

多人协同编辑

1.0 需求背景

当前场景

- 1. 开缺陷复盘会议时,多人在飞书上同时编辑文档,每个人填写自己跟进的bug信息.完成后将excel文档上传到web平台解析.优化目标:在web平台上自动拉取bug信息,多人同时补充bug信息.
 - 2. chekclist平台的作用是多方公共拆解任务,执行任务,并且形成进度反馈.这是一个多用户同时编辑一个对象的场景,所以需要实时同步和广播的能力.
- 经过研究,最终选用的是mysql+sharedb+mongo的方案,下面我们简单介绍一下对比的方案

2.0 架构选型

2. 0.1 C/S架构的集中式设施

为所有用户提供文档编辑服务。所有用户都连接到一个中心服务器，该服务器负责存储和处理文档数据，用户通过连接到该服务器来协作编辑文档。提供更好的和可控性，但有单点故障问题

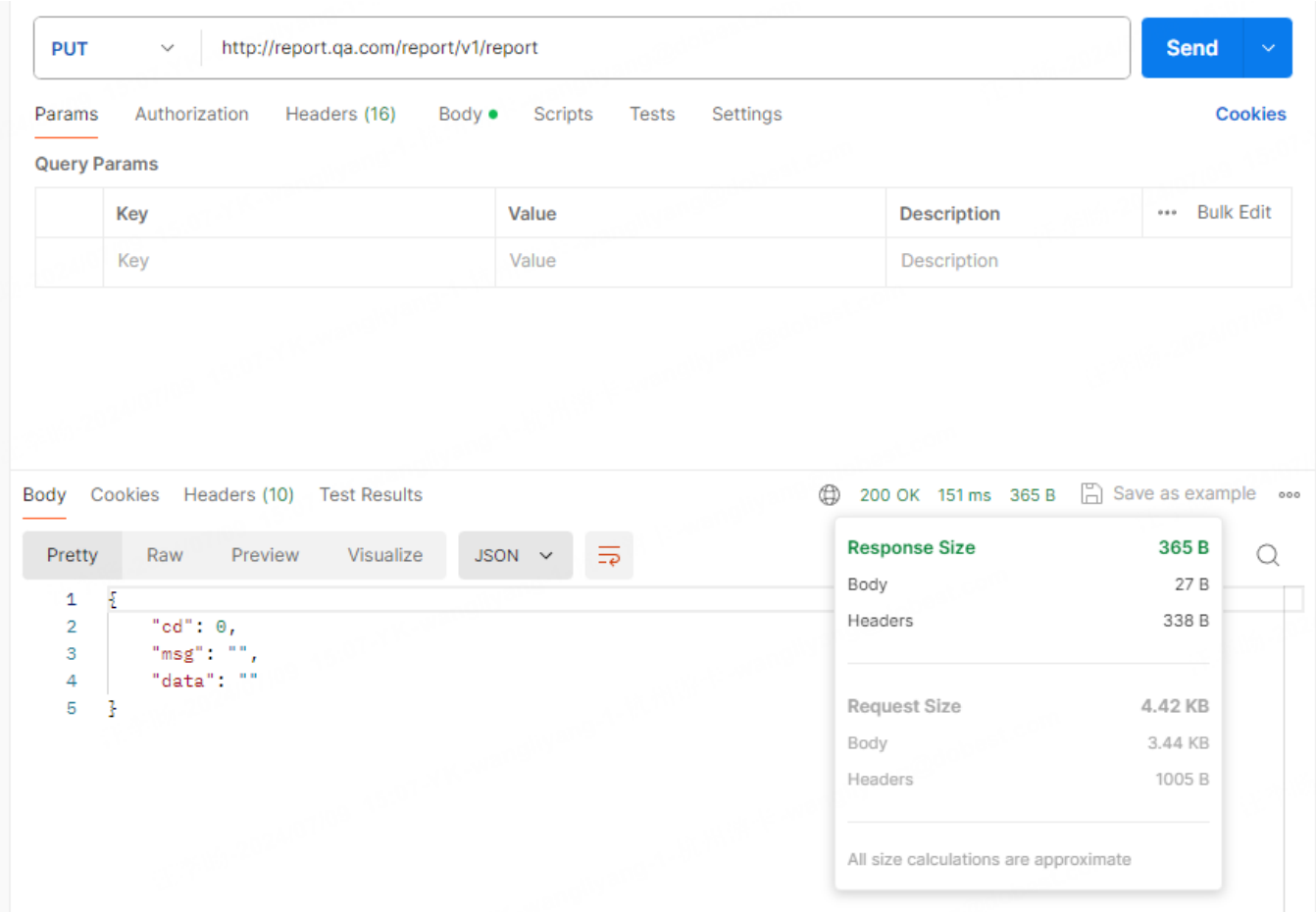
2.0.2 点对点技术设计

以便在单个文档上协作。将文档数据分散存储在多个用户设备，每个用户都可直接编辑文档并将更改同步到其他用户设备。提供更好灵活性和可扩展性，但可能会有数据同步不及时或数据冲突问题

