GOPS 2016
Shanghai

2016 全球运维大会

上海站

重新定义运维

会议时间： 9月23日-9月24日

会议地点： 上海・雅悦新天地大酒店

主办单位： 开放运维联盟 OOPSA Open OPS Alliance　高效运维社区 GreatOPS Community

指导单位： 数据中心联盟 Data Center Alliance

Production & Development like Google Using Open Source Technology

Minghua Ye (Google)

Scalability is vital for Cloud

1. Royal wedding: 15M Pageview and 42k QPS
2. Workiva: SEC reporting for Fortune 500
3. Spotify: 700,000 events per second

The cornerstone of a scalable system

1. Distributed lock and storage (mutex and register)
    - Chubby
    - Zookeeper
2. Service discovery (pthread)
    - Etcd
    - SkyDNS
3. Load Balancing (scheduler)
    - Google network/HTTP(S) balancer
    - AWS Elastic loadbalancer
    - HAproxy / NGINX
4. Protobuf (ipc)

Distributed Lock and Storage

1. Synchronization
2. Master Election
3. Global sequence number
4. Naming service
5. Distributed, persistent file system with strong consistency.

Automated Service discovery

1. Autoscale
2. Auto failover
3. Zero config

Load Balancer on Google Compute Engine

1. Google network load balancer
   - Support TCP / UDP
   - Session affinity
   - Websocket
2. Google SSL proxy
   - SSL termination
   - Websocket
   - TCP / SSL
3. Google HTTP(S) load balancer
   - SSL termination
   - Support HTTP2
   - Support cloud CDN

Customize load balancing

1. Proprietary hardware or software.
2. Open source
   - HAproxy
   - NGINX

Protobuf

Protocol buffers are a mechanism for describing extensible communication protocols, service definition and on-disk structures.

1. Backward compatibility
   - E.g. logs written in 2008 can still be read and used today.
2. New fields can be added without breaking backwards-compatibility.
   - Frontend server and backend server can be release at different schedule.
   - Development and testing can happen in parallel.
3. Works universally across binaries / languages / platforms
4. Monolithic code base, loose coupled services.

The core libraries used by Google service (C++)

1. Gflags
2. Glog
3. Googletest

Command line flags

https://gflags.github.io/gflags/

1. Command line flags are the most common way to control a binary behavior
   - Hide / unhide features (--enable-new-feature=true)
   - Fine tune the binary behavior (--max-request-timeout=10)
   - Store program settings (--language="english" --font-file=/srv/fancy.font)
2. Flags are global, definition can be localized
   - No more endless if-then-else parsing of args in the main
   - DEFINE_int32(port, 0, "What port to listen on");
   - DECLARE_int32(port) and refer it as FLAGS_port.

Logging

https://github.com/google/glog

1. Logging levels: INFO, WARNING, ERROR
   - LOG(INFO) << "Found " << num_cookies << " cookies";
2. CHECK Macros
   - CHECK(fp->Write(x) == 4) << "Write failed!";
3. Verbose Logging
   - VLOG(1) << "I'm printed when you run the program with --v=1 or higher";
   - --vmodule=mapreduce=2,file=1,gfs*=3 --v=0
4. Failure Signal Handler: get stack trace on fatal signal.
5. Work together with fluentd.

Googletest

https://github.com/google/googletest

1. Unit test
   - Tests should be independent and repeatable.
   - Tests sit together with the code organized in test cases.
   - Tests should be portable and reusable.
   - Tests should be fast and focused.
2. Mocking
   - lets you create mock classes trivially using simple macros.
   - supports a rich set of matchers and actions.
   - handles unordered, partially ordered, or completely ordered expectations.
   - is extensible by users.

