

媒体云转码的演进： MapReduce、DASH 与稳定婚姻

Alan Zhuang

cheedoong@acm.org

2014 年 3 月 9 日

Outline

1 背景

2 腾讯研究院 Transcoder

- Cloud Transcoding Reusing Idle Computing Resources
- Fast Traversal, Scanning and Deletion on Common Disk FS; Fast Replacement on CDN Edge Nodes without Indexes

3 Conclusions & Experience

- How to obtain new ideas?
- How to write good patent applications?

Outline

1 背景

2 腾讯研究院 Transcoder

- Cloud Transcoding Reusing Idle Computing Resources
- Fast Traversal, Scanning and Deletion on Common Disk FS; Fast Replacement on CDN Edge Nodes without Indexes

3 Conclusions & Experience

- How to obtain new ideas?
- How to write good patent applications?

多屏时代的挑战

多屏时代的挑战

■ 多种平台

多屏时代的挑战

■ 多种平台

多屏时代的挑战

■ 多种平台



多屏时代的挑战

■ 多种平台



多屏时代的挑战

■ 多种平台



■ 多种屏幕大小

多屏时代的挑战

■ 多种平台



■ 多种屏幕大小

多屏时代的挑战

■ 多种平台



■ 多种屏幕大小



多屏时代的挑战

■ 多种平台



■ 多种屏幕大小



多屏时代的挑战

■ 多种平台



■ 多种屏幕大小



多屏时代的挑战

多屏时代的挑战

■ 多种码率



多屏时代的挑战

■ 多种码率



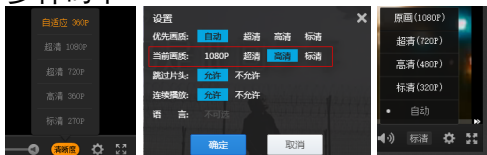
多屏时代的挑战

■ 多种码率



多屏时代的挑战

■ 多种码率



多屏时代的挑战

■ 多种码率



多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

■ 多种封装容器支持

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

■ 多种封装容器支持

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

■ 多种封装容器支持

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

■ 多种封装容器支持

多屏时代的挑战

■ 多种码率



■ 多种解码能力

	MT6572	MT6582	MT6588	MT6592
Display	960×540P	1280×720P	1920×1280P	1920×1280P
H.264 Decode	720P@30fps	1080P@30fps	1080P@30fps	1080P@30fps
HEVC Decode	N/A	N/A	720P@30fps	720P@30fps

■ 多种封装容器支持

目标与约束

- 十一假期基本完成

目标与约束

- 十一假期基本完成
- 回来后合并

目标与约束

- 十一假期基本完成
- 回来后合并
- alan: 总体，关键算法

目标与约束

- 十一假期基本完成
- 回来后合并
- alan: 总体，关键算法
- roshan: 统领协调整理，后台架构设计

目标与约束

- 十一假期基本完成
- 回来后合并
- alan: 总体，关键算法
- roshan: 统领协调整理，后台架构设计
- xiaojun, edge: 整理以往代码，补充

目标与约束

- 十一假期基本完成
- 回来后合并
- alan: 总体，关键算法
- roshan: 统领协调整理，后台架构设计
- xiaojun, edge: 整理以往代码，补充
- michael: 搜索相关，自然语言理解

目标与约束

- 十一假期基本完成
- 回来后合并
- alan: 总体，关键算法
- roshan: 统领协调整理，后台架构设计
- xiaojun, edge: 整理以往代码，补充
- michael: 搜索相关，自然语言理解
- molly: 后勤，其他

目标与约束



工欲善其事，必先利其器

What 利器?

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail
- Dropbox & WeiYun

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail
- Dropbox & WeiYun
- ACM Digital Library <http://dl.acm.org>

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail
- Dropbox & WeiYun
- ACM Digital Library <http://dl.acm.org>
- SOOPAT <http://soopat.com>

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail
- Dropbox & WeiYun
- ACM Digital Library <http://dl.acm.org>
- SOOPAT <http://soopat.com>
- Google Patents <http://patents.google.com>

工欲善其事，必先利其器

What 利器?