2.0.3架构对比

	服务器负担	实时性	一致性	扩展性	兼容性
C/S	大	弱	强	强	强
P2P	小	强	弱	弱	弱

文档格式为json,更新大小约为5KB,小组成员数量约10人,预估最大总并发数据量:单个包体(5KB)上传(10)分发(10)=500KB



当前场景下总数据量较小,时延较低,且大多数商业方案侧重C/S架构, 以实现更精细控制。因此, 本文也讨论一致性扩展性和兼容性更好的C/S架构设计服务。

3.0主流产品调研

首先,我们应当明确协同编辑功能的特性.你不能只让每个人自己工作, 然后合并每个人的副本或选取最后一次编辑。用户实际上应该可以看到对方正在做什么, 并获得即时反馈。

3.0.1 Google Docs

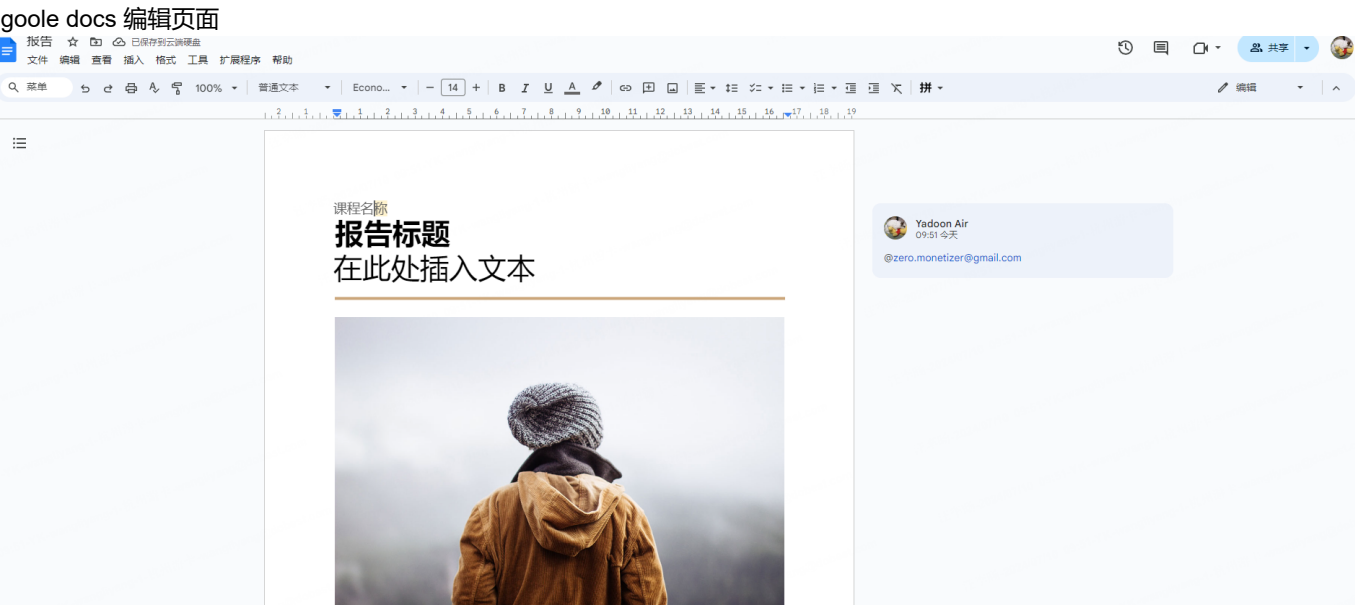
Google Docs 是基于云盘系统扩展的办公系统, 具有一系列功能, 包括文档存储, 共享, 格式化, 编辑等.系统有以下几个主要部分:

