# 高效运维社区分享系列

# 运维三十六计 v0.6

## 专栏撰写专家（按首字母拼音顺序）：

| 姓名 | 公司 |
| --- | --- |
| 范伦挺 | 阿里巴巴，技术专家 |
| 盖国强 | |
| 高向冉 | 腾讯，技术总监 |
| 韩方 | YY 直播，安全总监 |
| 梁定安 | 腾讯，织云负责人 |
| 潘晓明 | 京东，开发测试专家 |
| 涂彦 | 腾讯，技术总监 |
| 万千一 | 京东，测试经理 |
| 王磊 | 华为，资深架构师 |
| 胥峰 | 盛大游戏，高级研究员 |
| 徐奇琛 | 京东，技术总监 |
| 叶金荣 | 知数堂，联合创始人 |
| 闫林 | 中兴通讯，IT 技术学院院长 |
| 周小军 | 腾讯，运维专家 |
| 周正中 | 阿里巴巴，技术专家 |
| 张永福 | 大河云联，资深架构师 |
| 赵舜东 | |
| 张乐 | 百度，资深敏捷教练 |

# 目录

三十六计 - 运维管理 - 涂彦

三十六计 - 日常运维 - 梁定安

三十六计 - 自动化运维 - 胥峰

三十六计 - 基础监控 - 赵舜东

三十六计 - 安全运维 - 韩方

三十六计 - 网络运维 - 张永福

三十六计 - 数据库运维 - 周小军

三十六计 - MySQL 运维 - 叶金荣

三十六计 - PostgreSQL 运维 - 周正中

三十六计 –Oracle 运维 – 盖国强

三十六计 - 存储运营 - 高向冉

三十六计 – 数据中心节能运维 – 闫林

# 三十六计 - 运维管理 - 涂彦

1. 运维管理的两个主要目标，管人与理事。
2. 运维管理需要经历的三个阶段，生存、生长、生态。
3. 未来十年团队中坚力量是年轻一代。
4. 认同年轻团队的五个价值观：不惧权威、自主管理、HERO、女汉子、理想现实主义者。
5. 打好领导与领袖的组合拳，用法定权责奖赏和鼓励员工，用个人专业能力影响员工，用好事半功倍，用不好人财两空。
6. 运维团队在企业中的角色，不是外包和救火队员可以代替的，而是要与企业和业务共同成长的角色。
7. 不论公司规模大小，运维团队都可以从“是高想象力团队还是低想象力团队，是高创造力团队还是低创造力团队”来看发展潜力。
8. 高想象力与高创造力来源于如何建立运维价值输入输出关键路径，即 INPUT 与 OUTPUT 机制。
9. 对于运维团队的架构职能，平台型与垂直型是驱动团队发展的内在因子，而不仅仅只是一种僵硬的组织形式。
10. 混搭“平台与垂直”，是一种面向服务对象的灵活组织管理模式。
11. 运维团队的能力体系建设应该像互联网产品一样，不断做好迭代与交付能力。
12. 团队的发展好比一个人的成长，除了好的组织架构外，还需要好的思维模式。
13. 成长思维模式中，需要不断挖掘与释放团队内各种成员的潜在能力，做好兼容与适配。
14. 导师制度，对一个技术团队的人才培养非常重要，不同阶段不同领域都需要灯塔指引我们，在学习知识同时收获友情。

15. 营造创新环境也是成长思维模式中必不可少的环节，坚持与服务对象保持紧密沟通，特别是对一件事的持续专注。
16. 对遇到成长瓶颈的成员，需要让其自身成长的诉求与不断推陈出新的业务需求发生化学反应，才能走出困惑，重拾信心。
17. 运维团队应该根据企业的文化价值观形成带有鲜明团队自身属性的文化价值观，并且这样的团队文化能深入人心。
18. 团队文化可以看成是团队内各成员对于彼此工作方法的一种高度共识。
19. 随着企业业务规模发展，不少管理者将面临远程团队管理，要正确认识远程团队在企业发展中的真正价值。
20. 管理远程团队要经历的三个阶段：活下去（初创期）、追赶与超越（发展期）、自循环（稳健期）。
21. 管理远程团队要解决的三个问题：时间、距离、文化。
22. 时间：远程团队如何解决各类工作信息高效同步与任务执行。
23. 距离：远程团队如何自我发展、外部合作如何开展。
24. 文化：远程团队如何读懂和落地公司、部门文化。
25. 当作为一个职业经理人空降一个运维技术团队时，首先要学会快速融入，而不是快速换人。
26. 空降管理中，基于管理与被管理双方强烈的认知欲望，如何快速压缩认知成本，认识别人眼中的自己， 是破冰的关键路径。
27. 作为运维团队管理者，要学会培养自身与团队在核心业务上的技术洞见能力。
28. 运维团队随着业务发展，自身在经历不同时期的建设过程中将体现出技术洞见为团队能力提升所带来的重要变化。

29. 技术洞见在业务扩张建设和实现业务转型中，都扮演着关键角色，是管理者及团队的重要专业能力。
30. 运维管理者要理性看待技术洞见带来的风险，特别是失败案例以及早期技术方案雏形的不完善。
31. 创新与关注核心业务是可以不冲突的，运维管理者要围绕核心业务进行技术洞见的灰度和全量。
32. 运维团队作为面向服务对象的技术团队，要具备产品思维能力。
33. 对于运维服务的对象，可以分为“客户”与“用户”，前者偏向于企业内部业务团队，后者偏向于企业外部最终用户。
34. 运维的产品思维可以体现在工具文化中：创新方式解决问题、利用该方式快速成长与扩张、以产品为基础。
35. 运维团队可以用创业模式来管理建设中的项目以及待评估的技术洞见。
36. 将创业模式的项目管理引进运维团队，可以激发团队的狼性，产生更多好的运维服务与产品。

