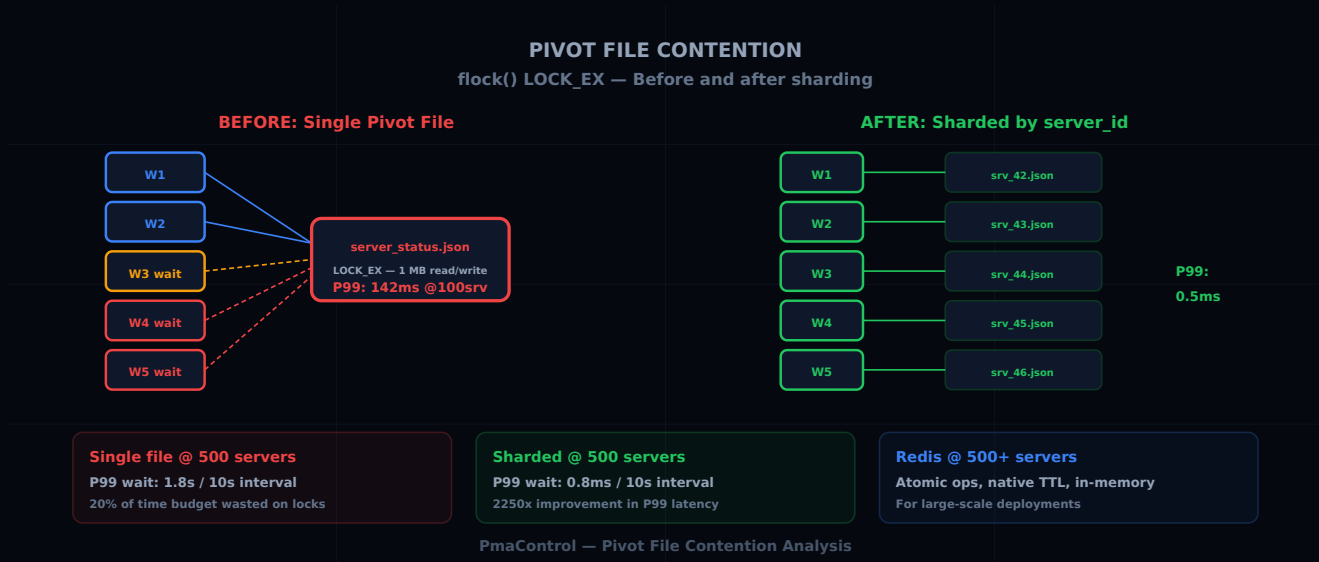


██████████ PmaControl ██████████

Aurélien LEQUOY · 2026-03-19

PMACONTROL PHP CONCURRENCY PERFORMANCE ARCHITECTURE



PHP ██████████

PmaControl PHP ██████████ MariaDB / MySQL ██████████ Web ██████████

PHP ██████████ Go goroutine Java ██████████ PHP pivot files ██████████ Web ██████████ PmaControl

StorageFile ██████████ SharedMemory flock() LOCK_EX ██████████

LOCK_EX ██████████

flock() LOCK_EX ██████████

LOCK_EX flock() LOCK_EX ██████████

██████████

```
// Test de concurrence sur flock()
// Lancé avec 50 processus simultanés
$fp = fopen('/tmp/pivot_test.json', 'c+');
if (flock($fp, LOCK_EX)) {
    $data = json_decode(fread($fp, filesize('/tmp/pivot_test.json')), true);
}
```

```

    $data['counter'] = ($data['counter'] ?? 0) + 1;
    ftruncate($fp, 0);
    rewind($fp);
    fwrite($fp, json_encode($data));
    flock($fp, LOCK_UN);
}
fclose($fp);

```

50 10,000 500,000

PmaControl

```

/var/lib/pmacontrol/pivot/
server_status.json ←
server_42_metrics.json ← 42
server_43_metrics.json
...

```

server_status.json worker

1. server_status.json LOCK_EX
2. JSON
- 3.
- 4.
- 5.

10 10 100 worker

StorageFile flock()

```

$start = microtime(true);
flock($fp, LOCK_EX);
$wait = microtime(true) - $start;

```


worker

	flock()	P99
100	0.1 ms	0.5 ms
200	0.1 ms	0.6 ms
500	0.2 ms	0.8 ms

worker

N

```
// Avant : un seul fichier
$status = json_decode(file_get_contents('pivot/server_status.json'), true);
```

```
// Après : N fichiers
$status = [];
foreach (glob('pivot/status/server_*.json') as $file) {
    $serverId = extractServerId($file);
    $status[$serverId] = json_decode(file_get_contents($file), true);
}
```

rename() flock()

Redis memcached

500

- **I/O** Linux
- **Inode** 500 = 500 inode
- **TTL**

StorageFile Redis memcached

```
// Interface abstraite
interface StorageBackend {
    public function get(string $key): ?array;
    public function set(string $key, array $data, int $ttl = 0): void;
}

// Implémentation fichier (actuelle)
class StorageFile implements StorageBackend { ... }

// Implémentation Redis (future)
class StorageRedis implements StorageBackend { ... }
```

Redis TTL

Redis

PmaControl PHP Redis RabbitMQ Debian

Redis StorageFile StorageRedis

1-50		StorageFile
50-200	server_id	StorageFile
200-500	+ SSD	StorageFile
500+	Redis / memcached	StorageRedis

flock() LOCK_EX worker 100

server_id worker Redis

PHP