- 文件存储:由于 Google 文档是 Google 云端硬盘的一部分, 因此也包含了存储功能。该系统允许用户将文件（文档）分组到文件夹, 并支持编辑/创建/删除等功能。它更像一个操作系统。
- 在线编辑和格式化:毫无疑问, Google 文档的核心功能之一就是在在线编辑。它支持几乎所有的微软 Office 操作。
- 协同编辑:Google Docs 允许多个人同时编辑单个文档
- 访问控制:你可以与你的朋友分享文档, 并给予不同的权限（所有者, 只读, 允许评论等）。

Google docs的协同编辑基于版本控制实现,每次修改都会对文档创建一个版本,并显示diff,你可以查看文档的历史版本并选择保存.服务器可以为每个人保留量份相同的文档, 并跟踪完整的修订历史。当 A 通过在开头添加 x 来编辑文档时, 这个改变将与 A 所看到的最后修订一起发送到服务器。假设此时 B 删除最后一个字符 c, 并且这个改变也是这样发送到服务器。

由于服务器知道修改在哪个版本上进行，因此会相应地调整更改。更具体地说，B 的变化是删除第三个字符 c，它将被转换为删除第四个字符，因为 A 在开头添加了 x。

这就是所谓的操作转换（Operational Transformation）基本思想是根据修改和其他合作者的修改来转换每个人的改动。

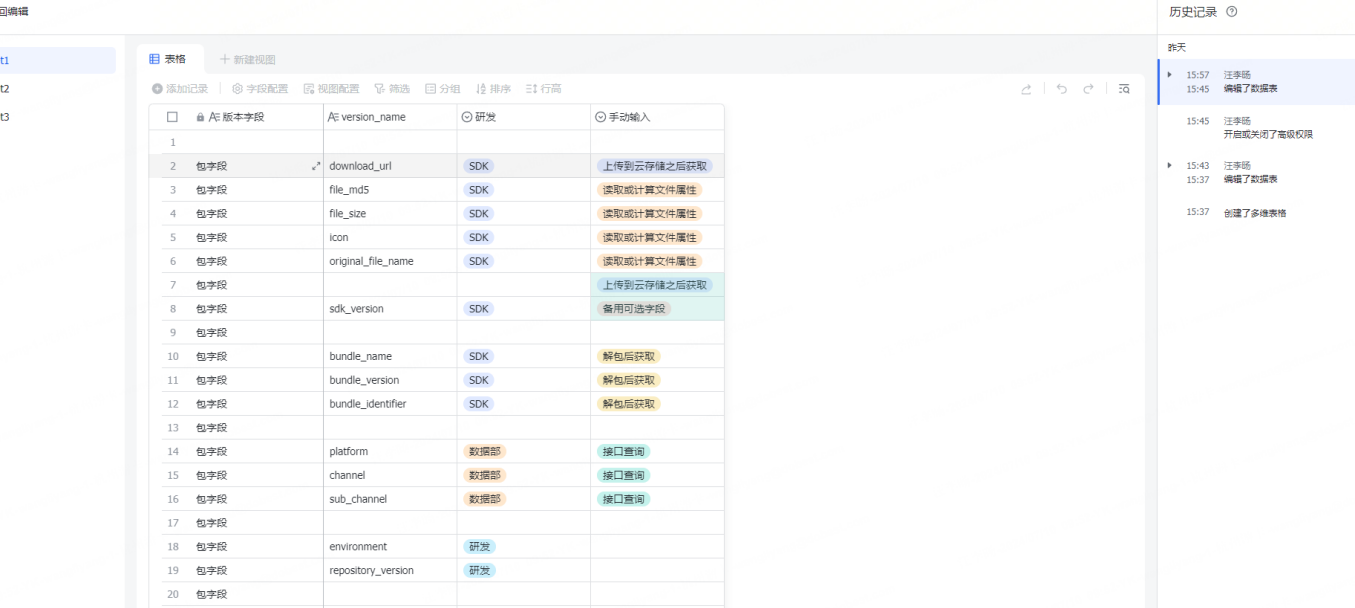


3.0.2 飞书云文档

