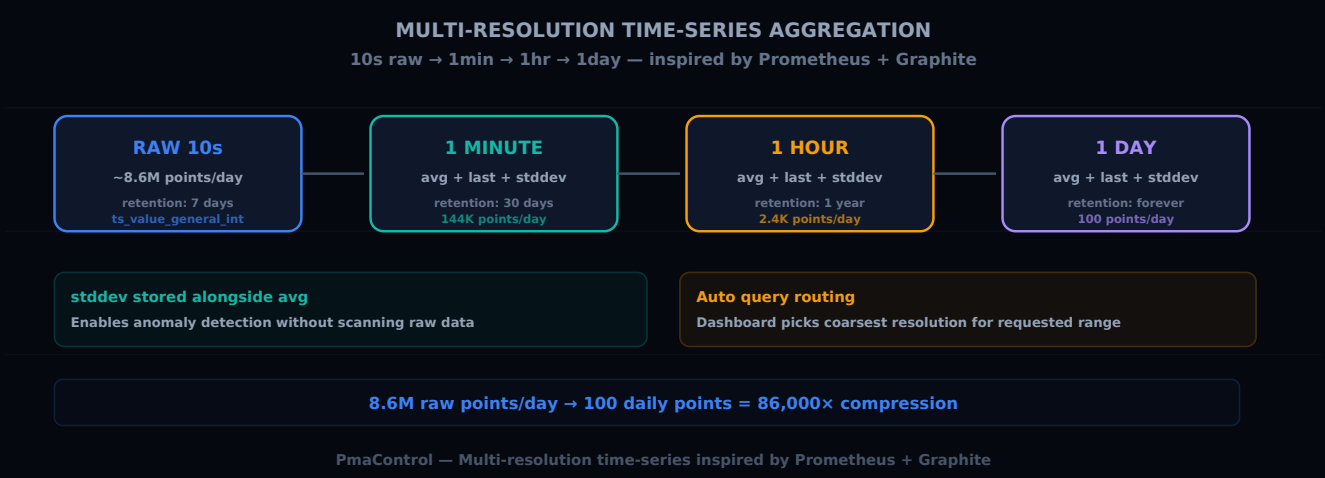
تجميع السلاسل الزمنية: من ملايين النقاط الأولية إلى الاستعلامات السريعة

Aurélien LEQUOY · 21 مارس 2026

PMACONTROL TIME-SERIES AGGREGATION MONITORING ARCHITECTURE



الحجم الإجمالي

يجمع PmaControl المقاييس من كل مثيل MariaDB / MySQL خاضع للإشراف كل **10 ثوانٍ** يمثل هذا لكل خادم ما يلي:

- 6 نقاط في الدقيقة
- 360 نقطة في الساعة
- 8,640 نقطة يوميا
- **60,480 نقطة في الأسبوع**

مع 100 خادم و50 مقياساً لكل خادم:

```
100 serveurs × 50 métriques × 8 640 points/jour = 43 200 000 points/jour
```

43 مليون نقطة يوميا. 302 مليون في الأسبوع. أكثر من مليار شهريا.

يعد تخزين كل هذا بدقة أولية (10 ثوانٍ) إلى أجل غير مسمى أمراً ممكناً من الناحية الفنية ولكنه عديم الفائدة من الناحية العملية. لا أحد ينظر إلى مخطط بيانات مدته 10 ثوانٍ منذ 6 أشهر. وتكون الاستعلامات التي تقوم بمسح ملايين الصفوف لعرض مخطط مدته سنة واحدة

بطيئة ومكلفة.

الإلهام: Prometheus وGraphite

المشكلة ليست جديدة. نظامان حلاها بأناقة:

- **Prometheus** مع **قواعد التسجيل**: استعلامات PromQL المحسوبة مسبقاً والتي تجمع البيانات الأولية في مقاييس مشتقة على فترات زمنية منتظمة
- **Graphite** بتنسيق **Whisper**: نظام احتفاظ متعدد الدقة حيث يتم تجميع البيانات تلقائياً مع مرور الوقت

PmaControl تستمد الإلهام من كل النهجين لتصميم نظام التجميع الخاص بها.

