

SSDB 入门基础

文档信息	2
SSDB 简介	3
PHP 使用示例	3
SSDB 数据类型	3
KV	3
Hashmap	3
Zset	4
安装和运行	4
使用 SSDB 命令行客户端(ssdb-cli)	6
运行 ssdb-cli	6
进行 KV 操作	6
进行 hashmap 操作	6
进行 zset 操作	7
SSDB 应用程序开发	7
示例 1: 在线书店	8
保存书籍信息	8
建立排序索引	8
分页浏览	8
相关链接	8

文档信息

作者: ideawu

网站: <http://www.ideawu.net/>

日期: 2013-09-29

更新: 2013-10-13

SSDB 简介

SSDB 是一个 C++ 语言开发的高性能开源 NoSQL 数据库服务器, 支持 Key-value, Key-hashmap, Key-zset(sorted set) 等数据结构, 十分适合存储数亿条级别的列表, 排序表等集合数据, 是 Redis 的替代和增强方案.

SSDB 具有和 Redis 相似的 API, 支持的客户端包括: PHP, C++, Python, Java, Lua, Ruby, Nodejs, etc.

项目主页: <https://github.com/ideawu/ssdb>

PHP 使用示例

```
<?php
require_once('SSDB.php');
$ssdb = new SimpleSSDB('127.0.0.1', 8888);
$res = $ssdb->set('key', '123');
$res = $ssdb->get('key');
echo $res; // output: 123
```

SSDB 数据类型

SSDB 支持三种数据类型, 分别是 KV(key-value), Hashmap(map), Zset(sorted set). 三种数据类型处于不同的命名区间, 所以不同类型的数据可以起相同的名字, 但不建议这么做.

KV

KV 数据类型支持基本的 set(), get(), del(), incr() 几种操作. KV 数据类型主要用于存储离散的, 之间没有关系(或者关系被忽略)的大数据, 如图片文件, 大段文本等. 一般 KV 类型都可以被 Hashmap 替代, 但 KV 会比 Hashmap 性能高一些.

key	value
k1	vc
k2	vx
k3	vy

Hashmap

Hashmap 类型和 KV 功能相似, 可用于存储大体积的数据, 但不同的数据项在业务上处于某个集合. 并且, Hashmap 维护了一个集合大小的计数.

Hashmap 中的数据项是 Key-value 的键值对, 并且按 Key 的字节数组顺序进行排序.

如果数据需要经常被遍历, 则应该使用 Hashmap 来替代 KV. 对于只添加, 不更新和删除的有排序需求的数据集合, 可以用 Hashmap 来存储而不需要使用 Zset, 因为 Hashmap 会比 Zset 性能高一些.

name	key	value
h	k1	vb
h	k2	va
h	k3	vc

Key 是按字节顺序排序的.

Zset

Zset 是一种根据数据项的权重(score, 整数值)进行排序的集合, Zset 集合中的数据项是唯一, 不可重复的. Zset 可以理解为关系数据中只有 ID 主键和整数 score 字段一共两个字段的表.

因为 Zset 的排序特性, 所以可用来存储排序列表, 如商品按价格的排序列表, 商品按上架日期的排序列表, 等等. 每一个排序列表对应一个 Zset 集合.

Zset 不能用来存储大体积的数据, 因为它是一种"索引"数据类型, 被索引的东西(集合中的数据项)只能是 200 字节以内的字节数组(包括字符串).

name	key	score
z	k2	0
z	k3	1
z	k1	2

Key 是按 score(64 位整数)的大小排序的.

安装和运行

SSDB 的建议安装方式是源码编译安装, 建议运行环境是主流 Linux 发行版. 远程 SSH 登录你的服务器, 然后用下面的命令下载, 编译, 安装和运行:

```
$ wget --no-check-certificate https://github.com/ideawu/ssdb/archive/master.zip
$ unzip master
$ cd ssdb-master
$ make
$ #optional, install ssdb in /usr/local/ssdb
$ sudo make install

# start master
```

```
$ ./ssdb-server ssdb.conf

# or start as daemon
$ ./ssdb-server -d ssdb.conf

# ssdb command line
$ ./tools/ssdb-cli -p 8888

# stop ssdb-server
$ kill `cat ./var/ssdb.pid`
```

SSDB 默认安装在 `/usr/local/ssdb` 目录下。ssdb-server 是服务器的程序，ssdb-cli 是命令行客户端。

在使用自带的 `ssdb.conf` 配置文件时，SSDB 生成的日志文件按体积进行分割，仅此而已。所以，你需要编写自己的 `crontab` 进行日志压缩和定期清理。

如果出现服务器掉电，`kernel panic` 等系统故障，在系统重新启动之后，你需要手动删除 `ssdb` 的 PID 文件 `ssdb.pid`，然后才能启动 `ssdb-server`。

另外，你可以参考下面的做法，在系统启动和关机时，启动和关闭 `ssdb-server`：

```
#!/bin/sh

case "$1" in
    'start')
        /usr/local/ssdb/ssdb-server -d /usr/local/ssdb/
ssdb.conf
        echo "ssdb started."
        ;;
    'stop')
        kill `cat /usr/local/ssdb/var/ssdb.pid`
        echo "ssdb stopped."
        ;;
    'restart')
        kill `cat /usr/local/ssdb/var/ssdb.pid`
        echo "ssdb stopped."
        sleep 0.5
        /usr/local/ssdb/ssdb-server -d /usr/local/ssdb/
ssdb.conf
        echo "ssdb started."
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac
```

把文件保存为 `/etc/init.d/ssdb.sh`(需要 `root` 权限)，然后执行：

```
chmod ugo+x /etc/init.d/ssdb.sh
```

这样, ssdb-server 就可以随着操作系统启动和关闭了.