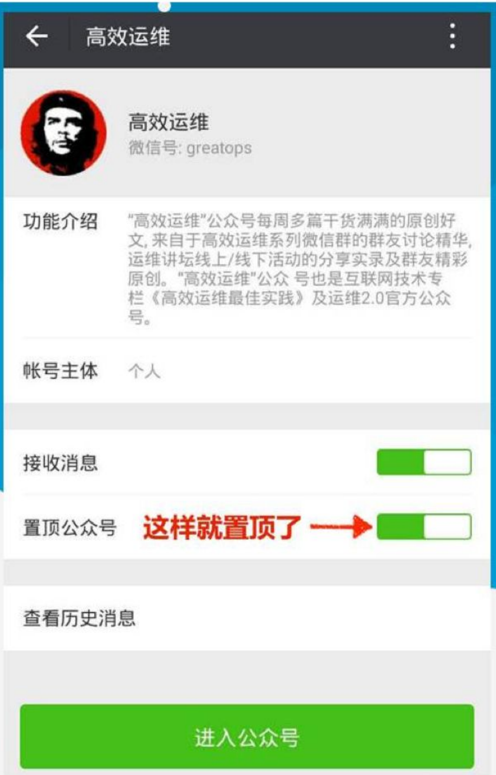
Thanks

高效运维社区
开放运维联盟

荣誉出品

想第一时间看到高效运维公众号的好文章么？

请打开高效运维公众号，点击右上角小人，并如右侧所示设置即可：



高效运维

高效运维
微信号: greatops

功能介绍 "高效运维"公众号每周多篇干货满满的原创好文, 来自于高效运维系列微信群的群友讨论精华, 运维讲坛线上/线下活动的分享实录及群友精彩原创。"高效运维"公众 号也是互联网技术专栏《高效运维最佳实践》及运维2.0官方公众号。

帐号主体 个人

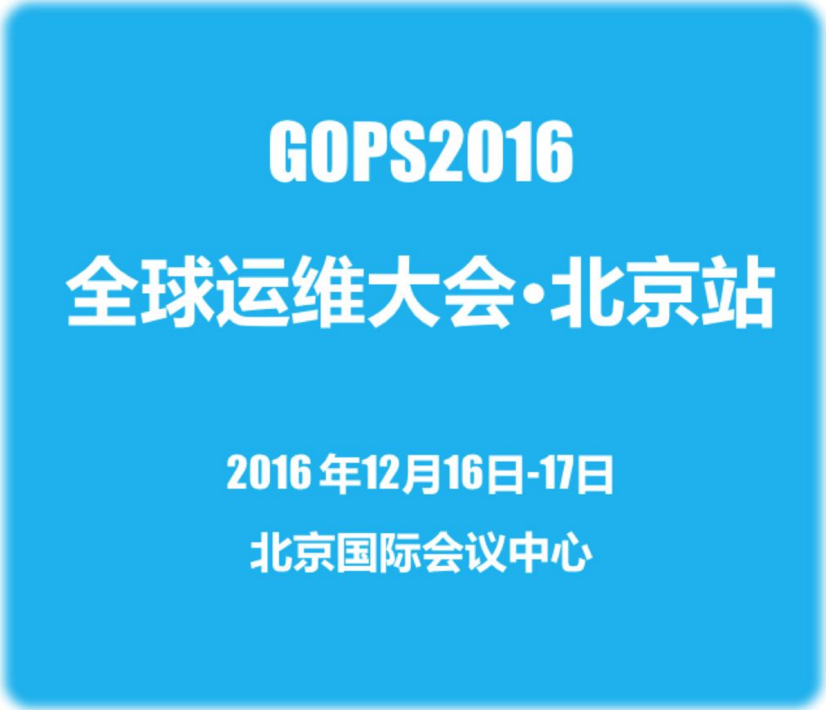
接收消息 ●

置顶公众号 这样就置顶了 ⟶ ●

查看历史消息

进入公众号

GOPS2016 全球运维大会更多精彩



GOPS2016

全球运维大会·北京站

2016 年12月16日-17日

北京国际会议中心