- L^AT_EX with a patent template
- Evernote & QQMail
- Dropbox & WeiYun
- ACM Digital Library <http://dl.acm.org>
- SOOPAT <http://soopat.com>
- Google Patents <http://patents.google.com>
- BibTeX to manage the references

整体架构

投影拟合

投影拟合

投影拟合

- 好不容易弄到一张符合要求的矢量地图

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

准确坐标和近似投影的坐标的关系：

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

准确坐标和近似投影的坐标的关系：

- 若仅有平移差别，那么 x, y 分别只是 m_x, m_y 的线性函数

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

准确坐标和近似投影的坐标的关系：

- 若仅有平移差别，那么 x, y 分别只是 m_x, m_y 的线性函数
- 若仅有两个方向上的拉伸，则 $x = f(m_x); y = g(m_y)$

投影拟合

- 好不容易弄到一张符合要求的矢量地图
- 发现该地图所采用的投影方法和高斯 - 克吕格尔投影、Lambert 投影都有些接近
- 但无论怎么调整参数，总存在稍许旋转、拉伸和扭曲

准确坐标和近似投影的坐标的关系：

- 若仅有平移差别，那么 x, y 分别只是 m_x, m_y 的线性函数
- 若仅有两个方向上的拉伸，则 $x = f(m_x); y = g(m_y)$
- 若存在旋转变换，甚至非仿射变换，则

$$x = f(m_x, m_y); y = g(m_x, m_y)$$

投影拟合

The RBF model is

$$f(x) = \sum_{i=1}^n \alpha_i g(x - x_i). \quad (1)$$

The output is a linear combination of non-linear functions of the input. The non-linearity is a function of distance only.

RBF neutral network

An RBF is the solution to the following interpolation problem:

$$\min \sum_{i=1}^n L(f(x_i), y_i) + \lambda f_H. \quad (2)$$

$$\min f_H \quad \text{st.} \quad f(x_i) = y_i, \quad i = 1, 2, \dots, n. \quad (3)$$

This is similar to a kernel density, except that the coefficients α_i are not restricted to a convex combination, and the basis function g does not have to be a density.

$$E = 1/2 \sum_i \int (y(x_i + z) - y_i)^2 p(z) dz \quad (4)$$

Let

$$f(x) = y(x) + \eta(x) \quad (5)$$

$$E(f) = 1/2 \sum_i \int (y(x_i + z) + \eta(x_i + z) - y_i)^2 p(z) dz \quad (6)$$

$$\approx 1/2 \sum_i \int (y(x_i + z) - y_i)^2 + 2(y(x_i + z) + \eta(x_i + z) - y_i)\eta(x_i + z) p(z) dz \quad (7)$$

$$(8)$$

$$\delta E = \sum_i \int (y(x_i + z) - y_i) \eta(x_i + z) p(z) dz = 0 \quad (9)$$

Let $\eta(x) = \delta(x - u)$.

$$\delta E = \sum_i \int (y(x_i + z) - y_i) \delta(x_i + z - u) p(z) dz \quad (10)$$

$$= \sum_i (y(u) - y_i) p(u - x_i) = 0 \quad (11)$$

$$y(u) \sum_i p(u - x_i) = \sum_i y_i p(u - x_i) \quad (12)$$

$$y(u) = \sum_i y_i \frac{p(u - x_i)}{\sum_j p(u - x_j)} \quad (13)$$

近似投影: $\forall Proj_k, k \in \{1, 2, \dots, P\}$

近似投影: $\forall Proj_k, k \in \{1, 2, \dots, P\}$

基于曲率的角点检测, 得到:

$(x_{ik}, y_{ik}), i \in \{1, 2, \dots, M_k\}, k \in \{1, 2, \dots, P\}$

近似投影: $\forall Proj_k, k \in \{1, 2, \dots, P\}$

基于曲率的角点检测, 得到:

$(x_{ik}, y_{ik}), i \in \{1, 2, \dots, M_k\}, k \in \{1, 2, \dots, P\}$

取相同数目 N 个对应角点, 目标:

$$\arg \min \sum_{i=1}^N [(x_i - x_{ik})^2 + (y_i - y_{ik})^2], k \in \{1, 2, \dots, P\} \quad (14)$$

近似投影: $\forall Proj_k, k \in \{1, 2, \dots, P\}$

基于曲率的角点检测, 得到:

$(x_{ik}, y_{ik}), i \in \{1, 2, \dots, M_k\}, k \in \{1, 2, \dots, P\}$

取相同数目 N 个对应角点, 目标:

$$\arg \min \sum_{i=1}^N [(x_i - x_{ik})^2 + (y_i - y_{ik})^2], k \in \{1, 2, \dots, P\} \quad (14)$$

