Thử nghiệm

Thiết lập:

Cho phép tối đa 2 cloud vm chạy 1 lúc (1 monitor, 1 scale tối đa).

Dữ liệu:

Sinh kịch bản số lượng request theo thời gian ra file, chu kỳ 8h, jmeter bắn request theo file này.

Dữ liệu train có độ di khoảng 4 ngày.

Mô hình:

Mỗi điểm dữ liệu lấy trung bình trong 4 phút, cứ 4 phút lấy được điểm mới rồi dự báo điểm của 4 phút sau và kiểm tra điều kiện scale.

window size = 4 điểm, tức là 4 điểm dữ liệu của 4 vòng lặp.

update mô hình lại sau 120 điểm, = 8h.

Điều kiện scale:

- Scale thêm: đự đoán 4 phút sau > 0.55 thì scale

- Scale bớt:

1 máy cần ít hơn 0.55, 2 máy cần ít hơn khoảng 0.27, tức là khi đã scale và đang có 2 máy, nếu monitor được điểm < 0.27 \* k thì scale bớt (k đang để 0.8).

2 điều kiện:

+ đự đoán 4 phút sau cũng < 0.27\*k

+ trung bình m điểm đã có < 0.27\*k, ở đây m = 3.

Kịch bản:

- Cho jmeter bắn request theo kịch bản trên.

- Trong mỗi chu kỳ 8h, em bắn thêm request trong khoảng 40 phút cuối.

Bảng kết quả thử nghiệm: Đường đỏ: dữ liệu thật, Đường xanh nét đứt: dữ liệu dự đoán, Hình tròn đen: thời điểm tạo thêm máy ảo, Hình tam giác đen: thời điểm xóa bớt máy ảo.

mae [0.038949210602516667, 0.033705406829620838, 0.030915710335116666, 0.057588497869451617]

rmses [0.0688761467834617, 0.06672340276363664, 0.05251998207602928, 0.09009188152144826]

mapes [0.11353999934093509, 0.098319513722659208, 0.089721215974831861, 0.17458590586608771]

Mô hình chạy được 3 lần update

Update 1:

Các điểm phát hiện scale up:

|  |  |  |
| --- | --- | --- |
|  | Giá trị thật | Giá trị dự đoán |
| 1 | 0.564515617227 | 0.552029 |
| 2 | 0.631060618395 | 0.594336 |
| 3 | 0.573064421474 | 0.576719 |

Update 2:

0.645878836204

0.659165

0.560714454337

0.553439

Update 3:

0.68267326644

0.56223

0.593808467451

0.570563