飞书是一款国产综合性的办公协作平台产品,飞书云文档在前端设计简洁，取消了传统工具栏，采用隐藏式工具栏，提供简洁的白纸样式编辑界面，旨在让用户专注于内容创作,用户目前使用的产品就是飞书。

在协同编辑功能上使用了和Google Docs相似的云盘加OT方案,在持久化方面也同样使用了版本控制的方式存储文件和相关信息

飞书历史版本页面



4.0 协同冲突处理OT与CRDT

在线文档实时协同编辑的难点之一在于协同冲突处理,OT算法是解决协同冲突处理的主要方案.

4.0.1 编辑锁

当有人在编辑某个文档时，系统会将这个文档锁定，避免其他人同时编辑。编辑锁的实现方式简单粗暴，但会直接影响用户体验。

4.0.2 diff-patch

基于 Git 等版本管理类似的思想，对内容进行差异对比、合并等操作，包括 GNU diff-patch、Myer's diff-patch 等方案。diff-patch 可以对冲突进行自助合并，也可以在冲突出现时交给用户处理。

4.0.3最终一致性实现

包括 Operational Transformation (OT)、Conflict-free replicated data type(CRDT，称为无冲突可复制数据类型)。OT 算法是 石墨文档，腾讯文档，飞书文档，Google Docs 中所采用的方案，Atom 编辑器使用的则是 CRDT。

4.0.4 OT 和 CRDT

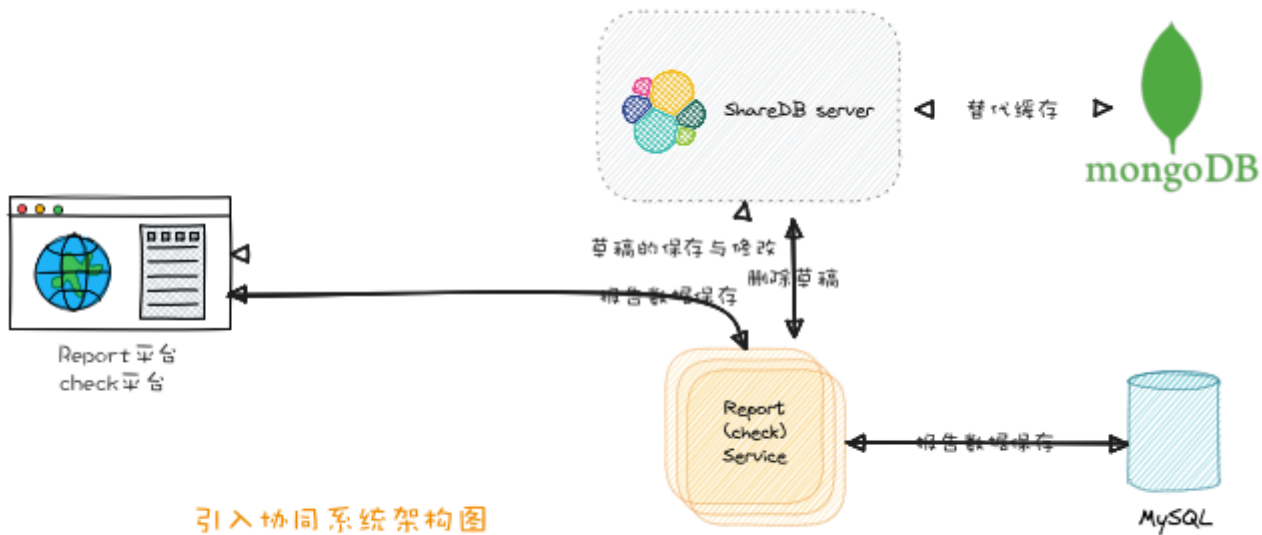
OT 和 CRDT 两种方法的相似之处在于它们提供最终的一致性。不同之处在于他们的操作方式：