注意: 别人用 RBF NN 来做插值/拟合、分类器, 而我们用它来消除误差, 得到了很高的精度

Trial-and-error: 10+ times

地址匹配与规整

地址状态机

IP 库合并

IP 库合并

存储：一级 Hash Array；二级一阶 KD-Tree。

IP 库合并

存储：一级 Hash Array；二级一阶 KD-Tree。考虑事实：

- 1 拥有完整 A 类地址段的单位只有 AT&T、IBM、MIT；

IP 库合并

存储：一级 Hash Array；二级一阶 KD-Tree。考虑事实：

- 1 拥有完整 A 类地址段的单位只有 AT&T、IBM、MIT；
- 2 同一地域的 IP 段往往正好是一个完整的 B 类或 C 类地址（更精确地讲是一段内的所有 IP 高 2 或 3 字节相同）。

日志分析

前端处理 — 偏移分布

Acknowledgement

- Some of the ideas were from:
“腾讯星云” 团队

Acknowledgement

- Some of the ideas were from:
“腾讯星云” 团队
- Thanks for insightful discussions with:
Nicholas Li

Acknowledgement

- Some of the ideas were from:
“腾讯星云” 团队
- Thanks for insightful discussions with:
Nicholas Li
- Library Support from:
Sergey Anatolyevich Bochkanov (Russia)

Outline

1 背景

2 腾讯研究院 Transcoder

- Cloud Transcoding Reusing Idle Computing Resources
- Fast Traversal, Scanning and Deletion on Common Disk FS; Fast Replacement on CDN Edge Nodes without Indexes

3 Conclusions & Experience

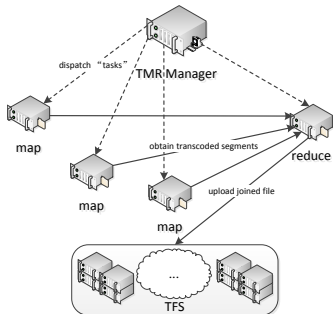
- How to obtain new ideas?
- How to write good patent applications?

Distributed Transcoding on MapReduce

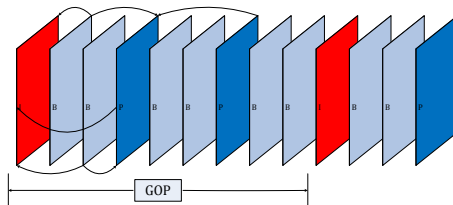
Gale Huang, Zhenhua Li, et al. Cloud transcoder: bridging the format and resolution gap between internet videos and mobile devices. ACM NOSSDAV 2012.



Cloud Transcoding Reusing Idle Computing Resources

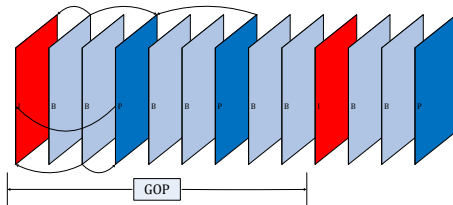


Distributed Transcoding on MapReduce



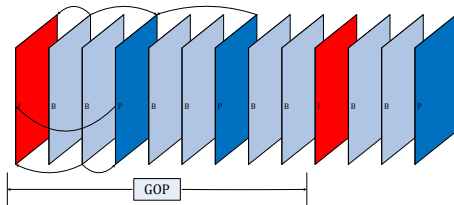
- GOP-level parallelism without REAL splitting

Distributed Transcoding on MapReduce



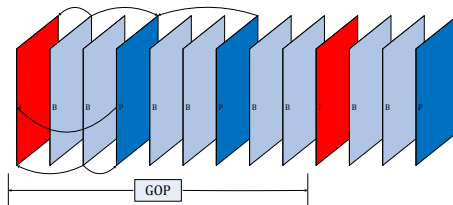
- GOP-level parallelism without REAL splitting
- Job/Map/Thread, accurate control and CPU & I/O limitation

Distributed Transcoding on MapReduce



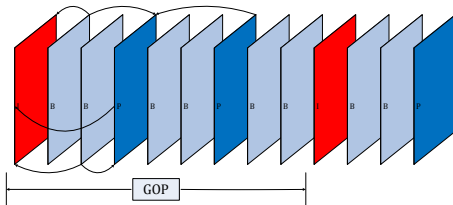
- GOP-level parallelism without REAL splitting
- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting

