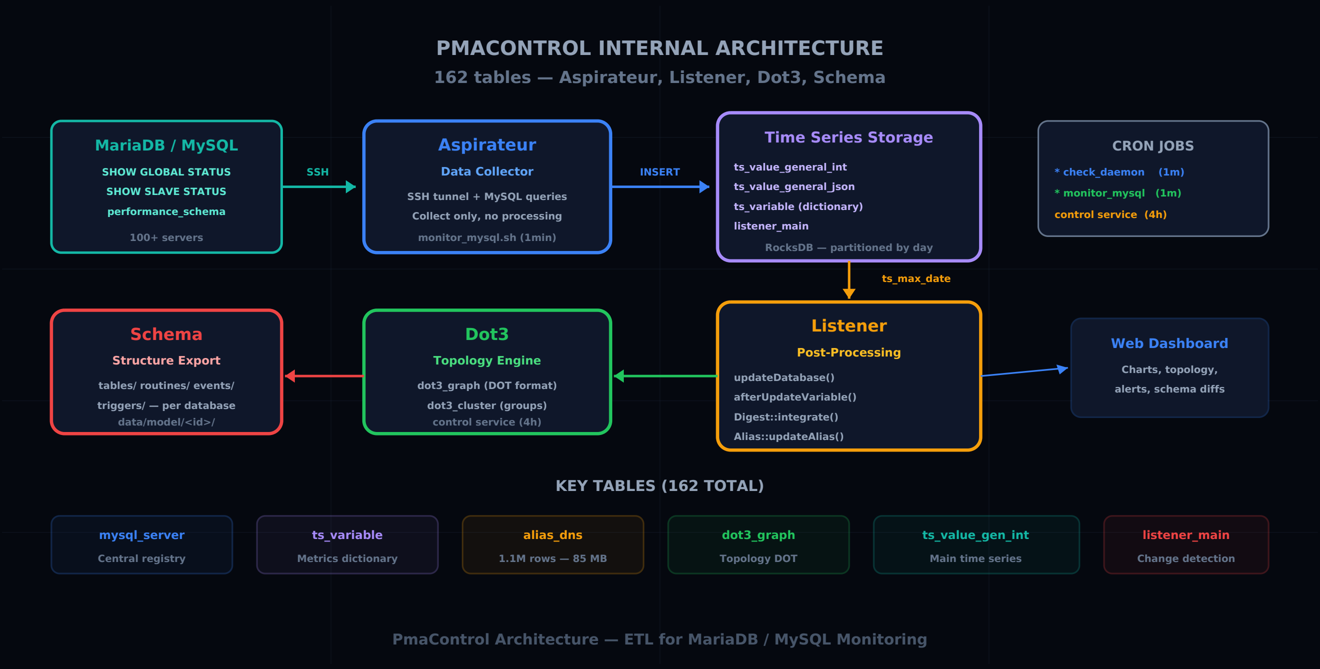
PmaControl [illegible]Aspirateur[illegible]Listener[illegible]Dot3 [illegible] 162 [illegible]

Aurélien LEQUOY · 2026[illegible]4[illegible]13[illegible]

PMACONTROL ARCHITECTURE AGENTS CRON MONITORING



162 [illegible]

PmaControl [illegible]**162** [illegible] MariaDB / MySQL [illegible]

[illegible]Aspirateur[illegible]Listener[illegible]Dot3[illegible]Schema[illegible] cron [illegible]

[illegible]

Aspirateur[illegible]

Aspirateur [illegible]

1. [illegible]**SSH**[illegible]**MySQL**[illegible]
2. `SHOW GLOBAL STATUS`[illegible]`SHOW GLOBAL VARIABLES`[illegible]`SHOW SLAVE STATUS`[illegible]`SHOW PROCESSLIST`[illegible] `performance_schema` [illegible]
3. [illegible] PmaControl [illegible] `ts_value_*` [illegible]

`ts_` [illegible]**time series**[illegible]

```
Aspirateur → SSH 隧道 → 本地 MySQL
           → SHOW GLOBAL STATUS
           → SHOW SLAVE STATUS
           → performance_schema 查询
           → INSERT INTO ts_value_general_int (...)
           → INSERT INTO ts_value_general_json (...)
```

