已知压力(P)和温度(T),返回对应的比容(V)
ifc67.pt2v	(P,T)	已知压力(P)和温度(T),返回对应的比容(V)
if97.pt2x	(P,S)	已知压力(P)和温度(T),返回对应的干度(X)
ifc67.pt2x	(P,S)	已知压力(P)和温度(T),返回对应的干度(X)
if97.pt2cp	(P,T)	已知压力(P)和温度(T),返回对应定压质量比热容(Cp)
ifc67.pt2cp	(P,T)	已知压力(P)和温度(T),返回对应定压质量比热容(Cp)
if97.pt2cv	(P,T)	已知压力(P)和温度(T),返回对应定容质量比热容(Cv)
ifc67.pt2cv	(P,T)	已知压力(P)和温度(T),返回对应定容质量比热容(Cv)
if97.pt2u	(P,T)	已知压力(P)和温度(T),返回对应的运动粘度(U)
ifc67.pt2u	(P,T)	已知压力(P)和温度(T),返回对应的运动粘度(U)
if97.ph2t	(P,H)	已知压力(P)和比焓(H),返回对应的温度(T)
ifc67.ph2t	(P,H)	已知压力(P)和比焓(H),返回对应的温度(T)
if97.ph2s	(P,H)	已知压力(P)和比焓(H),返回对应的比熵(S)
ifc67.ph2s	(P,H)	已知压力(P)和比焓(H),返回对应的比熵(S)
if97.ph2v	(P,H)	已知压力(P)和比焓(H),返回对应的比容(V)
ifc67.ph2v	(P,H)	已知压力(P)和比焓(H),返回对应的比容(V)
if97.ph2x	(P,H)	已知压力(P)和比焓(H),返回对应的干度(X)
ifc67.ph2x	(P,H)	已知压力(P)和比焓(H),返回对应的干度(X)
if97.ps2t	(P,S)	已知压力(P)和比熵(S),返回对应的温度(T)
ifc67.ps2t	(P,S)	已知压力(P)和比熵(S),返回对应的温度(T)
if97.ps2h	(P,S)	已知压力(P)和比熵(S),返回对应的比焓(H)
ifc67.ps2h	(P,S)	已知压力(P)和比熵(S),返回对应的比焓(H)
if97.ps2v	(P,S)	已知压力(P)和比熵(S),返回对应的比容(V)
ifc67.ps2v	(P,S)	已知压力(P)和比熵(S),返回对应的比容(V)
if97.ps2x	(P,S)	已知压力(P)和比熵(S),返回对应的干度(X)
ifc67.ps2x	(P,S)	已知压力(P)和比熵(S),返回对应的干度(X)
if97.pv2t	(P,V)	已知压力(P)和比容(V),返回对应的温度(T)
ifc67.pv2t	(P,V)	已知压力(P)和比容(V),返回对应的温度(T)
if97.pv2h	(P,V)	已知压力(P)和比容(V),返回对应的比焓(H)
ifc67.pv2h	(P,V)	已知压力(P)和比容(V),返回对应的比焓(H)
if97.pv2s	(P,V)	已知压力(P)和比容(V),返回对应的比熵(S)
ifc67.pv2s	(P,V)	已知压力(P)和比容(V),返回对应的比熵(S)
if97.pv2x	(P,V)	已知压力(P)和比容(V),返回对应的干度(X)
ifc67.pv2x	(P,V)	已知压力(P)和比容(V),返回对应的干度(X)
if97.px2t	(P,X)	已知压力(P)和干度(X),返回对应的温度(T)
ifc67.px2t	(P,X)	已知压力(P)和干度(X),返回对应的温度(T)
if97.px2h	(P,X)	已知压力(P)和干度(X),返回对应的比焓(H)
ifc67.px2h	(P,X)	已知压力(P)和干度(X),返回对应的比焓(H)

if97.px2s	(P,X)	已知压力(P)和干度(X),返回对应的比熵(S)
ifc67.px2s	(P,X)	已知压力(P)和干度(X),返回对应的比熵(S)
if97.px2v	(P,X)	已知压力(P)和干度(X),返回对应的比容(V)
ifc67.px2v	(P,X)	已知压力(P)和干度(X),返回对应的比容(V)
if97.t2p	(T)	已知温度(T),返回对应的饱和压力(P)
ifc67.t2p	(T)	已知温度(T),返回对应的饱和压力(P)
if97.t2hl	(T)	已知温度(T),返回对应的饱和水比焓(HL)
ifc67.t2hl	(T)	已知温度(T),返回对应的饱和水比焓(HL)
if97.t2hg	(T)	已知温度(T),返回对应的饱和汽比焓(HG)
ifc67.t2hg	(T)	已知温度(T),返回对应的饱和汽比焓(HG)
if97.t2sl	(T)	已知温度(T),返回对应的饱和水比熵(SL)
ifc67.t2sl	(T)	已知温度(T),返回对应的饱和水比熵(SL)
if97.t2sg	(T)	已知温度(T),返回对应的饱和汽比熵(SG)
ifc67.t2sg	(T)	已知温度(T),返回对应的饱和汽比熵(SG)
if97.t2vl	(T)	已知温度(T),返回对应的饱和水比容(VL)
ifc67.t2vl	(T)	已知温度(T),返回对应的饱和水比容(VL)
if97.t2vg	(T)	已知温度(T),返回对应的饱和汽比容(VG)
ifc67.t2vg	(T)	已知温度(T),返回对应的饱和汽比容(VG)
if97.t2cpl	(T)	已知温度(T),返回对应的饱和水定压比热容(CpL)
ifc67.t2cpl	(T)	已知温度(T),返回对应的饱和水定压比热容(CpL)
if97.t2cpg	(T)	已知温度(T),返回对应的饱和汽定压比热容(CpG)
ifc67.t2cpg	(T)	已知温度(T),返回对应的饱和汽定压比热容(CpG)
if97.t2cvl	(T)	已知温度(T),返回对应的饱和水定容比热容(CvL)
ifc67.t2cvl	(T)	已知温度(T),返回对应的饱和水定容比热容(CvL)
