

MongoDB GPS系统 架构设计案例分享

{

Name: “冯璟”,

Company: “Vastdata”,

mail :“fengjing@vastdata.com.cn”

Mobile: 15913113083

}



Agenda

1 背景与需求

2 Why MongoDB

3 How

背景与需求

客户为什么会有替换数据库的想法？

- 当前需求的GPS数据量巨大，无法存储
- 现架构的数据分散，难以分析挖掘
- 现系统无法满足处理需求，效率低
- 维护成本高，备份恢复困难
- 各项目组对GPS数据的诉求各有不同，原生DB对GEO数据不支持
- 数据使用问题多，影响衍生功能发展

背景与需求

要达到的目标

- 新环境采用分布式，降低成本
- GPS数据集中化管理
- 性能具有可扩展性
- 维护成本低、省事
- 能满足当前20多个应用场景，同时应用扩展性好

数据量预估：

业务系统	记录/天 (千万)	空间/天 (GB)	记录/年 (千万)	空间/年 (GB)
	10	20	3650	7300
	3	6	1095	1590
	4	7	1460	1855
	15	30	5475	7950
合计	32	63	11680	18695

Agenda

1 背景与需求

2 Why MongoDB

3 How

Why MongoDB



MongoDB Vs. PostGIS/PostgreSQL



	MongoDB	PostGIS
存储方式	GEOjson	行列式 (spatial tables and PostGIS administration)
平面几何运算支持能力 Geometry / Flat Shape	支持点、线的基础范围运算 (box/circle/polygon)	支持大量、多样的平面运算方式
地理运算支持能力 Geography / Spherical Shape	由于有2dsphere索引的支持，对于地理（球面）运算更简易，精确	原生支持点线的记录方式
查询性能	查询处理能力更强	
总结： 1.对于大部分的GPS应用，MongoDB是更快捷更好的选择 2.对于复杂的、高要求的GPS信息处理需求（如多图层地图运算），PostGIS更合适		

Why MongoDB



A Geometric bounding box query in its native planar environment, and warped onto a spherical system.

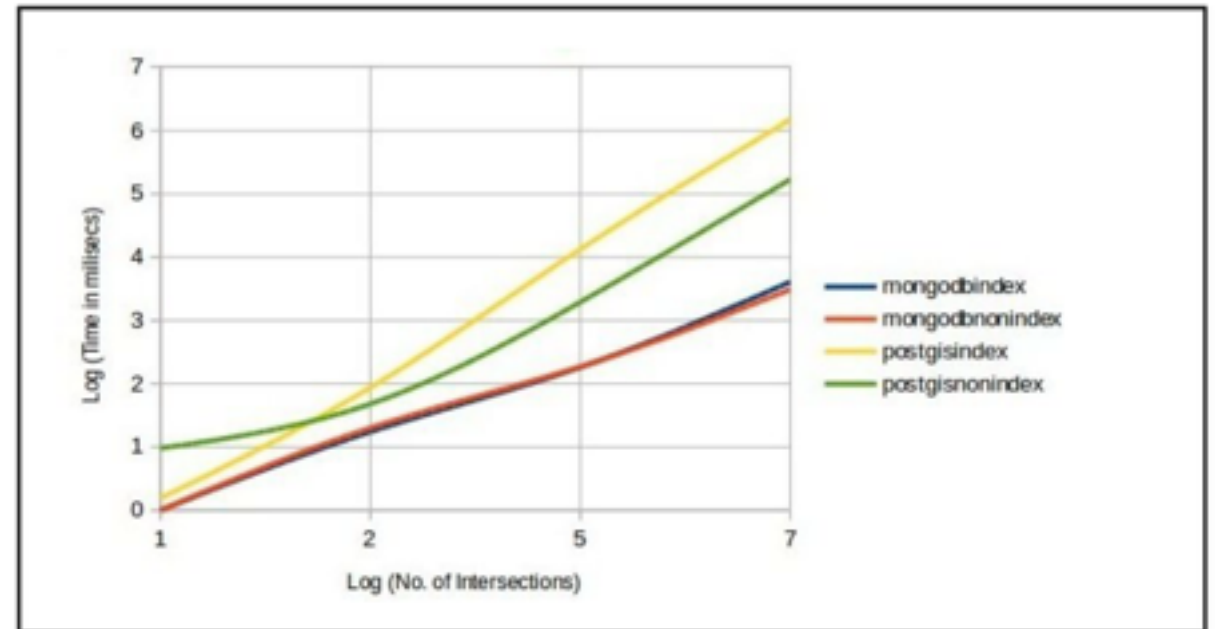
Why MongoDB

MongoDB Vs. PostGIS/PostgreSQL Performance 1

Line Intersection

Table 1. Time vs Intersection Table

No. of Intersections	MongoDB		PostGIS	
	Non-Index	Index	Non-Index	Index
21	1	1	1.546	9.721
1875	18	20	88.695	48.461
195691	185	190	13364.596	1963.123
< (1000*1000)	4093	3140	>1500000	172048



Why MongoDB

MongoDB Vs. PostGIS/PostgreSQL Performance 2

Table 1. Points within a polygon vs Time for all implementation.