# 三十六计 - 日常运维 - 梁定安

1. 运维值班要保证手机电量充足，7x24 为业务待命
2. 应对故障要先恢复再排查，无计可施重启试试
3. 对不可逆的删除或修改操作，尽量延迟或慢速执行
4. root 操作需留神，sudo 授权更安全可控
5. 配置文件不要写死 IP，巧用名字服务解耦更高效
6. 运维脚本和工具要版本化管理
7. 批量操作，请先灰度再全量
8. 删除操作脚本请交叉检查二次确认
9. 采用高可用的集群化部署，应防止单点
10. 将重复三次以上的操作脚本化
11. 开放外网高危端口需谨慎，网络安全要牢记
12. crontab 写绝对路径，输入输出重定向
13. 慎防进程 D 状态，及时监控保可用
14. 一人一次只做一个变更，降低人为失误风险
15. 变更操作先备份再修改
16. 数据备份任务要监控，并定时检查备份档的有效性
17. 尽量提前预警，避免告警救火
18. 敏感权限应定期回顾和检查，及时清理离职转岗的人员权限
19. 服务上线一定要有监控，保证质量可度量
20. 灾难的紧急预案一定要有演练的机制，养兵千日用兵一时

21. 修改内核参数需区分一次性修改或随机启动修改

22. 对生产环境的变更操作后，要有持续关注机制，确保服务质量不受影响

23. 尽可能保证发布操作能被回滚，并且发布故障要优先回滚

24. 保持应用运行的独立性，防止交叉依赖的程序存在

25. 运维工作互备，工作交接要留文档

26. 从每个故障中学习和提高，避免重犯同一个错误

27. 每个偶然的故障背后都深藏着必然的联系，找到问题根源并优化掉

28. 运维的标配软技能：责任心、沟通力、执行力

29. 运维规范变现步骤：文档化、工具化、系统化、自动化

30. 日常运维口令：打补丁、传文件、批处理、改配置、包管理、看监控

31. 容量管理要做好，每日关注高低负载

32. 用流程保证质量，用自动化保证效率

33. 日志管理使用轮换机制，防止硬盘空间使用率无限增大

34. 先量化管理运维对象，再优化管理运维对象

35. 容量规划牢记从 3 个角度评估：主机负载、应用性能、业务请求量

36. 保持运维对象的标准化与一致性，如处女座般梳理整洁生产环境

# 三十六计 - 自动化运维 - 胥峰

1. 思想上要树立“以自动化运维为荣，以手动运维为耻”的荣辱观。
2. 自动化运维体系的设计“以人为本”，减少学习成本才能更有效的发挥作用。
3. 自动化运维体系要涵盖所有运维需求，是全面的和完整覆盖的。
4. 自动化运维的产物必须是平台，只有平台才能永续。
5. 简洁的操作流程是自动化运维平台的设计原则。
6. 自动化运维的终极目标是消灭 SecureCRT 和 Putty 等一切远程客户端，让平台成为唯一入口。
7. 自动化运维的第一步是脚本化，通过脚本构建可重复的基础架构和环境。
8. 脚本加入版本控制，以能够追溯和审计变更。
9. 脚本语言要统一，以提高脚本的可维护性。
10. 你不用造轮子，可以先考虑开源方案加二次开发满足运维需求。
11. 高效是自动化运维的要求，使用多进程或者事件模型等提高并行效率。
12. 设计良好的 kickstart 提高物理机交付的效率和安全性。
13. 使用不同烧制级别的虚拟机镜像提高云计算资源的交付效率。
14. 安全必须是内置在自动化运维中的，通过主动发现和深度防御机制保障安全。
15. 网络层面使用防火墙保障集中控制节点的安全。
16. 采用双因素认证保障集中控制节点的系统授权访问。
17. 持续的网络安全扫描减少误操作带来的风险。
18. 集中控制节点和被控节点加密数据通信。
19. 循序渐进是从头创建运维自动化体系的正确姿势，不要一开始就设计大而全的系统，从

最痛的痛点开始解决。

20. 可以使用价值流程图分析当前的效率瓶颈和确认痛点。
21. 自动化运维的底层数据必须保证完整性，技术手段与流程保障并行。
22. 分层设计 CMDB，基础数据统一管理，业务数据向下授权。
23. 资产流转和变更中加入流程控制和审计，防止失控和数据不一致。
24. 以自动探测和上报提高 CMDB 配置的效率和维护数据准确性。
25. 监控体系的自动化是整个体系的纽带，它贯穿着事件和故障自愈。
26. 设计大规模监控体系的自动注册功能，不以手动添加被监控指标。
27. 业务分组、服务器角色分组，自动匹配监控项目。
28. 通过数据分析聚合和关联监控数据，提供故障排除和容量规划的有效信息。
29. 监控的目标是保障业务价值，不但要监控基础架构和应用端口，而且更要监控业务数据，比如订单数据和游戏玩家数量等。
30. 坚持持续改进的监控目标，持续减少漏报和误报比例。
31. 规范业务日志的格式化输出，统一日志的集中存储和分析。
32. 设计自动化的数据备份体系，设计通用的备份客户端。
33. 备份客户端内置加密功能，密码由服务器下发。
34. 以并发或者 UDP 方式提高备份传输效率。
35. 结合离线备份和在线备份，提供备份文件的自动化下载接口。
36. 自动化备份数据恢复测试，检查数据有效性。