if97.t2cvg	(T)	已知温度(T),返回对应的饱和汽定容比热容(CvG)
ifc67.t2cvg	(T)	已知温度(T),返回对应的饱和汽定容比热容(CvG)
if97.t2ul	(T)	已知温度(T),返回对应的饱和水运动粘度(UL)
ifc67.t2ul	(T)	已知温度(T),返回对应的饱和水运动粘度(UL)
if97.t2ug	(T)	已知温度(T),返回对应的饱和汽运动粘度(UG)
ifc67.t2ug	(T)	已知温度(T),返回对应的饱和汽运动粘度(UG)
if97.th2plp	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(低压)
ifc67.th2plp	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(低压)
if97.th2slp	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(低压)
ifc67.th2slp	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(低压)
if97.th2vlp	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(低压)
ifc67.th2vlp	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(低压)
if97.th2php	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(高压)
ifc67.th2php	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(高压)
if97.th2shp	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(高压)
ifc67.th2shp	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(高压)
if97.th2vhp	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(高压)
ifc67.th2vhp	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(高压)
if97.th2p	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(缺省低压)
ifc67.th2p	(T,H)	已知温度(T)和比焓(H),返回对应的压力(P)(缺省低压)

if97.th2s	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(缺省低压)
ifc67.th2s	(T,H)	已知温度(T)和比焓(H),返回对应的比熵(S)(缺省低压)
if97.th2v	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(缺省低压)
ifc67.th2v	(T,H)	已知温度(T)和比焓(H),返回对应的比容(V)(缺省低压)
if97.th2x	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)
ifc67.th2x	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)
if97.th2xlp	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)(低压)
ifc67.th2xlp	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)(低压)
if97.th2xhp	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)(高压)
ifc67.th2xhp	(T,H)	已知温度(T)和比焓(H),返回对应的干度(X)(高压)
if97.ts2plp	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(低压)
ifc67.ts2plp	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(低压)
if97.ts2hlp	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(低压)
ifc67.ts2hlp	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(低压)
if97.ts2vlp	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(低压)
ifc67.ts2vlp	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(低压)
if97.ts2php	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(高压)
ifc67.ts2php	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(高压)
if97.ts2hhp	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(高压)
ifc67.ts2hhp	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(高压)