Distributed Transcoding on MapReduce



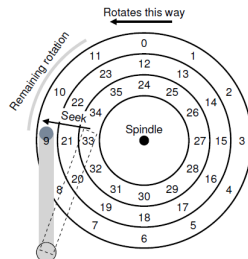
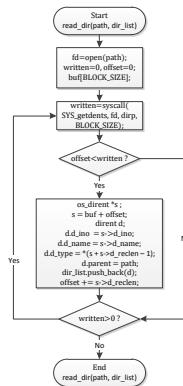
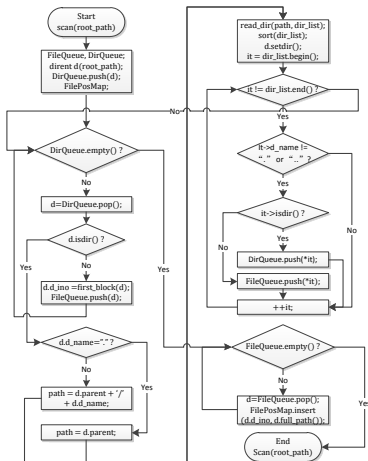
- GOP-level parallelism without REAL splitting
- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech

Distributed Transcoding on MapReduce

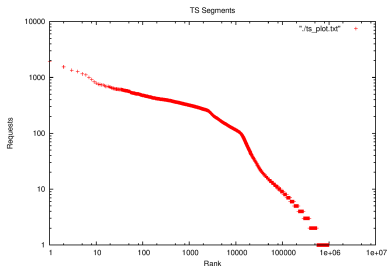
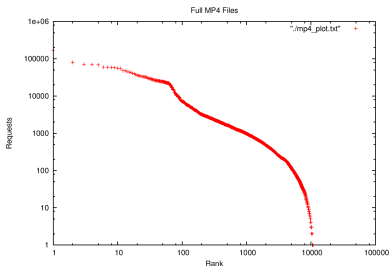


- GOP-level parallelism without REAL splitting
- Job/Map/Thread, accurate control and CPU & I/O limitation
- Real-time support for DASH and Live Broadcasting
- Migration from Computing to Storage: soul of cloud tech
- It later supported WeChat and part of Tencent Video

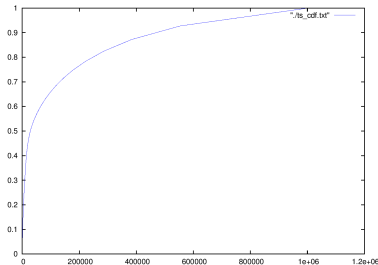
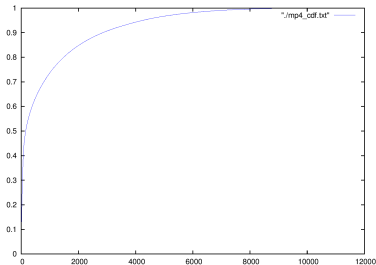
Traversal



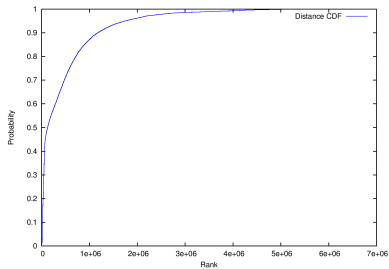
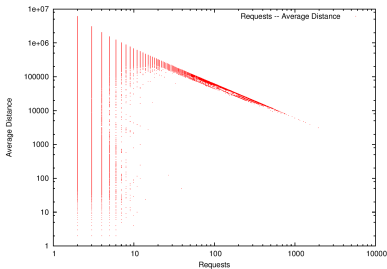
Power-Law(Zip'f) Distribution

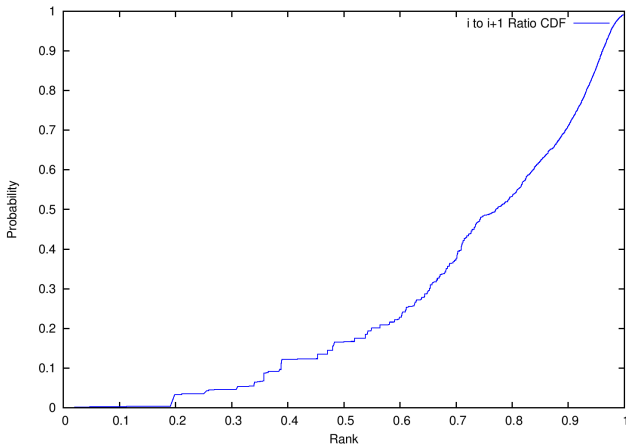


Cumulative Distribution Function (CDF)