# 三十六计 - 基础监控 - 赵舜东

1. 监控对象收集要完整，不然每次故障就刚好是忘记监控的对象。
2. 监控对象要完全理解，这样才知道要监控哪些指标。
3. 监控对象指标需要提前确定性能基准线。
4. 使用 IPMI 监控物理服务器，建议为服务器均配置远程控制口，配备单独的网络。
5. 使用 SNMP 监控物理设备，建议使用 V3 版本。
6. 机房巡检不能少，设置巡检制度，节假日前一定要巡检。
7. 机房网络情况要时刻掌握，开源的选择可以用 Smokeping。
8. 基础的网络安全监控，可以定期扫描网络端口，发现异常端口进行告警。
9. 开源的 Web 漏洞扫描器 w3af，定期对线上服务进行安全扫描。
10. CPU 监控不可少的对象：CPU 利用率、运行队列、上下文切换。
11. 使用 top、vmstat、mpstat 监控 CPU 性能指标，CPU 利用率中 user/system 比例维持在 70/30。
12. 内存监控主要关注使用率，free、vmstat 输出很详细，注意添加告警触发器。
13. 磁盘利器 iotop，有硬盘报警需及时更换，千万不能等等再说。
14. 网络监控利器 iftop，TCP 状态监控不能少，time_wait 多不用怕，可以调整内核参数缓解，close_wait 多才恐怖。
15. 在性能测试场景下，nmon 可以给测试工程师提供更好的性能报表。
16. Apache、Nginx 均有状态模块，可以直接开启，并集成到监控平台。
17. Memcached、Redis 均自带状态命令，使用 nc 就可以进行通讯。
18. 各种网络服务均提供相关的监控接口，集成到监控平台之前要理解受监控对象。

19. 所有线上业务都要添加 Web 监控或者 API 监控，监控其存活状态。
20. Nagios、Cacti 都不错，Zabbix 可以挑大梁。
21. 监控有痛点可以再试试 open-falcon，国外的 Datadog 也不错。
22. 想要更灵活的监控：Sensu、Grafnana+collectd+InfluxDB 等多种选择。
23. 监控也需要标准化，统一脚本、统一模板。
24. 通过 API 和 CMDB 集成，保证监控覆盖率。
25. 大规模监控使用 Zabbix-Agent 主动模式。
26. 为相同类型的监控对象创建 Screen 更直观的进行展示。
27. Zabbix Proxy 专门为分布式监控做准备，多机房监控必备。
28. Zabbix Discovery 是利器，利用好 Low Level Discovery 会事半功倍。
29. 使用 TiDB 替换 MySQL 作为 Zabbix 后端存储，解决监控数据量大的性能瓶颈。
30. 某些重点业务指标也可以集成到 Zabbix 中，如日活用户、注册用户、每分钟订单等。
31. 告警方式可以选择邮件、微信和短信，重要告警短信是必备的。
32. Zabbix Agent 可以自动进行注册、网络发现可以自动添加监控。
33. 使用 Zabbix API 主动管理监控对象，结合 CMDB 自动化管理监控。
34. 最简单的故障自愈可以使用 Zabbix Action 触发对应操作。
35. 告警的去重可以借助于 Elasticsearch，对告警进行分词处理再去重。
36. 使用单独的工具对监控数据进行计算，努力实现动态化告警阈值。