المخطط متعدد الدقة

أربعة مستويات من القرار:

المستوى	الفاصل	الاحتفاظ	الحجم المقدر (100 خادم)
الخام	10 ثواني	7 أيام	302 مليون نقطة/أسبوع
1 دقيقة	1 دقيقة	30 يوما	216 مليون نقطة/شهر
1 ساعة	1 ساعة	1 سنة	43.8 مليون نقطة/سنة
يوم واحد	يوم واحد	غير محدد	1.8 مليون نقطة/سنة

إجمالي الحجم المخزن في أي وقت (مع 100 خادم):

```
Raw (7 jours) :    302M points
1min (30 jours) :  216M points
1hr (1 an) :        43.8M points
1day (tout) :        ~2M points
Total :            ~564M points
```

وبدون التجميع، فإن الاحتفاظ بالبيانات الأولية لمدة عام واحد سيمثل **15.8 مليار نقطة**. التجميع يقلل من التخزين بعامل 28.

ما يتم تخزينه في كل مستوى إجمالي

لكل نقطة مجمعة، يتم تخزين ثلاث قيم:

```
CREATE TABLE ts_aggregated_1min (
    server_id     INT,
    metric_id     INT,
    timestamp     DATETIME,
    last_value    DOUBLE,    -- dernière valeur de l'intervalle
    avg_value     DOUBLE,    -- moyenne de l'intervalle
    stddev_value  DOUBLE,    -- écart-type de l'intervalle
    PRIMARY KEY (server_id, metric_id, timestamp)
);
```

لماذا `last_value`؟

بالنسبة لمقاييس النوع المضاد (عدد الطلبات والبايتات المرسلة)، غالبًا ما تكون القيمة الأخيرة للفاصل الزمني أكثر صلة من المتوسط. وهو يمثل أحدث دولة.

لماذا `avg_value`؟

بالنسبة للمقاييس من نوع المقياس (استخدام وحدة المعالجة المركزية، والذاكرة، والسلاسل النشطة)، فإن المتوسط هو التمثيل الأكثر دقة للسلوك خلال الفاصل الزمني.

لماذا `stddev_value`؟ البصيرة الرئيسية

هذا هو الابتكار الرئيسي لهذا التصميم. **تخزين الانحراف المعياري إلى جانب المتوسط يسمح باكتشاف الحالات الشاذة بدون البيانات الأولية.**

خذ بعين الاعتبار ساعتين بنفس متوسط وحدة المعالجة المركزية وهو 45%:

- **الساعة أ**: استقرار وحدة المعالجة المركزية بين 42% و48%. `avg=45%, stddev=2%`
- **الساعة ب**: تذبذب وحدة المعالجة المركزية بين 5% و85%. `avg=45%, stddev=28%`

بدون stddev، لا يمكن تمييز هاتين المرتين في البيانات المجمعة. مع stddev، يمكن تحديد الوقت B على الفور على أنه غير طبيعي.

يتيح لك هذا إنشاء تنبيهات بناءً على البيانات القياسية التاريخية:

```
SI stddev_actuel > 3 × stddev_moyen_30_derniers_jours
ALORS alerte : comportement anormal détecté
```

عملية التجميع

يعمل التجميع بشكل متتالي، مدفوعاً بوظيفة cron:

الخطوة 1: خام → دقيقة واحدة

في كل دقيقة، يقرأ العامل آخر 6 نقاط خام لكل زوج (خادم، متري) ويحسب:

```
INSERT INTO ts_aggregated_1min (server_id, metric_id, timestamp, last_value, avg_value,
stddev_value)
SELECT
    server_id,
    metric_id,
    DATE_FORMAT(timestamp, '%Y-%m-%d %H:%i:00') AS minute,
    -- last_value : sous-requête pour le dernier point
    (SELECT value FROM ts_raw r2
     WHERE r2.server_id = ts_raw.server_id
       AND r2.metric_id = ts_raw.metric_id
       AND r2.timestamp >= DATE_FORMAT(ts_raw.timestamp, '%Y-%m-%d %H:%i:00')
       AND r2.timestamp < DATE_FORMAT(ts_raw.timestamp, '%Y-%m-%d %H:%i:00') + INTERVAL 1
MINUTE
     ORDER BY r2.timestamp DESC LIMIT 1),
    AVG(value),
    STDDEV(value)
FROM ts_raw
WHERE timestamp >= NOW() - INTERVAL 1 MINUTE
GROUP BY server_id, metric_id, minute;
```

الخطوة 2: دقيقة واحدة → ساعة واحدة

في كل ساعة، يقوم العامل بتجميع 60 نقطة مدتها دقيقة واحدة في نقطة مدتها ساعة واحدة. يستخدم حساب stddev المدمج صيغة التباين المجمعة:

```
σ_combiné = sqrt( mean(σ²_i) + var(μ_i) )
```

حيث `σ_i` هي القيم القياسية للفترات الفرعية و `μ_i` وسيلتها. هذه الصيغة دقيقة رياضياً ولا تحتاج إلى بيانات أولية.

