



SRE 部門中的軟體工程實踐

Raixlai
2018/01/11

Google 幕後軟體工程實踐

- Google 許多不為人知的軟體來自 SRE 部門。
- Google 生產環境是人造最複雜的系統之一
- SRE 有豐富的運維經驗，非常適合開發內部工具解決運維問題
- SRE 開發的工具也是一個完整的軟體工程項目
 - 需要針對內部用戶的需求進行產品規劃
 - 制定未來發展的方向

為什麼軟體工程項目對 SRE 很重要

- 生產環境的複雜度導致需要 google 自己開發
- SRE 進行軟體工程非常適合和有效
 - 擁有 google 特有的生產環境建構知識
 - 需要設計出實現大規模部署, 災難處理, 和容易和基礎設施整合的工具
 - SRE 是工具的直接使用者, 了解開發工具的重點
 - 容易獲取直接和高品質的用戶回饋
- 團隊大小不應該與用戶服務規模成比例成長

Auxon 案例分析

Google SRE 內部開發的自動化容量規劃工具

傳統的容量規劃方法

- 收集對未來項目需求的預測
 - 使用擁有的數據來計劃明天
 - 預測長度一般是幾個季度到幾年
- 制定資源的採購，建構和分配計劃
 - 如何最好的滿足未來的資源需求？
 - 需要在“哪裡”建構“多少”資源？
- 評審，並且批准這個計畫
 - 計劃是否合理？
 - 是否符合產品期望？
- 部署和配置對應的資源
 - 哪些服務最終會使用？
 - 如何將資源合理分配？

傳統的容量規劃方法缺點

- 不可靠性
 - 服務出現效能下降, 需要更多的資源滿足業務需求
 - 用戶需求增加, 導致資源也隨之增加
 - 新的計算集群上線推遲
 - 產品的決策變化(ex. 備份從一份變兩份), 導致資源需求大幅增加
- 事倍功半
 - 收集足夠的數據以預測未來的需求是非常麻煩且容易出錯的
 - 將帶有限制的資源請求與實際可用的資源進行結合也是一個麻煩的過程
 - 使用的工具也常常是不可靠或很難用

解決方案：基於意圖的容量規劃

列出你的要求，而不要拘泥於具體實現細節

- 將服務的依賴和資源的參數用編程的方式記錄下來
- 利用演算法自動產生資源的配給方案
- 好處
 - 由電腦自動處理可以節省大量人力
 - 服務負責人可以專注在更高級的目標
 - 提供更高的計畫精確度，最終降低成本