# 三十六计 - 安全运维 - 韩方

1. 进程启动权限最小化，尽可能使用非 root 账号启动进程

2. 禁用操作系统不再使用的账号，免密码登录的账号要慎重！

3. 关闭 telnet 等明文远程登录服务

4. 停用和关闭无用的服务，系统服务最小化

5. Linux 下的 ps，netsat 系统命令看到的不一定是操作系统的返回信息， 也可能是木马伪装后的信息，系统命令有可能篡改，系统内核调用可能被替换

6. 大量的会话状态跟踪表 full 日志异常也可能是被攻击导致；

7. CC 攻击(http flood)服务器上最简单的对抗方法就是限制单 ip 的同时并发请求数;

8. 定期或不定期的漏洞扫描，可以使用开源软件，也可以自主研发

9. Syslog,authlog 等日志定期备份，便于安全事件的追溯和审计

10. Linux 下内核参数优化不仅仅可以提高性能，也可以提升防御攻击能力，比如：tcp_syncookie, ip_conntrack_max 等

11. 使用 selinux 或 appmrmor 可以提升 Linux 的安全防御等级,

12. 操作系统的 history 条目要限制一定数量，尽可能不要过大，一旦被入侵，可以直接看到命令历史输入， 避免入侵后的“战果”被进一步扩大；

13. 重要密码一定不能同其他互联网账号密码相同 ,特别是同其他小网站的账号密码相同，避免被撞库

14. 切记！ 密码口令不能明文保存在服务器的某个文件上

15. Linux,mysql,redis 等密码口令要有一定复杂度，不能过于简单；

16. 密切关注操作系统(ubuntu，centos 等)的 0day 漏洞，及时升级版本或补丁；

17. 数据安全方面，账号等敏感数据在数据库中的保存一定不能明文， 同时避免密文可逆和碰撞分析，可以通过 salt+sha1 等

18. 定期更换相关系统（操作系统、管理后台等）的密码是一个好习惯！

19. 多因素认证可以进一步提升安全防护等级，比如密码+证书

20. 切记！及时删除运行服务器上的源代码，测试代码以及文档， 一旦服务器被入侵，源代码或者测试代码将导致入侵的影响被进一步放大

21. 运行的业务进程尽量不要输出敏感信息到日志文件中，比如避免 java 代码打印数据库连接的账号信息等；

22. Shell 或 python 等脚本代码的敏感逻辑一定要进行加密， 比如 shell 中的数据库访问使用的账号和密码就需要进行加密来提升安全性

23. 安全意识培训要不定期宣导！ 持续宣导的过程， 不仅仅针对运维，也针对开发和测试；

24. 安全运维是一个立体工程，尽可能降低每个环节的风险，才能降低整体的风险面！ 单一防御面不可能 100%

25. 当服务器数量达到一定量级的时候， 相关漏洞扫描，入侵检测很难手工实施，尽可能自动化；

26. 不要低估数据化和可视化对于安全运维工作的价值

27. Tomcat 等管理后台一定要限制访问 ip，敏感后台管理系统必须白名单原则！

28. SSH 等远程登录一定要限制访问 ip 或者限制跳板机 IP

29. Iptables 的防火墙规则数量过多，影响性能，可以使用其他基于 hash 查找的防火墙规则实现的组件

30. 访问控制规则最小化原则， 白名单规则安全性高于黑名单规则，限制访问 ip，限制

被访问 port，限制访问 protocol

31. 除非特殊要求，一定要限制缓存类 Mogodb, redis, memcache 的匿名登录默认配置

32. 密切关注开源组件(MySQL, Nginx, Apache, Openssl 等)的 0day 漏洞，技术升级版本

33. Nginx/Tomcat 等容器配置访问权限尽可能最小化，比如限定于仅可读写当前目录，避免入侵后的影响扩大到其他的目录

34. Nginx 里面限制单 ip 的并发连接数可以缓解 CC 攻击带来的影响

35. PHP 的相关危险函数和不需要的远程功能可以关闭

36. Apche/Nginx 等定制统一 40x 或 50x 等错误页面返回，避免显示业务的错误敏感堆栈逻辑等；

37. 引用 Strusts,Openssl 等第三方库尽可能使用公司统一的库文件，或者相对较新版本的第三方库文件版本，避免引入很旧的版本第三方库导致漏洞

# 三十六计 - 网络运维 - 张永福

1. 生产网络的变更切记三思而后行，一个回车敲下去是永远无法撤回。
2. 网络运维工程师的苦逼程度直接反应了公司运维系统的自动化程度。
3. 网络攻城狮要想解放自己，要么学会 coding，要么和程序猿搞好关系。
4. 运维人员的经验都是通过踩坑积累出来的，你如果不想掉坑里，就需要有一颗好学好问的心，因为运维的道路上不止有坑，坑里还有钉。
5. 没有 AAA 认证的网络不是好网络。
6. 网络中的单点设计总会在关键时刻要了你的命。
7. 好的割接方案能让你提前 2 小时睡觉。不好的割接方案能让你一星期睡不好。
8. 网络监控不是监控网络，目的是监控业务。
9. 不要轻易相信厂商的方案，在 lab 里面验证后再上线，你比厂商懂自己网络上的业务。
10. 割接方案提前写在纸上，而不是割接时留在脑子里
11. 不惧怕故障，惧怕的是没有排障思路
12. 想好方案选产品，还是选好产品组方案
13. 重要割接最好有 A/B 角，包括方案评审和实施
14. 链路的物理承载类型分清楚：裸纤直连、传输电路或是二层专线
15. 管理网络与业务网络要理顺，不管带内还是带外，逃生路径都要准备好
16. 链路死了不可怕，可怕的是不死不活，频繁闪断
17. 面对闪断，要确定好抑制策略和回切策略
18. 传输运维工程师三板斧：看告警、查光功率、环回测试
19. 变更方案合格的定义是：交给不是写方案的人做变更也能顺利完成割接

20. 变更执行的关键，是现场实施人员受控
21. 意识问题，提高重视程度。往往都是小变更出现故障，大变更因为非常重视，一般不出故障。
22. 制定变更方案模板，统一按照模板执行。
23. 严格按照变更方案执行，与预期不符，必须回退，并重新安排变更时间。
24. 变更方案审批制度建立，审核不通过，必须打回去。
25. 变更前环境检查、信息收集必须到位，变更后的前后对比。
26. 从建设抓起，不给后续环节留坑。
27. 明确运维红线，不能触碰高压线
28. 运维体系化建设，利用制度保障效率
29. 运维建设没有最好的模板，只有最适合的模式。所谓的各种体系、标准，在实际中需要谈落地。
30. 运维变更中的人、过程、技术都是浮云，重要的是有没有达到安全变更的目的。
31. 建立服务化、产品化，取代人肉运维
32. 建立完善的流程制度是运维管理的核心价值
33. 谈网络运维要看上下游环境，下有服务器、线路，上有应用、业务
34. 口说无凭，以工单办事
35. 故障处理的目的不是找 rootcause，而是恢复业务
36. 每个运维工程师都要有自己的 backup

# 三十六计 - 数据库运维 - 周小军

1. 任何时候做好最坏的打算
2. 坚持数据库运维定期演习
3. 核心岗位手机 24 小时畅通状况，任何岗位都要主备责任人
4. 没有规则创造规则，有规则遵守规则；
5. 养成日常巡检核心监控属性的习惯
6. 对生产环境保持敬畏之心
7. 非工作时间不要实施普通变更
8. 变更自动推送通知和报告，保持信息对齐
9. 上线 SQL 先 Explain 一把，执行计划可以做一定的固化
10. 知己知彼，了解所做操作产生的结果才去做
11. 减操作确保可逆，最少一套恢复方案，重大变更要有操作和回滚方案，要双人检验且审批通过
12. 数据库要具备限流能力
13. 数据迁移后要双向记录对比匹配
14. 角色权限要划分清楚，开发权限要最小化原则
15. 权限管理自助化，做好审核和审计
16. 业务初期做好分库和分表的规划
17. 建立业务放量流程沟通机制，事前周知快速扩容，事中容量监控，事后资源总结
18. 做好日常数据库容量度量，用历史数据推算下一个容量高峰
19. 节假日前做好数据库容量规划