- OT 通过更改操作来做到这一点
- OT 会对编辑进行操作的拆分、转换，实现冲突处理的效果
- OT 并不包括具体的实现，因此需要项目自行实现，但可以根据项目需要进行高精度的冲突处理
- CRDT 通过更改状态来做到这一点
- 基本上，CRDT 是数据结构，当使用相同的操作集进行更新时，即使这些操作以不同的顺序应用，它们始终会收敛在相同的表示形式上
- CRDT 有两种方法：基于操作和基于状态

对于富文本编辑等更高级的结构，OT 用复杂性换来了对用户预期的实现，而 CRDT 则更加关注数据结构，随着数据结构的复杂度上升，算法的时间和空间复杂度也会呈指数上升的，会带来性能上的挑战。因此，如今大多数实时协同编辑都基于 OT 算法来实现。

4.0 技术方案

4.0.1 总系统架构图



5.0 结果展示

以checklist平台对任务数据的协同编辑为例
用户A更新数据后,会提示数据同步到云端

CA Checklist平台 首页 模板管理

项目: !!!测试专用-品控工单 版本: 2.2

首页

责任人: 全部 执行人: 全部

名称	任务数	状态
测试协同	3	阻塞中

×

编辑CheckList

操作成功

已经保存到云端

* 名称: 测试协同 * 负责人: yangjian

* 截止时间: 2024-09-03

任务内容	描述	执行人	截止时间	操作
asdf	asdf11	yangjian	2024-09-03	编辑 删除
asdf22	改动222	yangjian	2024-09-03	编辑 删除
asd1f	asdf	ww	2024-09-03	编辑 删除

+添加任务

云端接收了数据后会广播信息,并提示其他用户是谁更新了数据

CA Checklist平台 首页 模板管理

项目: !!!测试专用-品控工单 版本: 2.2

首页

责任人: 全部 执行人: 全部

名称	任务数	状态
测试协同	3	阻塞中

×

编辑CheckList

管理员 更新了一个任务【asdf】

* 名称: 测试协同 * 负责人: yangjian

* 截止时间: 2024-09-03

任务内容	描述	执行人	截止时间	操作
asdf	asdf11	yangjian	2024-09-03	编辑 删除
asdf22	改动222	yangjian	2024-09-03	编辑 删除
asd1f	asdf	ww	2024-09-03	编辑 删除

