**AISQL（暂定名）**

1. **架构简图**



【其中灰色部分，为自开发部分】

整个系统可分为五个主要部分：

* 数据源层

系统会支持Oracle、MySQL数据库。

* 采集层

采集层功能是完成数据采集工作。主要由collectd、pycollect（暂定名）两个组件，分别完成主机端性能数据收集和数据库端数据收集工作。

* 存储层

采用InfluxDB时序数据库作为核心存储。

* 计算层

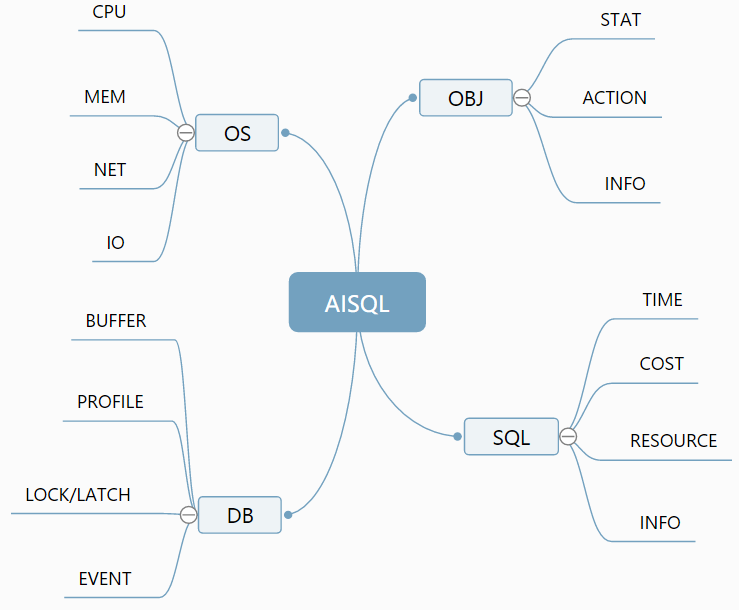
SQL\_AI（暂定名），完成算法计算部分，计算后结果回写到InfluxDB。

* 展示层

采用Grafana作为统一展示，这部分不要开发工作，但需要大量的配置工作。最好可做成PlugIn的方式。待调研

1. **采集内容梳理**

采集内容的丰富程度，直接决定了系统的“上限”。我梳理了与SQL相关的采集内容。未来对于采集内容，肯定还是需要后添加的。因此在采集程序设计上，要保留必要的扩展能力，这部分我随后会谈到。



1. **OS**

收集CPU、MEM、NET、IO等指标。这部分是通过Collectd完成的，其指标可根据需要随时扩展即可。其丰富程度应可以满足我们的需求。

1. **DB**

收集数据库实例级别的指标，包括BUFFER（缓冲区）、PROFILE（整体性能指标）、LOCK/LATCH（栓锁情况）、EVENT（事件情况）。这部分是需要采集程序采集。

1. **OBJ**

收集对象级别的指标，包括STAT（统计信息类）、ACTION（动作类，例如DML次数）、INFO（其他信息类，例如表大小）。

1. **SQL**

收集SQL级别的指标，包括TIME（执行时间类）、COST（成本类）、RESOURCE（资源类）、INFO（其他信息类，例如游标情况）。

1. **采集模块**

采集模块，完成了采集多类别（四个类别）、多维度（数十个维度）、多时间间隔（采集频率不同）的信息收集工作。根据数据源不同，其采集方式也不同，我简单梳理了基本采集结构（以MySQL为例）。



1. **Thread说明**

Main Thread，完成初始化、配置读取、子线程监控、输出日志等功能。

Collect DB Thread，完成DB级别信息收集。

Collect Obj Thread，完成OBJ级别信息收集。

Collect SQL Thread，完成SQL级别信息收集。

1. **Interface说明**

公共访问接口，提供对某类服务的访问能力。例如：

1. 通过SQL访问。例如OBJ级别访问，需要通过访问数据字典完成，调用此公共服务。
2. 通过MEM访问。例如通过show innodb status完成。DB级收集，调用此公共服务。
3. 通过日志访问。例如SQL级收集，调用此公共服务。

建议先抽象出公共服务部分。其中show innodb status，我已完成了。

1. **计算模块**

还没有详细规划，暂定为可配置选择算法，进行计算。待细化

1. **工作分工**
2. MySQL — 阿娇

* Collected mysql plugin调研
* 采集程序开发

OBJ：information\_schema

DB：show status + show innodb status

SLOW LOG：dir + prefix，sqlid，parse obj into sql info

1. Oracle — 玉龙

* Collected oracle plugin调研
* 采集程序开发

AWR、ASH

1. InfluxDB — 韩锋、王琦

* 调研函数

1. DEMO数据 — 韩锋

* Swing bench + oracle => data

1. **AI SQL产品设计**
2. **数据特征**

* 统计指标

极值（最大、最小值）、均值、中位数、众数

* 适用数据

OS.CPU，OS.MEM，OS.NET，OS.IO

OBJ.ACTION、OBJ.INFO(size)

DB.PROFILE

SQL.TIME，SQL.COST，SQL.RESOURCE

* 典型场景
* 中位数较均值更能反映上述指标，例如SQL时长、各种资源消耗等。
* 众数可反映出最常见的情况，也很有价值。如众数超过一个，则说明数据有明显分类。

1. **数据趋势**

* 统计指标

导数

* 适用数据

OS.CPU，OS.MEM，OS.NET，OS.IO

OBJ.ACTION、OBJ.INFO(size)

DB.PROFILE

SQL.TIME，SQL.COST，SQL.RESOURCE

* 典型场景

反应时间序列上数据差异。例如，执行时间导数持续增大，反应了执行恶化程度呈指数增长。

1. **数据离散度**

* 统计指标

分位距、标准差、标准分

* 适用数据

SQL.TIME，SQL.COST

* 典型场景
* 分位距，反映了去除噪声后的极值间的差距（最大-最小），可用来反映数据的差异性。传统的四分位距，粒度比较粗，可考虑95%的情况。
* 标准差，反映了典型值与均值查询的指标。标准差越小，反映数值离均值越近。
* 标准分，反映了相较于均值的位置。正的表示高于均值，负的表示低于均值。数据大小体现了数据与均值的距离。这个将数据形状保持不变，可做跨对象类比，更有意义。

1. **数据概率**

前提条件，假设SQL执行时长是符合泊松分布特征的。

* 统计指标

泊松分布的均值，期望和方差。

* 适用数据

SQL.TIME

* 典型场景
* 泊松分布期望(λ)，该值越大，执行特征越正态；该值越小越偏态。可用该指标进行类别划分。
* 泊松均值和方差，反映了执行时长差异化。

1. **数据回归（筛选关联因素）**

* 统计指标

回归系数（最小二乘法）

* 适用数据

SQL.TIME+XXX

* 典型场景
* 计算SQL执行时长与其他指标的回归系数(r)，反映出其线性拟合度。如果r=1，为完全线性相关；如r=0，则不存在相关性。

SQL受哪类资源的影响最大，可从多个因素中分析其受影响程度。

【RESOURCE】

评估系统资源影响SQL运行，例如可区分是IO敏感性、CPU敏感性等，用于优化方向的区分。

【OBJECT】

可评估哪个对象变化（大小、记录数、DML次数等）最影响SQL运行。