Aspirateur 不做任何处理。它只负责采集和写入。这是一个基本设计原则：将采集与处理分离，使它们可以独立扩展。

Listener：后处理引擎

Listener 是 PmaControl 的 `st` 它监控时间序列表，并在新数据到达时触发操作。其机制依赖一张枢纽表：

`listener_main` 表包含：

列	角色
`ts_file`	数据源文件
`ts_max_date`	最后处理的时间戳
`ts_date_by_server`	每台服务器的最后时间戳

Listener 在一个持 循环中运行 每次迭代，它将 录的 `e_*` 表中最新的时间戳进行比较。如果检测到差异，说明 Aspirateur 已写入新数据 — Listener 随即触发后处理链：

```
Listener 循环:
  1. 检查 ts_max_date 与实际 max(timestamp)
  2. 如果有变化 → 触发管道:
     a. updateDatabase()       — 更新服务器元数据
     b. afterUpdateVariable()  — 触发条件规则
     c. Digest::integrate()    — 聚合 performance_schema 指标
     d. Alias::updateAlias()   — 刷新 DNS 别名
```

updateDatabase() 同步基本信息：服务器版本、复制状态、数据库大小、活跃连接数。

afterUpdateVariable() 是规 引擎。它将新值与配 的阈 进 比 要时超过 警0，例如创建果个 Warning 告警。

Digest::integrate() 处理 `erformance_schema` 数据。它聚合查询统计信息（执行时间、扫描行数、频率）并存储到 PmaControl 的 digest 表中。这为性能仪表盘提供数据

Alias::updateAlias() 维护 `lias_dns` 表，该 友好名称映射 实际 IP 地址，显示可读名称而非 IP 地址。

Dot3：实时拓扑

Dot3 是拓扑映射组件。它分析服务器之间的复制关系，并生成 DOT 格式（Graphviz）的有向图。

流程如下：

1. Dot3 读取每台服务器的复制元数据（主从关系、GTID、通道）
2. 它构建一个依赖图：谁是谁的主库，谁是谁的从库
3. 它生成带有集群（相关服务器组）的可视化表示

涉及的表：

- `dot3_graph`：完整的 DOT 图，可直接渲染
- `dot3_cluster`：服务器集群（一个集群 = 一个复制组）

Dot3 特别适用于检测损坏的拓扑：一个从库指向不存在的服务器、意外的环形复制回路，或一台本应属于某个集群却被隔离的服务器。

Schema：结构导出

Schema 组件导出每个受监控数据库的完整结构。对于每台服务器，它创建一个文件树：

```
data/model/<server_id>/databases/<db_name>/
├── schema/
│   └── tables/
│       ├── users.sql
│       ├── orders.sql
│       └── ...
├── routines/
│   ├── calculate_total.sql
│   └── ...
├── events/
│   ├── daily_cleanup.sql
│   └── ...
└── triggers/
    ├── before_insert_users.sql
    └── ...
```

每个文件对应一个 `CREATE TABLE`、`CREATE PROCEDURE`、`CREATE EVENT` 或 `CREATE TRIGGER`。这使得：

- 在 Git 中对结构进行版本控制（两次导出之间的 diff）
- 比较生产环境和预发布环境的结构
- 检测 Schema 漂移（在生产环境手动添加的索引、未经迁移修改的列）

Glial CLI

PmaControl [illegible] Glial [illegible]

```
./glial <controller> <action> [params]
```

[illegible]

```
# [illegible]
./glial agent check_daemon

# [illegible]
./glial control service

# [illegible] Schema
./glial schema export 42

# [illegible]
./glial dot3 generate
```

CLI [illegible] cron [illegible]

Cron [illegible]

[illegible] cron [illegible] PmaControl [illegible]

1. `./glial agent check_daemon` — [illegible]

[illegible] cron [illegible] — [illegible] cron [illegible]

```
* * * * * cd /srv/www/pmacontrol && ./glial agent check_daemon >> /tmp/pmacontrol_agent.log
2>&1
```

[illegible] 3 [illegible] Telegram [illegible]

2. `./glial control service` — [illegible] 4 [illegible]

[illegible] cron [illegible]

- [illegible]
- [illegible]
- [illegible] Dot3 [illegible]

- 同步服务器元数据
- 表间一致性检查

4 小时是数据新鲜度和负载之间的良好折中：这些操作开销较大，不需要实时执行。

```
0 */4 * * * cd /srv/www/pmacontrol && ./glial control service >> /tmp/pmacontrol_control.log 2>&1
```

3. `./monitor_mysql.sh` — 每分钟

这个脚本是 Aspirateur 的入口点。它触发一个完整的采集周期：

```
* * * * * cd /srv/www/pmacontrol && ./monitor_mysql.sh >> /tmp/pmacontrol_monitor.log 2>&1
```

该脚本处理并行化：如果你监控 200 台服务器，它不会逐一联系它们。它将工作分配到并行批次中，工作线程数可配置。

关键表

以下是理解或调试 PmaControl 需要了解的最重要的表：

`mysql_server`

[illegible] MariaDB / MySQL 服务器，还包括：

- **MariaDB / MySQL** 服务器（主要用例）
- 代[illegible]MaxScale、ProxySQL、HAProxy
- **VIP**（虚拟 IP）