+添加任务

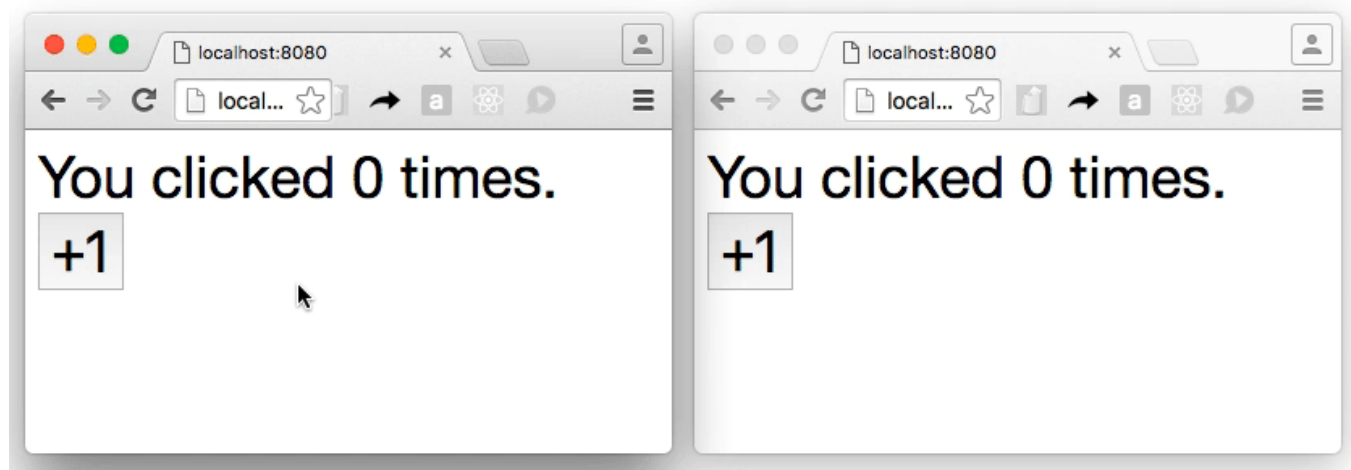
6.0 总结

在深入对比了当前市场上流行的协同编辑框架，包括 `sharedb` 和 `yjs` 之后，我们团队经过充分的技术评估和需求分析，最终决定采用 `sharedb` 作为我们项目的核心协同框架。这一决策基于以下几个关键方面的考量：

1. **生态系统与兼容性：** `sharedb` 建立在强大的 `ottypes`（Operational Transformation Types）和 `json0` 等协议之上，这些协议为文档的实时协同编辑提供了坚实的基础。此外，`sharedb` 与 `derbyjs` 等前端框架的良好集成，以及丰富的社区支持和插件生态系统，使得我们能够快速构建并扩展我们的协同应用。
2. **灵活性与可扩展性：** `sharedb` 的设计注重灵活性和可扩展性，它允许我们自定义文档类型、操作转换函数以及错误处理逻辑。这种灵活性使我们能够根据项目的具体需求调整和优化协同机制，确保系统的稳定性和性能。
3. **安全性与权限控制：** 在协同编辑环境中，数据的安全性和用户的权限控制至关重要。`sharedb` 提供了丰富的权限控制接口，允许我们精细地管理用户对文档的访问和操作权限。这有助于保护敏感数据，防止未授权访问和修改。
4. **性能与稳定性：** 经过实际测试，`sharedb` 在多种场景下均表现出良好的性能和稳定性。它能够高效地处理大量的并发操作和实时数据更新，确保用户之间的协作流畅无阻。这对于需要高实时性和高可靠性的协同应用来说至关重要。
5. **文档与社区支持：** `sharedb` 拥有详尽的文档和活跃的社区支持，这为我们在开发过程中遇到的问题提供了丰富的解决方案和参考资源。此外，社区中的专家和贡献者还不断为 `sharedb` 添加新功能和优化现有功能，使其始终保持与时俱进。

综上所述，`sharedb` 以其强大的生态系统、灵活的架构、可靠的安全机制、卓越的性能表现以及丰富的社区支持，赢得了我们团队的青睐。我们相信，在 `sharedb` 的支持下，我们能够打造出更加优秀、稳定且易于维护的协同编辑应用，为用户提供更加流畅和高效的协同体验。

Simple realtime client/server sync with ShareDB



参考链接

[checklist平台试用](#)

[shardb](#)

[sharedb协同可视化](#)

<https://www.cnblogs.com/WindrunnerMax/p/17114099.html>

<https://zhuanlan.zhihu.com/p/692480370>