数据模型

示例数据

timestamp	cpu_1m_avg	free_mem	temperature	location_id	dev_ty
2017-01-01 01:02:00	80	500MB	72	335	field
2017-01-01 01:02:23	90	400MB	64	335	roof
2017-01-01 01:02:30	120	0MB	56	77	roof
2017-01-01 01:03:12	80	500MB	72	335	field
2017-01-01 01:03:35	95	350MB	64	335	roof
2017-01-01 01:03:42	100	100MB	56	77	roof

窄表模型

- 每个指标作为一个单独的实体类来表示(例：cpu_lm_avg和free_mem表示成两种不同的实体类)
- 指标存储用"时间","值"的键值对来存储
- 将元数据值表示为与该指标/标签集组合相关联的 “标签集”

```
1. {name:  "cpu_1m_avg",  device_id: abc123,  location_id: 335,  dev_type
ield}
2. {name:  "cpu_1m_avg",  device_id: def456,  location_id: 335,  dev_type
oof}
3. {name:  "cpu_1m_avg",  device_id: ghi789,  location_id:  77,  dev_type
oof}
4. {name:    "free_mem",  device_id: abc123,  location_id: 335,  dev_type
ield}
5. {name:    "free_mem",  device_id: def456,  location_id: 335,  dev_type
oof}
6. {name:    "free_mem",  device_id: ghi789,  location_id:  77,  dev_type
oof}
7. {name: "temperature",  device_id: abc123,  location_id: 335,  dev_type
ield}
8. {name: "temperature",  device_id: def456,  location_id: 335,  dev_type
oof}
```

```
9. {name: "temperature", device_id: ghi789, location_id: 77, dev_type
oof}
```

优点：如果单独收集每个指标，那么窄表模型就有意义。它允许您通过添加新标记来添加新指标，而无需进行正式的架构更改。
缺点：如果要收集具有相同时间戳的很多度量标准，则窄表模型就不具有高效性，因为它需要为每个度量标准写入时间戳。这最终会导致更高的存储和获取要求。

宽表模型

timestamp	cpu_1m_avg	free_mem	temperature	location_id	dev_ty
2017-01-01 01:02:00	80	500MB	72	335	field
2017-01-01 01:02:23	90	400MB	64	335	roof
2017-01-01 01:02:30	120	0MB	56	77	roof
2017-01-01 01:03:12	80	500MB	72	335	field
2017-01-01 01:03:35	95	350MB	64	335	roof
2017-01-01 01:03:42	100	100MB	56	77	roof