No of Points Within	Mongo Non-Index	Mongo Index	PostGIS Non-Index	PostGIS Index
5002	6.75	7.75	52.95	54.97
5471	12	12	85.422	77.508
17004	20.8	24	142.53	158.76
45999	27.66	27.33	160.02	168.23
46008	50	51	675.33	631.556
122006	51.667	53.33	1339.23	1110.42
122016	96	99	2783.66	2609.47
204024	99.667	106	3991.564	3121.11223
298.23	125	112	-	-

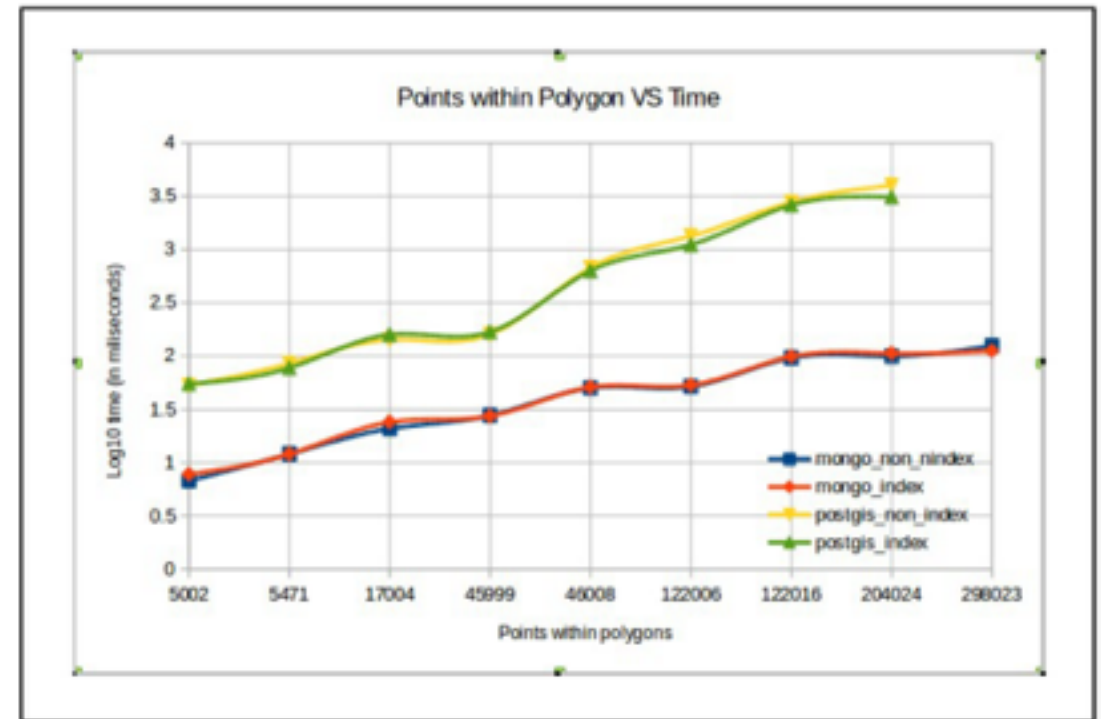


Figure 1. Graph of Point within Vs Time

Agenda

1 背景与需求

2 Why MongoDB

3 How

HOW

但...

在上线前，有几个问题要先解决

- 我们该用怎样的架构才能满足??
- 怎样去满足各个不同项目组的业务需求差异??
- 我们的历史数据怎么办?? 怎么处理??

HOW

对于物理架构，我们是这样考虑的...

- ◆ 为了性能的扩展，分片是必须的
- ◆ 为了业务连续性，复制集是必须的
- ◆ 初期shard我们计划做3到4个
- ◆ 每个片IOPS在6k以上，cache \geq 128G
- ◆ 总可用空间30T

可是...

要是这些资源不能都申请下来呢

HOW

看看到手上有什么的...

应用的写IOPS需求= 1W/S

- ◆ 服务器IOPS= (11+1 raid5, 7.2k sas) =11*80=880
- ◆ 总可用空间30T
- ◆ 能组成的分片 ——2个, PSA架构, cache128G

<https://docs.mongodb.com/v3.6/reference/read-concern-majority/#disable-read-concern-majority>

HOW

所以...

- 对于基础架构

- p 2 PSA shards

- p 通过业务ID作为片键，以哈希的方式入库，以达到数据尽量打散、写入性能高的目的

- p 数据入库前会先做整合处理，通过bulk insert降低写入请求次数

- 对于应用

- p 根据业务应用系统，拆分成不同的数据库，降低后期解耦难度的

- p 将collection分为配置类与日志信息（GPS）两大类，按业务类型定义规则

- p 日志信息类的每天一个collection

HOW



THANKS

北京海量数据技术股份有限公司

内部资料 严禁外传