20. 对索引要根据访问类型做战略性规划

21. 数据下线后环境及时清理，不要残留

22. 主动推动业务对热记录、肥胖记录的优化

23. 避免单点：有效可恢复的数据备份，有效可切换的从节点

24. 定期的性能优化避免业务量突增导致的雪崩

25. 精通业务，推动业务采用更合适的架构方案

26. 备份系统自动化，中心化调度，保障故障效率和可用性

27. 数据备份 100%覆盖，100%可恢复，每年至少 2 次恢复演练

28. 数据恢复手段简单高效，提纯成 WEB 化工具，减少脚本使用

29. 工具上线前要严格测试和灰度验证

30. 工具开发要实施代码审阅，工具代码逻辑间要打好日志

31. 故障处理自动化，缩短影响业务质量时长

32. 数据监控多维化，立体化，覆盖所有的监控节点和粒度

33. 数据垂直分层自动调度（内存，SSD，SAS，SATA），做到成本与效率的性价比最高

34. 数据搬迁调度自动化，聚焦资源调度管理

35. 调度任务集中化，保障关键调度任务可管理，可监控

36. 主动分析业务数据访问行为，了解业务数据生命周期，优化业务成本并推动业务改进

# 三十六计 - MySQL 运维 - 叶金荣

1. 重要事情说三遍：备份、备份、备份，定期全备+增备/差异备份，并且开启 binlog
2. 如果写成 Mysql、mySQL、MySql 的人，我看 MySQL 不适合您，改用其他的吧
3. 如果还坚持认为 MyISAM 比 InnoDB 表好的话，也请别再使用 MySQL
4. 光做好备份还不够，还要做恢复测试，并且检查数据有效性
5. 数据库密码要合规，弱密码等于没密码，没密码就等着被勒索吧
6. 管理用户和业务用户区分不同权限角色，业务用户切记不可授权过高
7. SLAVE 备库谨记关闭写入权限(read_only=1)
8. 存储过程、触发器、表分区想用就用，用好就行，有性能瓶颈优化就是
9. 绝不监听公网 IP，并用防火墙挡住非外部连接，降低被入侵风险
10. InnoDB 表一定要用自增列或呈递增属性的列做主键（该列最好无业务意义），可有效提高 InnoDB 表性能、避免主从数据复制延迟
11. 总是创建合适的索引，否则 InnoDB 的行锁会升级成为类似表级锁
12. 基数低的列，强烈不建议单独创建索引（可以放在联合索引中）
13. 联合索引中，基数高的列放在前面，基数低的列放在后面
14. 想保证宕机时数据不丢失，烧香拜佛不管用，设置双 1 才靠谱(innodb_flush_log_at_trx_commit=1 & sync_binlog=1)
15. 命令行下写 SQL 时，先写好 WHERE 条件，或先全部写好确认再三后才提交执行
16. EXPLAIN 结果中重点关注 type=All/Index，或者 Extra 中出现 Using temporary、Using filesort 的情况并进行优化
17. 性能、压力测试时，测试机客户机一定要和 Server 端分开

18. 连接数爆满时更应该调低最大连接数，而非调高，并且尽快用上 thread pool
19. SHOW PROCESSLIST 结果重点关注频繁出现的 Sending data、Sorting result、Copying to tmp table、Copying to tmp table on disk、Creating sort index、Waiting for xx lock
20. 不想 MySQL 死得快，就赶紧关闭鸡肋的 Query Cache(query_cache_type=0)
21. 默认开启 autocommit；需一次性写入大量数据时，则应关闭 autocommit，最后手工提交
22. 监控 InnoDB 表空间碎片率：ibd 文件实际大小/(Data_length+Index_length)，并决定是否需要重整表空间
23. 环境初始化之一：开启 CPU 最大性能模式
24. 环境初始化之二：关闭 NUMA
25. 环境初始化之三：使用 xfs/ext4 文件系统，以及 deadline/noop io scheduler
26. mysqld 进程占用 CPU %user 突然飙高，99.99%是因为索引不当导致
27. 优先解决频次最高的 Slow Query，其次核心业务高峰时段的 Slow Query
28. 每个表都增加 create_time、update_time 字段，对 DB 运维帮助非常大
29. 每个 SQL 条件都加上引号，并对用户输入强制类型转换，避免 SQL 注入及类型隐式转换风险
30. 只 SELECT 必要字段，不要总是 SELECT *，避免额外 I/O 读
31. 设置 innodb_buffer_pool_size 为物理内存的 50%~70%为宜
32. 疑似 SQL 注入一般都会调用 SLEEP()函数，或访问 information_schema 下的视图，每见必杀
33. 不要直接删除数据表，而是先 RENAME；删除大表用硬链接方式更高效

34. 要特别注意监控是否有内存泄露问题，尽早排除风险
35. 优化的核心目标是提高 I/O 效率，无论是增加内存，还是换高性能 I/O 设备，亦或提高 CPU 性能、增加索引等
36. 少用 TEXT/BLOB 等大对象列，每行长度字节数尽量不超过 innodb_data_page_size 的一半
37. 最后一计，想玩好 MySQL，来知数堂（http://zhishuedu.com），发现惊喜