if97.ts2vhp	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(高压)
ifc67.ts2vhp	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(高压)
if97.ts2p	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(缺省低压)
ifc67.ts2p	(T,S)	已知温度(T)和比熵(S),返回对应的压力(P)(缺省低压)
if97.ts2h	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(缺省低压)
ifc67.ts2h	(T,S)	已知温度(T)和比熵(S),返回对应的比焓(H)(缺省低压)
if97.ts2v	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(缺省低压)
ifc67.ts2v	(T,S)	已知温度(T)和比熵(S),返回对应的比容(V)(缺省低压)
if97.ts2x	(T,S)	已知温度(T)和比熵(S),返回对应的干度(X)
ifc67.ts2x	(T,S)	已知温度(T)和比熵(S),返回对应的干度(X)
if97.tv2p	(T,V)	已知温度(T)和比容(V),返回对应的压力(P)
ifc67.tv2p	(T,V)	已知温度(T)和比容(V),返回对应的压力(P)
if97.tv2h	(T,V)	已知温度(T)和比容(V),返回对应的比焓(H)
ifc67.tv2h	(T,V)	已知温度(T)和比容(V),返回对应的比焓(H)
if97.tv2s	(T,V)	已知温度(T)和比容(V),返回对应的比熵(S)
ifc67.tv2s	(T,V)	已知温度(T)和比容(V),返回对应的比熵(S)
if97.tv2x	(T,V)	已知温度(T)和比容(V),返回对应的干度(X)
ifc67.tv2x	(T,V)	已知温度(T)和比容(V),返回对应的干度(X)
if97.tx2p	(T,X)	已知温度(T)和干度(X),返回对应的压力(P)
ifc67.tx2p	(T,X)	已知温度(T)和干度(X),返回对应的压力(P)
if97.tx2h	(T,X)	已知温度(T)和干度(X),返回对应的比焓(H)
ifc67.tx2h	(T,X)	已知温度(T)和干度(X),返回对应的比焓(H)
if97.tx2s	(T,X)	已知温度(T)和干度(X),返回对应的比熵(S)
ifc67.tx2s	(T,X)	已知温度(T)和干度(X),返回对应的比熵(S)

if97.tx2v	(T,X)	已知温度(T)和干度(X),返回对应的比容(V)
ifc67.tx2v	(T,X)	已知温度(T)和干度(X),返回对应的比容(V)
if97.hs2p	(H,S)	已知比焓(H)和比熵(S),返回对应的压力(P)
ifc67.hs2p	(H,S)	已知比焓(H)和比熵(S),返回对应的压力(P)
if97.hs2t	(H,S)	已知比焓(H)和比熵(S),返回对应的温度(T)
ifc67.hs2t	(H,S)	已知比焓(H)和比熵(S),返回对应的温度(T)
if97.hs2v	(H,S)	已知比焓(H)和比熵(S),返回对应的比容(V)
ifc67.hs2v	(H,S)	已知比焓(H)和比熵(S),返回对应的比容(V)
if97.hs2x	(H,S)	已知比焓(H)和比熵(S),返回对应的干度(X)
ifc67.hs2x	(H,S)	已知比焓(H)和比熵(S),返回对应的干度(X)
if97.hv2p	(H,V)	已知比焓(H)和比容(V),返回对应的压力(P)
ifc67.hv2p	(H,V)	已知比焓(H)和比容(V),返回对应的压力(P)
if97.hv2t	(H,V)	已知比焓(H)和比容(V),返回对应的温度(T)
ifc67.hv2t	(H,V)	已知比焓(H)和比容(V),返回对应的温度(T)
if97.hv2s	(H,V)	已知比焓(H)和比容(V),返回对应的比熵(S)
ifc67.hv2s	(H,V)	已知比焓(H)和比容(V),返回对应的比熵(S)
if97.hv2x	(H,V)	已知比焓(H)和比容(V),返回对应的干度(X)
ifc67.hv2x	(H,V)	已知比焓(H)和比容(V),返回对应的干度(X)
if97.hx2plp	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(低压)
ifc67.hx2plp	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(低压)
if97.hx2tlp	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(低压)
ifc67.hx2tlp	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(低压)
if97.hx2slp	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(低压)
ifc67.hx2slp	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(低压)
if97.hx2vlp	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(低压)
ifc67.hx2vlp	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(低压)