`is_proxy` 和 `is_vip` 列区分类型：

is_proxy	is_vip	类型
0	0	普通 MariaDB / MySQL 服务器
1	0	代理（MaxScale、ProxySQL、HAProxy）
0	1	VIP（虚拟 IP）

```
-- 仅 MariaDB/MySQL 服务器
SELECT id, ip, port, name, display_name, id_environment
FROM mysql_server
```

```
WHERE is_deleted = 0 AND is_proxy = 0 AND is_vip = 0;

-- [illegible]MaxScale[illegible]ProxySQL[illegible]HAProxy[illegible]
SELECT id, ip, port, name, display_name
FROM mysql_server
WHERE is_deleted = 0 AND is_proxy = 1;

-- VIP
SELECT id, ip, port, name, display_name
FROM mysql_server
WHERE is_deleted = 0 AND is_vip = 1;
```

[illegible] VIP [illegible] MySQL [illegible] JOIN `timeout` [illegible] Dot3 [illegible] VIP → [illegible] → [illegible] → [illegible]

[illegible]

- **`maxscale_server`** / **`maxscale_server__mysql_server`** — MaxScale [illegible]
- **`proxysql_server`** — ProxySQL [illegible]
- **`haproxy_main`** / **`haproxy_main_input`** / **`haproxy_main_output`** / **`link__haproxy_main_output__mysql_server`** — HAProxy [illegible]
- **`vip_server`** — VIP [illegible]

`ts_variable`

[illegible] `Threads_connected` [illegible] `Innodb_buffer_pool_pages_data` [illegible]

```
SELECT id, name, source
FROM ts_variable
WHERE name LIKE 'Innodb%';
```

`ts_value_general_int`

[illegible] — [illegible] PmaControl [illegible]

```
SELECT server_id, variable_id, value, timestamp
FROM ts_value_general_int
WHERE server_id = 42
  AND variable_id = 107  -- Threads_connected
  AND timestamp > NOW() - INTERVAL 1 HOUR;
```

[illegible] `ALTER TABLE ... DROP PARTITION` [illegible]

`ts_value_general_json`

`SHOW PROCESSLIST` [illegible] `performance_schema` [illegible] `SHOW ENGINE INNODB STATUS` [illegible] JSON [illegible] `SHOW SLAVE STATUS` [illegible] `ts_value_slave_*` [illegible]

`alias_dns`

DNS [illegible] — 110 [illegible] 85 MB [illegible] IP [illegible]

```
SELECT ip, alias, source, updated_at
FROM alias_dns
WHERE ip = '10.0.1.42';
```

`dot3_graph` [illegible] **`dot3_cluster`**

`dot3_graph` [illegible] DOT [illegible] `dot3_cluster` [illegible]

[illegible]

[illegible]

[illegible]	[illegible]	[illegible]
1	**CRON** `monitor_mysql.sh` [illegible]	[illegible] Aspirateur
2	**ASPIRATEUR**	SSH [illegible] → MySQL → `SHOW GLOBAL STATUS`
		[illegible] `ts_value_general_int` / `ts_value_general_json`
3	**LISTENER** [illegible] `ts_max_date` [illegible]	`updateDatabase()` — [illegible]
		`afterUpdateVariable()` — [illegible]
		`Digest::integrate()` — [illegible] `performance_schema`
		`Alias::updateAlias()` — [illegible] `alias_dns`
4	**DOT3** [illegible] 3 [illegible]	[illegible]
5	**CRON** `control service` [illegible] 4 [illegible]	[illegible]
6	**WEB** [illegible]	[illegible] → [illegible]

[illegible]

对于典型的 100 台 MariaDB / MySQL 服务器部署：

- **PmaControl** 数据库：约 15 GB（数据在 `ts_value_*` 表为主）
- **CPU**：2-4 核即可（Listener 消耗最大）
- **RAM**：最低 4 GB，建议 8 GB（用于 PmaControl 数据库自身的 buffer pool）
- 磁盘：必须使用 SSD — 时间序列表产生大量随机 I/O

`ts_value_*` 表使用 **RocksDB**（通过 MyRocks）作为存储引擎：更好的压缩、更好的顺序写入性能，以及原生的按天分区。

调试指南

当出现问题时，以下是检查清单：

1. 代理在运行？ `agent check_daemon` — 如果代理死了，它管理的服务器将不再采集数据
2. **Listener** 是否在 `Listener_main` 中推进 `ts_max_date` — 如果不再推进，说明 Listener 卡住了
3. **cron** 任务 `/tmp/pmacontrol_*.log` 中是否有错误
4. **SSH** 连接是否正常？使用配置的密钥和 `<port> <user>@<host>`
5. **PmaControl** 数据库：PmaControl 数据库自身的磁盘空间、锁、慢查询

总结

PmaControl 的架构遵循经典的 ETL（提取-转换-加载）模型，适配于监控场景：

- 提取（**Extract**）：Spirateur 只采集不处理
- 转换（**Transform**）：Listener 应用规则并聚合
- 加载（**Load**）：直接读取转换后的数据

162 张表不是偶然的复杂性 — 它们反映了从每台 MariaDB / MySQL 服务器采集的数据的丰富性。理解这套架构对于任何想要调试采集问题、为 PmaControl 扩展新指标类型或优化监控系统自身性能的