الخطوة 3: ساعة واحدة ← يوم واحد

بنفس المبدأ، مرة واحدة يومياً، تصبح 24 نقطة مدتها ساعة واحدة نقطة واحدة مدتها يوم واحد.

الخطوة 4: مسح البيانات القديمة

بعد كل تجميع، يتم حذف البيانات التي تتجاوز الاحتفاظ:

```
DELETE FROM ts_raw WHERE timestamp < NOW() - INTERVAL 7 DAY;
DELETE FROM ts_aggregated_1min WHERE timestamp < NOW() - INTERVAL 30 DAY;
DELETE FROM ts_aggregated_1hr WHERE timestamp < NOW() - INTERVAL 1 YEAR;
-- ts_aggregated_1day : jamais purgé
```

طلب التوجيه

عندما تعرض لوحة المعلومات PmaControl مخططًا، يجب عليها اختيار الدقة المناسبة. المبدأ بسيط: **استخدم الدقة الأكثر خشونة التي تغطي النطاق المطلوب**.

```
function selectResolution(int $timeRangeSeconds): string {
    if ($timeRangeSeconds <= 3600) {       // <= 1 heure
        return 'ts_raw';                   // 10s resolution
    } elseif ($timeRangeSeconds <= 86400 * 2) {  // <= 2 jours
        return 'ts_aggregated_1min';       // 1min resolution
    } elseif ($timeRangeSeconds <= 86400 * 90) { // <= 90 jours
        return 'ts_aggregated_1hr';        // 1hr resolution
    } else {
        return 'ts_aggregated_1day';       // 1day resolution
    }
}
```

النتيجة: يقوم الرسم البياني لمدة عام واحد بتحميل **365 نقطة** فقط (دقة يوم واحد) بدلاً من 3.1 مليون (دقة 10 ثوانٍ). ينتقل الطلب من عدة ثوانٍ إلى بضعة أجزاء من الثانية.

التأثير على الاستعلامات

النطاق المطلوب	القرار	النقاط المحملة	وقت الاستعلام
1 ساعة	10ث (خام)	360	<10 ملي ثانية
24 ساعة	1 دقيقة	1,440	<20 ملي ثانية
30 يوما	1 ساعة	720	<15 ملي ثانية
1 سنة	يوم واحد	365	<10 ملي ثانية

تصبح أوقات الاستعلام **مستقلة عن النطاق الزمني**. الرسم البياني لمدة عام واحد هو بنفس سرعة الرسم البياني لمدة ساعة واحدة.

الكشف عن الشذوذ باستخدام stddev المخزن

بفضل stddev المحسوب مسبقاً، يمكن لـ PmaControl اكتشاف الحالات الشاذة في البيانات المجمعة دون الرجوع إلى البيانات الأولية:

1. **حساب خط الأساس**: المتوسط والمستوى القياسي للتطوير القياسي على مدار آخر 30 يوماً لكل مقياس
2. **المقارنة**: تتم مقارنة المعيار القياسي للوقت الحالي بخط الأساس
3. **تنبيه**: إذا تجاوز المعيار القياسي 3 أضعاف خط الأساس، فهذا سلوك غير طبيعي

مثال ملموس:

- تشغيل المواضيع الأساسية: `avg_stddev = 2.1, stddev_stddev = 0.8`
- الوقت الحالي: `stddev = 14.3`
- النتيجة: `(14.3 - 2.1) / 0.8 = 15.25` سيجما - **شذوذ معين**

تكتشف هذه الآلية الحالات الشاذة التي قد يفوتها المتوسط البسيط: الخادم الذي تتأرجح وحدة المعالجة المركزية الخاصة به بعنف ولكنها تعود دائماً إلى المتوسط الصحيح.

الخلاصة

يعد التجميع متعدد الدقة وهو المفتاح لإدارة بيانات السلاسل الزمنية على نطاق واسع. يعد تخزين stddev جنباً إلى جنب مع المتوسط خياراً تصميمياً غير عادي ولكنه قوي: فهو يحافظ على معلومات حول التباين، مما يسمح باكتشاف الحالات الشاذة حتى في البيانات المجمعة.

باستخدام هذا النظام، يمكن لـ PmaControl مراقبة أكثر من 100 خادم MariaDB / MySQL على مدار عام كامل مع ضمان استعلامات لوحة المعلومات في أقل من 20 ملي ثانية.