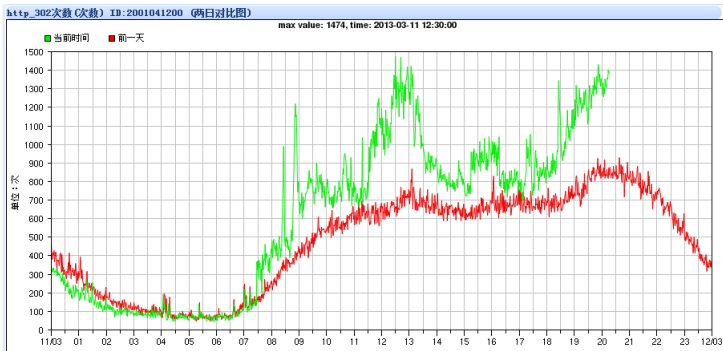


Distance and ...

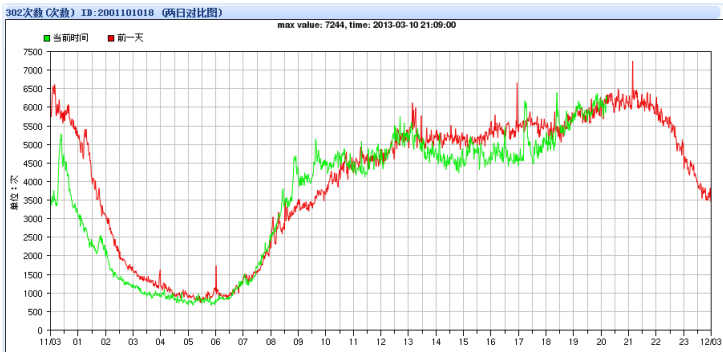


i to $i+1$ 

One of the Effects



One of the Effects



Outline

1 背景

2 腾讯研究院 Transcoder

- Cloud Transcoding Reusing Idle Computing Resources
- Fast Traversal, Scanning and Deletion on Common Disk FS; Fast Replacement on CDN Edge Nodes without Indexes

3 Conclusions & Experience

- How to obtain new ideas?
- How to write good patent applications?

How to obtain new ideas?

While working...

How to obtain new ideas?

While working...

- Think more, 多想

How to obtain new ideas?

While working...

- Think more, 多想
- Try more, 勇于尝试, 经得起多次失败

How to obtain new ideas?

While working...

- Think more, 多想
- Try more, 勇于尝试, 经得起多次失败
- Retrieve more, 避免闭门造车, 侧重别人没做过或做得不好的方面

How to obtain new ideas?

While working...

- Think more, 多想
- Try more, 勇于尝试, 经得起多次失败
- Retrieve more, 避免闭门造车, 侧重别人没做过或做得不好的方面
- Teamwork, 每个人都有自己擅长的领域 + 臭皮匠理论

How to write good patent applications?

Something look like paradoxes...

How to write good patent applications?

Something look like paradoxes...

- 数理逻辑和推理不能够成为专利
但有一些数理支持会为专利增色不少

How to write good patent applications?

Something look like paradoxes...

- 数理逻辑和推理不能够成为专利
但有一些数理支持会为专利增色不少
- 纯粹的算法不能够成为专利
但结合具体问题、设备的算法往往能够成为高质量的专利

How to write good patent applications?

Something look like paradoxes...

- 数理逻辑和推理不能够成为专利
但有一些数理支持会为专利增色不少
- 纯粹的算法不能够成为专利
但结合具体问题、设备的算法往往能够成为高质量的专利
- 专利不需要数据测量和效果数据
但详尽的数据测量会提供方法论上的支持，而效果数据则会提供更强的意义支撑

How to write good patent applications?

- 发明专利必须具有新颖性的内容，该技术领域中具有中等知识的人所不能演绎出的创造性步骤

但现在这个时代，真正具有创新性的东西太少了。所谓创新无非是：用新方法解决旧问题；或用其他领域的非新方法解决新问题。所以就要求人涉猎广泛、融会贯通。

How to write good patent applications?

- 发明专利必须具有新颖性的内容，该技术领域中具有中等知识的人所不能演绎出的创造性步骤

但现在这个时代，真正具有创新性的东西太少了。所谓创新无非是：用新方法解决旧问题；或用其他领域的非新方法解决新问题。所以就要求人涉猎广泛、融会贯通。

- 我们显然没有足够的时间来撰写专利

但一个自我感觉良好的专利需要至少一个月的业余时间