if97.hx2php	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(高压)
ifc67.hx2php	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(高压)
if97.hx2thp	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(高压)
ifc67.hx2thp	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(高压)
if97.hx2shp	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(高压)
ifc67.hx2shp	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(高压)
if97.hx2vhp	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(高压)
ifc67.hx2vhp	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(高压)
if97.hx2p	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(缺省低压)
ifc67.hx2p	(H,X)	已知比焓(H)和干度(X),返回对应的压力(P)(缺省低压)
if97.hx2t	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(缺省低压)
ifc67.hx2t	(H,X)	已知比焓(H)和干度(X),返回对应的温度(T)(缺省低压)
if97.hx2s	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(缺省低压)
ifc67.hx2s	(H,X)	已知比焓(H)和干度(X),返回对应的比熵(S)(缺省低压)
if97.hx2v	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(缺省低压)
ifc67.hx2v	(H,X)	已知比焓(H)和干度(X),返回对应的比容(V)(缺省低压)
if97.sv2p	(S,V)	已知比熵(S)和比容(V),返回对应的压力(P)
ifc67.sv2p	(S,V)	已知比熵(S)和比容(V),返回对应的压力(P)

if97.sv2t	(S,V)	已知比熵(S)和比容(V),返回对应的温度(T)
ifc67.sv2t	(S,V)	已知比熵(S)和比容(V),返回对应的温度(T)
if97.sv2h	(S,V)	已知比熵(S)和比容(V),返回对应的比焓(H)
ifc67.sv2h	(S,V)	已知比熵(S)和比容(V),返回对应的比焓(H)
if97.sv2x	(S,V)	已知比熵(S)和比容(V),返回对应的干度(X)
ifc67.sv2x	(S,V)	已知比熵(S)和比容(V),返回对应的干度(X)
if97.sx2plp	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(低压)
ifc67.sx2plp	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(低压)
if97.sx2tlp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(低压)
ifc67.sx2tlp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(低压)
if97.sx2hlp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(低压)
ifc67.sx2hlp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(低压)
if97.sx2vlp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(低压)
ifc67.sx2vlp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(低压)
if97.sx2pmp	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(中压)
ifc67.sx2pmp	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(中压)
if97.sx2tmp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(中压)
ifc67.sx2tmp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(中压)
if97.sx2hmp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(中压)
ifc67.sx2hmp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(中压)
if97.sx2vmp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(中压)
ifc67.sx2vmp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(中压)
if97.sx2php	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(高压)