使用 SSDB 命令行客户端(ssdb-cli)

SSDB 的命令行客户端 ssdb-cli 可用于日常管理, 包括查看 SSDB 的运行状态(info), 还能做一些 API 没有的操作, 如清空整个数据库.

运行 ssdb-cli

连接到 127.0.0.1:8888,

```
$ /usr/local/ssdb/ssdb-cli -h 127.0.0.1 -p 8888
ssdb (cli) - ssdb command line tool.
Copyright (c) 2012-2013 ideawu.com

'h' or 'help' for help, 'q' to quit.

ssdb 127.0.0.1:8888>
```

你可以输入字母 h, 然后按回车查看帮助信息.

进行 KV 操作

```
ssdb 127.0.0.1:8888> get k
error: not_found
(0.007 sec)
ssdb 127.0.0.1:8888> set k 1
ok
(0.000 sec)
ssdb 127.0.0.1:8888> get k
1
(0.000 sec)
ssdb 127.0.0.1:8888> del k
ok
(0.000 sec)
ssdb 127.0.0.1:8888> get k
error: not_found
(0.000 sec)
ssdb 127.0.0.1:8888>
```

进行 hashmap 操作

```
ssdb 127.0.0.1:8888> hsize h
0
(0.007 sec)
ssdb 127.0.0.1:8888> hset h k1 vb
ok
(0.000 sec)
ssdb 127.0.0.1:8888> hset h k2 va
ok
```

```

(0.000 sec)
ssdb 127.0.0.1:8888> hset h k3 vc
ok
(0.000 sec)
ssdb 127.0.0.1:8888> hscan h "" "" 10
key          value
-----
k1           : vb
k2           : va
k3           : vc
3 result(s) (0.000 sec)
ssdb 127.0.0.1:8888> hclear h
hclear 'h' 3 key(s).
ssdb 127.0.0.1:8888> hscan h "" "" 10
key          value
-----
0 result(s) (0.000 sec)
ssdb 127.0.0.1:8888>

```

进行 zset 操作

```

ssdb 127.0.0.1:8888> zsize z
0
(0.000 sec)
ssdb 127.0.0.1:8888> zset z k1 2
ok
(0.000 sec)
ssdb 127.0.0.1:8888> zset z k2 0
ok
(0.000 sec)
ssdb 127.0.0.1:8888> zset z k3 1
ok
(0.000 sec)
ssdb 127.0.0.1:8888> zscan z "" "" "" 10
key          score
-----
k2           : 0
k3           : 1
k1           : 2
3 result(s) (0.000 sec)
ssdb 127.0.0.1:8888> zclear z
zclear 'z' 3 key(s).
ssdb 127.0.0.1:8888> zscan z "" "" "" 10
key          score
-----
0 result(s) (0.000 sec)

```

SSDB 应用程序开发

SSDB 的客户端 API 支持 PHP, Java 等语言, 下文通过一个例子, 如何使用 SSDB 开发一个在线书店网站, 以 PHP 为例.

示例 1: 在线书店

保存书籍信息

```
$book = array(
    'id'=>'10001',
    'title'=>'Honglou Meng',
    'author'=>'CaoXueqin',
    'pub_year'=>'1784'
);
$ssdb->hset('books', $book['id'], json_encode($book));
```

书籍信息经过 `json_encode()` 序列化成本文后, 以书籍 ID 为 key 保存在一个 Hashmap 集合里. 当然, 你也可以使用其它的序列化方式, SSDB 支持任意字节流作为数据.

建立排序索引

在 books 集合中, 书籍信息按 ID(字符串, 不是整数) 进行排序, 但实际应用中, 书籍的排序需求有很多, 例如按出版日期年份, 所以需要 Zset 来建立排序索引.

```
$score = intval($book['pub_year']);
$ssdb->zset('books_sorted_by_pub_year', $book_key, $score);
```

如何还有其它的排序需求, 再使用更多的 Zset 来建立索引.

分页浏览

```
// list 10 books that published after year 1784
$zset = 'books_sorted_by_pub_year';
$book_keys = $ssdb->zscan($zset, '', 1784, null, 10);
$books = $ssdb->multi_hget('books', array_keys($book_keys));
```

这段代码获取了 1784 年之后(不包括 1784 年)出版的 10 本书籍. Zscan() 函数从 Zset 排序集合中读取这 10 本书的 ID, 然后批量从 books Hashmap 集合中读取 10 本书籍的详细信息.

Zscan() 函数的第 2 个参数被设置为空字符串, 该参数用于处理相同年份的情况, 这里为了简单演示, 所以传空字符串, 表示不关心相同年份的情况.

第 4 个参数被设置为 null, 这个参数表示截止年份, 通过 null 来表示忽略截止年份.

相关链接

SSDB 项目主页和源码下载

<https://github.com/ideawu/ssdb>

SSDB 作者的网站

<http://www.ideawu.net/blog/>

SSDB API 文档

(English)<http://www.ideawu.com/ssdb/docs/php/>

(中文)http://www.ideawu.com/ssdb/docs/zh_cn/php/

Redis

<http://redis.io/>

LevelDB 项目主页

<https://code.google.com/p/leveldb/>