# 三十六计 - PostgreSQL 运维 - 周正中

1. 任意字段组合查询有高招，GIN 复合倒排索引来帮忙。
2. 物联网、智能 DNS、金融、气象范围查询很苦恼，效率低下量不少，range 类型来帮忙，一条记录顶千条，查询索引 GiST 来帮忙，零点几毫秒要不要。
3. O2O，社交没有 GIS 可不得了。天气预报、导航、路由规划、宇航局、测绘局没有 GIS 也要乱。PostGIS、OpenStreetMap、pgrouting 不可不知道。
4. 监控系统要颠覆，主动问询模式能耗比低，百分之 99 是无用功。PostgreSQL 异步消息、流式计算一出手，能耗比提升 99，千万 NVPS 有木有。
5. DT 时代数据多得不得了，传统关系数据库扛不住，来看看 PostgreSQL 流式实时处理溜不溜。
6. 支付宝 AR 红包闹新年，即有位置又有图片比对，敢问数据库能不能做，PostGIS、PostgreSQL imgsmlr 亮高招。
7. 相似的数组、相似的文本、相似的分词、相似的图像数据库能处理吗？PostgreSQL 火眼金睛，实时辨别相似数据。盗图、盗文跑不掉。
8. 数据库 CPU 杀手：模糊查询、正则匹配有解吗？PostgreSQL GIN, GiST 索引一把抓，亿级数据毫秒响应很靠谱。
9. 云端高招，冷热分离、多实例数据共享。分析师、快速试错、OLTP、OLAP 一网打尽。
10. HTAP 是趋势，OLTP 数据库能同时实现 OLAP 吗？PostgreSQL 大补丸。多核并行、向量计算、JIT、列式存储、聚合算子复用。提升两个数量级小 case。
11. 商业时代，广告满天飞，提高营销转化率有高招。PostgreSQL 实时用户画像与圈人来帮忙，万亿 user tags 毫秒响应开心么。

12. 危化品管理有痛点，PostgreSQL GIS、化学类型、流计算来帮忙。
13. 群居社会关系多，金融风控、公安刑侦、社会关系、人脉分析，图式数据搜索很头疼，PostgreSQL 函数式编程，异步消息，复杂 JOIN 等手段，解决高效的图式数据查询需求。
14. PostgreSQL 递归查询有妙用。大量数据的求差集、最新数据搜索，最新日志数据与全量数据的差异比对，递归收敛扫描，提升数百倍性能。
15. 企业数据品种多，跨平台数据共享很头疼，实时性难解决。PostgreSQL 流式数据泵，延迟低，扩展性好。
16. PostgreSQL ad lock 效率高，百万/s 秒杀小 CASE.
17. 用户画像 TAG 多，万列宽表谁家有？PostgreSQL 妙招解，bitpack 支持实时用户画像，单机支持十万亿 user tags 体量，毫秒级实时圈人。
18. 物流配送、打车软件、导航软件、出行软件、高速、高铁哪个都离不开路径规划，PostgreSQL PostGIS, pgrouting, OSM, 机器学习库(madlib) 一站式解决。
19. 可靠性，要弹性，事务级可选最牛逼。PostgreSQL 金融级可靠性，事务级可控多副本，正面解决性能与可靠性的矛盾问题。
20. PostgreSQL brin 块级瘦索引，解决物联网、金融、日志、行为轨迹类数据快速导入与高效查询的矛盾。
21. 数据压缩要注意，旋转门时序数据有损压缩，列存储块级压缩。
22. 数据类型选择要注意，不要什么都用字符串，准确诠释数据类型最重要，基因类型能不能接，PostBIS 插件亮出来。
23. 数据类型选择要注意，不要什么都用字符串，准确诠释数据类型最重要，化学类型能不能接，RDkit 插件亮出来。

24. 数据预测、挖掘有插件, MADlib 来自伯克利，几百种学习算法够你用，不够还能 PLR。
25. 数据库只能增删改查？不能处理复杂逻辑？又快又狠就不怕。数据库端编程，处理复杂业务逻辑。解决一致性、低延迟问题好不好。
26. 金融行业 Oracle ProC 很流行，PostgreSQL ECPG 高度兼容 ProC。
27. 被裹脚式 sharding 吓怕了吗？这也不能那也不能要 sharding 干啥？PostgreSQL real sharding 来帮忙，裹脚布不再要，数据库水平拆分、跨平台数据融合样样行。
28. 开发规约 -命名很重要 ,比如不要使用小写字母、数字和下划线以外的字符作为对象名。
29. 开发规约 - 设计不可忽视，比如全球化业务，建议使用 UTF-8 字符集。
30. 开发规约 - QUERY 很重要，病从口入。比如任何地方都不要使用 select * from t ，用具体的字段列表代替，不要返回用不到的任何字段。另外表结构发生变化也容易出现问题。
31. 管理规约 - 安全与审计，上市公司不可少。从密码到认证、从链路到存储、从 DBA 到开发账号，一个都不能少。
32. 管理规约 - 诊断少不了，活动视图、插件、日志、DEBUG、隐含参数、PERF 样样都要了如指掌。
33. 管理规约 - 优化有高招，熟悉环境、数据库原理、操作系统、网络、业务逻辑必不可少。
34. 管理规约 - 备份与恢复，数据库要爱护，逻辑备份物理备份要得当。
35. 管理规约 - 日常维护，制度化。保养很重要，日常小保养，月度大保养，年度复盘都重要。
36. 开箱即用，用为上计