ifc67.sx2php	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(高压)
if97.sx2thp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(高压)
ifc67.sx2thp	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(高压)
if97.sx2hhp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(高压)
ifc67.sx2hhp	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(高压)
if97.sx2vhp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(高压)
ifc67.sx2vhp	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(高压)
if97.sx2p	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(缺省低压)
ifc67.sx2p	(S,X)	已知比熵(S)和干度(X),返回对应的压力(P)(缺省低压)
if97.sx2t	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(缺省低压)
ifc67.sx2t	(S,X)	已知比熵(S)和干度(X),返回对应的温度(T)(缺省低压)
if97.sx2h	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(缺省低压)
ifc67.sx2h	(S,X)	已知比熵(S)和干度(X),返回对应的比焓(H)(缺省低压)
if97.sx2v	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(缺省低压)
ifc67.sx2v	(S,X)	已知比熵(S)和干度(X),返回对应的比容(V)(缺省低压)
if97.vx2plp	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(低压)
ifc67.vx2plp	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(低压)
if97.vx2tlp	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(低压)
ifc67.vx2tlp	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(低压)
if97.vx2hlp	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(低压)
ifc67.vx2hlp	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(低压)

if97.vx2slp	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(低压)
ifc67.vx2slp	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(低压)
if97.vx2php	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(高压)
ifc67.vx2php	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(高压)
if97.vx2thp	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(高压)
ifc67.vx2thp	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(高压)
if97.vx2hhp	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(高压)
ifc67.vx2hhp	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(高压)
if97.vx2shp	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(高压)
ifc67.vx2shp	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(高压)
if97.vx2p	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(缺省低压)
ifc67.vx2p	(V,X)	已知比容(V)和干度(X),返回对应的压力(P)(缺省低压)
if97.vx2t	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(缺省低压)
ifc67.vx2t	(V,X)	已知比容(V)和干度(X),返回对应的温度(T)(缺省低压)
if97.vx2h	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(缺省低压)
ifc67.vx2h	(V,X)	已知比容(V)和干度(X),返回对应的比焓(H)(缺省低压)
if97.vx2s	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(缺省低压)