# 三十六计 –Oracle 运维 – 盖国强

1. 有效的备份重于一切，有了有效的备份，即使遭遇灾难，也可以心中有底，手中不慌；
2. 明确连续性或一致性优先原则，首要优先级在紧急故障时会直接影响决策，必须事前明确；
3. 制定应急预案和进行演习，这是确保方案有效可执行的必要工作，没有演练的预案全是纸上谈兵；
4. 建立容灾或异地备份，确保在极端情况下，可以保持数据的留存，DataGuard 架构是最简单保护手段；
5. 数据归档和读写分离，无限累积的数据必然影响性能和备份效率，建立数据归档机制、实现读写分离需要在架构上优先设计；
6. 制订规范并贯彻执行，良好的规范是减少故障的基础，全面的规范提升开发和运维人员的标准化；
7. 部署标准和完善的监控体系，监控是一切自动化运维的基础，监控可以让我们更早发现故障，更快应对故障；
8. 树立安全意识和开始安全审计，安全问题最大的敌人是侥幸，制定安全方案，定期分析数据库风险，逐步完善数据库安全；
9. 建立顺畅的部门协作流程，数据库运维外延包括主机、存储、网络、开发等，往往需要多个部门的协作才能有效解决或推进变革；
10. 测试和生产环境隔离，数据网络隔离。数据库应处于应用系统最后端，避免将其置于对外的访问连接之下，并且绝对不能在生产环境进行测试；
11. 严格管控权限，明确用户职责。遵循最小权限授予原则，避免因为过度授权而带来的安

全风险；明确不同的数据库用户能够用于的工作范围，防范和隔离风险；

12. 密码策略强化，防范弱口令带来的安全风险，定期更换密码，同时生产和测试环境严格使用不同的密码策略；
13. 限制登录工具，明确限制不同管理工具的使用场景和访问来源，防范未知工具的注入风险；
14. 监控监听日志，分析数据库访问的来源、程序等信息，确保清晰可控，记录在案；
15. 重要数据加密，尤其是用户和密码等信息，在数据库中应当进行加密存储；
16. 适时的软件升级，持续关注 Oracle 软件及更新，参考行业警示，尤其应关注已发布的安全补丁，防范已知漏洞被恶意利用；
17. 防范内部风险，绝大部分安全问题都来自于企业内部，通过规章、制度与技术手段规避安全风险；
18. 使用绑定变量，在开发过程中，严格使用绑定变量，提升性能同时防范 SQL 注入攻击；
19. 审核全表扫描和隐式转换等，这是 OLTP 系统性能的常见问题，需要在开发端进行 SQL 审核，建立开发规范；
20. 关注新版本的新特性，尤其是版本升级之后，需要提前关注和预防新特性引起的改变，如 11g 的串行直接路径读，12c 的自适应 LGWR 等；
21. 持续保存和记录 AWR 信息，建立性能基线，这是性能诊断的核心，应该持续保存或转储重要系统的性能数据；
22. 铭记 Oracle 的闪回特性，尤其是闪回查询，可以在误更新数据等操作后快速回退，纠正错误；
23. 优化 Redo 日志存储和效率，关注和优化 Log File Sync 等待，这是数据库事务的重要影响因素；

24. 禁止远程 DDL 和业务时间的 DDL 操作，限制高危 DDL 操作仅能在数据库服务器本地进行，严格禁止业务时间的 DDL 操作；
25. 避免任何不可回退的操作，谨记 rm 是危险的，在数据库内部执行 DROP/TRUNCATE 等破坏性操作时，同样应当谨慎；
26. 不要轻易删除任何一个归档日志，在归档模式一定要做好归档备份和空间监控，确保日志的连续性是恢复根本；
27. 增进对业务的理解和架构规划参与，数据库的很多优化必须基于对业务的深刻理解，最佳优化时机在于架构设计和开发环节，Oracle DBA 应该不断向前走；
28. 对生产环境保持敬畏，不放过任何性能波动疑点，不想当然和轻视任何数据操作。针对任何业务数据库的操作都不能草率，在接触数据时都不能掉以轻心；
29. 严格的变更测试和流程操作，并做到变更记录审计，变更之前做到仿真系统严格验证，形成详细流程、步骤和指令并遵照执行；记录操作日志，任何数据库操作做到有迹可查、有踪可寻；
30. 变更必须制定回退方案，不走单行线，确保出现异常时能够将系统恢复原貌；
31. 选择合适的变更窗口，不可过度乐观草率，避免陷入不可预期的变更陷阱；
32. 变更之后进行日志核查，在维护期间应当提炼摘取维护期生成的所有日志，确保无误无错；
33. 重要操作实现人员备份，在执行重要操作时由两个人同时在场，互相监督审核，不做疲劳变更和草率决策；不要在维护中冒险，当数据库的表征超出了你的预期，那么停下来，不做现场的风险性尝试；
34. 数据恢复必须具备明确的方案和步骤，在面对灾难时，不要急于进行恢复尝试，以免导致次生故障，需要明确分析、清晰决策，才能万无一失；