ifc67.vx2s	(V,X)	已知比容(V)和干度(X),返回对应的比熵(S)(缺省低压)

5.6. 公司业绩表

序号	用户名称	建设规模
一、电网		
1	广东电网公司电力科学研究院火电机组 实时/历史数据共享平台项目	
2	广东电网公司广东节能调度在线煤耗监测系统建设项目	
3	上海外二电厂环保监测信息接入电力调度网项目	
4	上海吴泾电厂环保监测信息接入电力调度网项目	
5	广东省佛山市南海发电一厂节能调度信息系统	
二、核电		
1	大亚湾核电站	2×1000MW
2	中广核红沿河核电站	6×1000MW
3	中广核宁德核电站	4×1000MW
4	中广核阳江核电站	6×1000MW
5	中广核台山核电站	2×1000MW
6	中广核防城港核电站	2×1000MW
三、火电		
1	大唐国际托克托发电厂	8×600MW
2	神东哈密电厂	4×660MW
3	大唐国际国际南通吕泗电厂	4×660MW

4	大唐国际国际乌沙山电厂	4×600MW
5	中电投贵州盘南电厂	4×600MW
6	华能海门电厂	2×1036MW
7	粤电集团平海电厂	2×1000MW
8	华能金陵电厂	2×1000MW
9	大唐国际潮州电厂	2×1000MW+2×600MW
10	申能上海外高桥第二发电有限公司	2×900MW
11	广东粤电集团沙角 C 电厂	3×660MW
12	湖北能源鄂州电厂	2×600MW+2×300MW
13	河北建投秦皇岛发电厂	2×200+4×300MW
14	粤电集团珠海发电厂	2×700MW
15	中电投贵州习水二郎发电有限公司	2×660MW
16	粤电集团红海湾电厂	2×660 MW
17	粤电集团沙角 A 电厂	1×200MW+2×210MW+2×300MW
18	大唐国际国际宁德电厂	4×600MW
19	大唐国际大坝发电厂	2×600MW
20	中电投白音华金山坑口电厂	2×600MW
21	河北建投西柏坡发电厂	2×600MW+4×300MW
22	粤电集团金湾发电厂	2×600MW
23	山西武乡和信电厂	2×600MW
24	大唐国际运城发电厂	2×600MW
25	华电广西贵港电厂	2×600MW
26	大唐国际王滩电厂	2×600MW
27	粤电集团靖海电厂	2×600MW
28	福煤石狮鸿山热电厂	2×600MW
29	山西同华电力轩岗电厂	2×600MW
30	中电投朝阳燕山湖发电有限公司	2×600MW
31	国投哈密发电有限公司	2×600MW
32	云投威信电厂	2×600MW
33	中电投贵州黔北电厂	4×300MW
34	粤电集团湛江电厂	4×300MW
35	粤电集团前湾电厂	3×390MW
36	华能上海燃机电厂	3×390MW
37	粤电集团韶关发电厂	2×200MW +2×330MW
38	大唐集团吉林珲春电厂	2×100MW+2×330MW
39	大唐国际绍兴燃机热电有限公司	2×400MW
40	粤电集团茂名电厂	3×100MW+1×200MW+1×300MW
41	华能海口东方电厂	2×350MW
42	华能辽宁丹东发电厂	2×350MW
43	新疆天富热电股份有限公司	2×330MW

44	大唐国际北京高井热电厂	6×110MW
45	大唐集团长山电厂	1×600MW
46	中电投吉林白山电厂	2×300MW
47	大唐集团长春第三热电厂	2×300MW
48	大唐国际锦州电厂	2×300MW
49	大唐集团辽源电厂	2×300MW
50	大唐集团哈尔滨第一热电厂	2×300MW
51	中电投抚顺发电厂	2×300MW
52	大唐集团调兵山煤矸石电厂	2×300MW
53	中电投山西永济电厂	2×300MW
54	华能天津杨柳清电厂	2×300MW
55	大唐国际红河电厂	2×300MW
56	大唐国际呼和浩特国能发电厂	2×300MW
57	蒙能锡林电厂	2×300MW
58	顺德五沙电厂	2×300MW
59	皖能淮北国安发电厂	2×300MW
60	粤电集团黄浦电厂	2×300MW
61	京能内蒙古酸刺沟矸石电厂	2×300MW
62	蒙能准大电厂	2×300MW
63	蒙能新丰电厂	2×300MW
64	中电投吉电股份四平热电厂	2×300MW
65	中电投吉电股份松花江热电厂	2×300MW
66	华能平凉电厂	2×300MW
67	华能营口电厂	2×300MW
68	华能井冈山电厂	2×300MW
69	河北建投宣化电厂	2×300MW
70	大唐国际临汾电厂	2×300MW
71	中电投石家庄良村热电厂	2×300MW
72	大唐集团河北马头发电有限责任公司	2×300MW
73	山西国际能源瑞光热电有限责任公司	2×300MW
74	粤电集团奥里油电厂	2×300MW
75	蒙东能源鄂温克电厂	2×300MW
76	蒙能呼和浩特电厂	2×300MW
77	国电贵州鸭溪电厂	2×300MW
78	粤电集团云浮电厂	2×300MW
79	石河子天富南热电有限公司	2×300MW
80	山西漳泽电力股份有限公司侯马热电分公司	2×300MW
81	山西国际电力嘉节燃气热电有限公司	2×300MW
82	粤电集团梅县电厂	2×125MW+2×135MW
83	粤电集团连州电厂	2×125MW+2×135MW
84	粤电集团云浮电厂	2×125MW+2×135MW

85	中电投南阳热电厂	2×200MW
86	大唐河北迁安发电厂	2×200MW
87	大唐河北保定热电厂	2×200MW
88	湖北武汉钢电股份公司	2×200MW
89	华能海拉尔电厂	2×200MW
90	中电投吉林浑江电厂	2×200MW
91	国网能源上海闸电燃机汽轮机发电有限公司	4×100MW
92	广西金桂纸业自备电厂	2×150MW
93	神东电力集团上湾电厂	2×150MW
94	国成能源西乌金山发电有限公司	2×150MW
95	阳城煤矿自备电厂	2×135MW
96	粤电集团罗定电厂	2×135MW
97	粤电集团阳山电厂	2×135MW
98	京能内蒙古赤峰煤矸石电厂	2×135MW
99	新疆金川热电有限责任公司	2×135MW
100	大唐国际江山燃机热电有限公司	2×115MW
101	中电投吉林电力股份有限公司二道江发电公司	2×100MW
102	神东陕西大柳塔热电厂	2×50MW
103	江苏武进亚能热电厂	2×50MW
104	济矿集团民生电厂	2×15MW+1×12MW
105	内蒙古多伦自备电厂	5 炉 3 机
四、清洁能源		
1	大唐国际彭水水电开发有限公司生产实时监测系统	5×390MW
2	粤电集团天生桥一级水电生产实时监测系统	4×300MW
3	粤电集团枫树坝发电有限责任公司生产实时监测系统	2×100MW
4	粤电集团新丰江发电有限责任公司生产实时监测系统	2×100MW
5	粤电集团长湖发电有限责任公司生产实时监测系统	1×36MW+2×40MW
6	粤电集团溪发电有限责任公司生产实时监测系统	4×36MW
7	粤电集团南水发电有限责任公司生产实时监测系统	3×25MW
8	粤电集团长潭发电有限责任公司生产实时监测系统	4×15MW
9	粤电集团流溪河水发电有限责任公司生产实时监测系统	4×12MW
10	粤电集团大埔梅江蓬辣滩水电站生产实时监测系统	4×11MW
11	国华东台风电厂	140MW
12	国华河北尚义风电厂	130MW
13	国华齐齐哈尔风电厂	50MW
五、化工行业		
1	江阴嘉盛化学工业公司	24 万吨 EPS
2	南通宏信化工有限公司	3 万吨苯酐/年
3	南通大学电气工程学院	实验室演示系统

4	内蒙古多伦煤化工项目	138 万吨煤基烯烃
5	大唐国际托克托再生资源（动力和氧化铝项目）	openPlant 1 套（25 万点）
6	大唐国际克旗化工项目（动力和化工项目）	openPlant 1 套（25 万点）
7	内蒙古大唐国际克什克腾煤制气天然气有限责任公司	煤化工动力区工况系统
六、环保行业		
1	内蒙古环保局工况在线监测系统（省级平台）	openPlant 2 套
2	内蒙古环保局工况在线监测系统（终端平台）	openPlant 60 套
3	山东省环保局工况在线监测系统（省级平台）	openPlant 2 套
4	山东省环保局工况在线监测系统（终端平台）	openPlant 71 套
5	西柏坡电厂烟气排放接入系统	openPlant 1 套
6	北京新源天净环保技术有限公司（托电 6#机组）	openPlant 1 套
7	北京新源天净环保技术有限公司（托电 7#机组）	openPlant 1 套
8	北京国电清新环保技术股份有限	openPlant 1 套
9	江西环保厅环境监测联网项目（省级平台）	openPlant 1 套
10	江西环保厅环境监测联网项目（市级平台）	openPlant 10 套
11	江西环保厅工况在线监测系统（省级平台）	openPlant 2 套
12	江西环保厅工况在线监测系统（终端平台）	openPlant 6 套
13	湖北环保厅工况在线监测系统试点（省级平台）	openPlant 2 套
14	湖北环保厅工况在线监测系统（终端平台）	openPlant 4 套
15	甘肃环保厅工况在线监测系统试点（省级平台）	openPlant 2 套
16	甘肃环保厅工况在线监测系统试点（终端平台）	openPlant 12 套
17	新疆环保厅工况在线监测系统试点（省级平台）	openPlant 2 套
18	新疆环保厅工况在线监测系统试点（终端平台）	openPlant 4 套
19	河南环保厅工况在线监测系统（终端平台）	openPlant 9 套
20	黑龙江环保厅工况在线监测系统（省级平台）	openPlant 2 套
21	黑龙江环保厅工况在线监测系统（终端平台）	openPlant 5 套
22	唐山市环保局工况监控试点项目（厂级平台）	openPlant 65 套
23	江苏省环保厅工况监控试点项目（厂级平台）	openPlant 3 套
24	广西省环保厅工况监控试点项目（厂级平台）	openPlant 4 套
七、海外市场		
1	印度嘉佳 5×660MW 发电厂监控系统	5×660MW
2	印度 APL 电站 5×660MW RMIS 系统项目	5×660MW
3	印度 APL 电站 5×660MW 电厂性能计算	5×660MW
4	印度 JHASUGUDA 集团 4×600MW 电厂性能计算	4×600MW
5	印度 TALWANDI 电厂 4×600MW 监控系统	4×600MW
6	印度 JHARSUGUDA 4×600MW 燃煤独立电站监控系统	4×600MW
7	伊拉克华事德项目专家远程诊断系统	6×330MW
8	印度 JHASUGUDA 集团 9×135MW 性能计算	9×135MW
9	印度 JHASUGUDA 9*135 1、2、3 号机组项目监控系统	9×135 MW

10	印度科瑞希纳电厂 2×660MW 监控系统	2×660MW
11	印尼亚龙湾电厂性能计算	3x315MW
12	土耳其 EREN 公司 2×600MW 电厂监控系统	2×600MW
13	约旦 SAMRA 联合循环热电站监控系统	2×390MW
14	印度 MPPL 1×135MW 燃煤电站项目监控系统	1×135MW
八、石油		
1	中石油长庆油田数字化油田生产指挥系统	openPlant 23 套（15 个油田 8 个气田）
2	中石油煤层气公司生产调度系统	openPlant 3 套
九、风电		
1	申能新能源风电信息化项目	8 个风场
2	粤黔风电集中监控平台项目	5 个风电场+6 个瓦斯电站
3	国华新能源公司风电场信息化	5 个风场
十、其他行业		
1	江苏海澜新能源有限公司	海澜能源互联网平台数据库
2	南京莱斯信息技术股份有限公司	安监监管信息平台数据库
3	雪链物联网技术服务有限公司	冷链监管信息平台数据库