35. 自动化，把上述的各种策略尽可能用脚本或者工具管理实现，做到自动化。

36. 关键时刻保护现场寻求支持，在数据库出现超出常规、无法把握的问题时，要保护现场，寻求支援，避免无序尝试带来的数据损失！

# 三十六计 - 存储运营 - 高向冉

1、数据安全是底线，即使不服务也不能丢数据。

2、容量是根本，绝对不要让自己陷入没法扩容的境地。

3、做存储务必要理解业务，不理解业务的存储平台就是别人的垃圾站。

4、存储集群建立 set 标准，标准模型严格执行。

5、现网操作标准化到前台，避免后台操作

6、多份存储变更时先变更单份数据节点

7、变更先少量灰度，变更之前先准备回退方案

8、数据迁移是核心频繁操作，工具程序必须稳定，并且要支持各类型的迁移数据操作，不同机型不同量级相互间迁移。

9、索引数据很重要，带状态的模块要注意数据安全，不要随意迁移和清 cache。

10、格式化盘操作务必确认谨慎

11、业务突发要有应对预案，建立故障升级机制

12、对监控工具也要进行监控

13、更换磁盘必须检查 SN 号

14、不能过分信任自动化工具

15、现网环境要干净、统一，如果做不到，要定期扫描

16、一定要关注数据删除后回收，即时回收删除数据。

17、运维删除数据务必备份，并且要谨慎，禁止人工线上删除数据

18、磁盘更换机器死机必须在一个周期内恢复，否则无法达到 N 个 9 的要求。

19、存储机架和普通设备不一样，用电也不同，做好机架和交换机级别的容灾备份。

20、运维一定要有大招进行柔性和恢复业务，否则会死的很惨。

21、核心业务做到异地备份，同地多份是没用的。

22、存储不仅仅关注容量还要关注 inode 情况

23、数据必须要要有冷备，冷备是最后一道防线，也要监控和运营。

24、去单演习不可少，定期演习保稳定

25、热点数据分散存储，单机要能限流。

26、冷备修复数据要理解业务场景，记得流水和冷备一样重要。流水 log 也要做好备份。

27、存储机型要定制，存储模型要支持设备的更新换代

28、提早规划，存储设备制造厂商和机房建设周期都很长。

29、存储资源采购会受大环境影响，比如 ssd、内存，不止做好容量 buffer，还要做好采购 buffer

30、存储平台是 IO 操作型集群，要和计算资源一起复用做到设备最大化利用。

31、不同年限的设备性能不同，磁盘读写能力不一致，要区别对待，老化磁盘要定期淘汰。

32、存储冷热数据分离，业务一定要能识别冷数据。

33、有热点数据要进行资源隔离，上层业务加 cache。

34、存储引擎特点要熟悉，不同业务文件选择不同存储引擎

35、对于频删业务要特殊对待，删除看似场景不多，却是最消耗资源的操作。

36、分片存储场景要牢记一台机器的数据或者一块磁盘的数据影响的文件数远不止单机或单盘的比例。

# 三十六计 – 数据中心节能运维 – 闫林

1. 选址是第一要素。DC 尽量选择高海拔、高纬度、温度低、湿度适中的地点。
2. 大型 DC 制冷方式选择参考：能源利用效率是风冷<水冷<自然冷却
3. 设计全封闭型的数据中心，提高机房密闭性能，取消外窗，防止太阳辐射消耗能量，满足 LEED 要求的绿色节能建筑要求。
4. 在机房制冷控制区域对维护结构进行保温处理，防止机房相邻区域因温度、湿度差异较大而产生冷凝水，同时降低制冷的负荷。
5. 合理摆放空调设备位置，风口地板与空调保持 6 英尺以上距离，避免气流短路。
6. 对线缆穿出地板的开口处进行密封处理，优化机房地板下结构降低风阻。
7. 空调摆放与机柜排垂直；避开冷通道
8. 在数据中心机房中建设冷通道，并配置下送风机房专用风冷式精密空调；在数据中心机房中建设热通道，并配置下送风机房专用风冷式精密空调。
9. 在数据中心机房建设专用大型水冷式机房精密空调和芯片冷却管道，直接给 IT 设备芯片散热。
10. 在数据中心机房采用机房风冷式精密空调+大型新风机 1：1 配置，合理利用自然新风冷源。
11. 机房功率密度高时，空调机分散安放，应适当提高活动地板铺高和地板风口出风速度，送风均匀。
12. 有条件时可在空调机顶部接回风道，热通道上方加回风口（有吊顶机房）
13. 空调机管线尽量布置在空调机后部
14. 将大功率、高负荷的服务器摆放在机柜的底部或中间

15. 选用节能的加湿系统，能耗：超声波加湿<湿膜加湿<电极加湿<红外线
16. 设定空调最佳工况，防止出现部分空调正在加湿，部分空调正在除湿的情况
17. 推荐空调机冷凝器自动雾化技能技术，采用磁悬浮离心冷水机组，采用太空纤维的风机和冰蓄冷技术
18. 空调采用高能效比压缩机，电机使用变频系统，末端空调使用，下沉 EC 风机
19. 各种数据中心皆可广泛采用板式热交换器
20. 大型数据中心采用大容量蓄冷罐
21. 水冷背板机柜，选用高效率、模块化 UPS
22. 使用 UPS 的 ECO 模式（智能休眠），使 UPS 运行在经济状态下
23. 谐波治理：加隔离变压器；对谐波进行抑制，12 次脉冲附加 11 次滤波
24. 对电力系统进行无功补偿
25. 在配电柜断路器和 UPS 输出端加装节电器
26. 市电直供+240V 高压直流供电系统
27. 选择节能灯具，并引入智能照明系统，提高自动化程度，减少不必要的光照强度
28. 机柜加盲板，有效使用机柜封闭盲板，减少冷热空气的混合
29. 推荐柜门：网孔六角形设计，通风率 78%
30. 关闭不用的 IT 负载。找出并且淘汰没有使用的或者利用率低的设备和应用。
31. 选用低能耗 IT 设备，淘汰高能耗设备
32. 采用耐 35 度高温服务器，采用液冷服务器
33. 采用多种能源：太阳能、地热能、核能、潮汐能、风能
34. 燃气冷热电三联供系统
35. 采用数据中心微模块技术。

36. 作好除尘：空调、风机尤其是风扇、滤网除尘