基於意圖的容量規劃

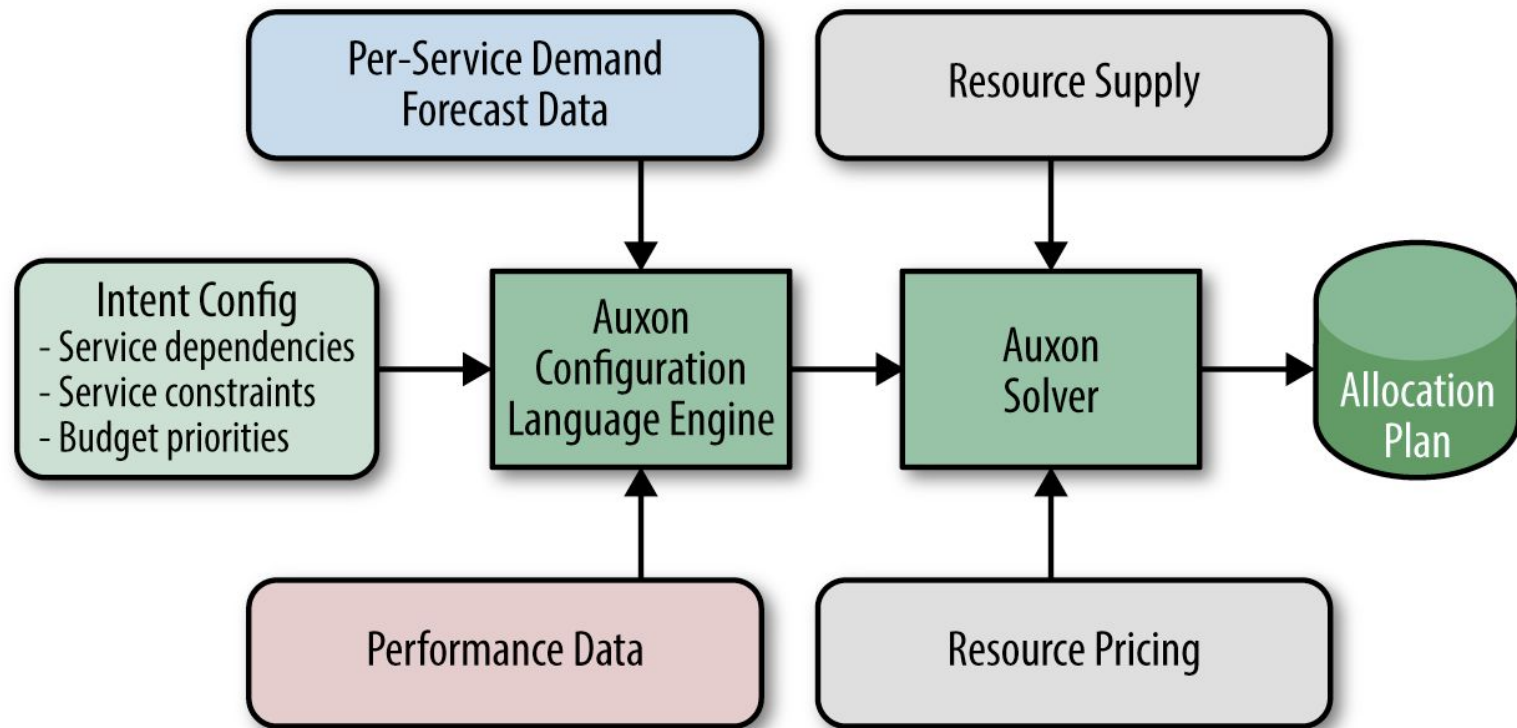
“意圖”是服務負責人對如何維運的一個理性表達

1. 我需要 50 個 CPU 資源，必須在集群 X, Y, Z 中，為服務 Foo 使用
 - 具體的資源請求
 - 為什麼需要這麼多資源？一定要指定的集群中？
2. 我需要 50 個 CPU 資源，在地理區域 YYYY 中任意三個集群中，為 Foo 使用
 - 增加更多自由度，更容易滿足，但為什麼要三個集群呢？
3. 我想要滿足 Foo 在每個地理區域的需求增長，同時保障 N+2 冗餘度
 - 有了更多自由度，同時更好理解如果沒有滿足需求，會有什麼後果
4. 我們想要將 Foo 以 99.99% 的可靠度運行
 - 最大的自由度，當容量得不到滿足的後果也最清楚。

表達產品意圖的先導條件

- 依賴關係
 - 服務依賴的其他服務和基礎設施會影響位置的選擇
- 性能指標
 - Foo 服務需要多少計算資源以服務用戶？
 - Bar 服務需要為 Foo 提供多少 Mb/s 的頻寬？
- 優先級
 - 在資源不夠的情況下，哪些資源請求可以被犧牲？

Auxon



需求和實現：成功和不足

Auxon 起源於一個 SRE 和一個技術項目經理。

- 任何一個足夠複雜的軟體工程項目都會遇到一定程度的不確定性。
 - 不要過於關注完美的解決方案，尤其當問題不夠清楚時
 - 應該更快的發佈和迭代。
- Auxon 一開始先建構簡單的邏輯來處理資源分配，讓團隊更有實際的體感。
- 不確定性不意味項目無法推進，可以促使開發更具通用和模組化的軟體
- 很多次的“發布和迭代”
- 要確保提供的解決方案有一個真實的使用案例。

提升了解程度，推進採用率

- 不要低估人們對產品了解的難度
- 向大型團隊推廣內部軟體工具需要以下幾點：
 - 持續和完整的推廣方案
 - 用戶的擁護
 - 資深工程師和管理層的協助
- 如果解決方案使用起來太困難，SRE 就會選擇自己開發

提升了解程度，推進採用率

- 設立期望值
 - 給產品設計一個最小成功條件，或是開發最小可行 產品 (MVP)
- 識別合適的用戶
 - 初期 Auxon 目標是沒有容量規劃流程的團隊
 - 早期的成功案例會變成工具的推廣者
- 客戶協助
 - 不要害怕在項目初期給新用戶提供協助，幫助他們更好的使用工具
- 在合適的層次上進行設計
 - 不可知性
 - 確保用戶不管有什麼數據型式，都可以採用 Auxon
 - 避免落入“100%”採用率的陷阱

團隊的組成

- 最好的團隊是合併通用型人才和領域專家
 - 通用型人才可以快速開始工作
 - 資深領域專家提供知識和經驗
- 多樣化的團隊可以避免設計盲點

在 SRE 團隊中培養軟體工程風氣

- 是什麼使 Auxon 可以演化為一個長期的軟體工程項目？
 - 相關領域專家表示興趣
 - 技術實力很強的用戶群體
 - 提供足夠明顯的優勢，減少 SRE 瑣事，優化現有的基礎設施
- 一個項目的目標應該與整個組織的發展目標一致
- 什麼樣的項目不合適？
 - 同時改變太多組件
 - 要求全有或全無的上線方式
 - 目標太大、太抽象，沒有足夠的實際案例來證明

招聘和開發時間

- SRE 通常是通用型人才
 - 廣度優先的學習方式使 SRE 對全局知識掌握很多
 - 缺少產品討論和挖掘需求的經驗
- 結合對使用者需求經驗豐富的 PM
 - 更好地培養一個兼顧 產品開發和實戰經驗的氛圍。
- 確保專注、沒有干擾的開發時間
- SRE 內部的主要軟體項目都是從一個小項目開始
 - 停留為某個工程師的草根項目，在業餘時間開發
 - 通過某種正式內部渠道成為一個正式項目
 - 從 SRE 主管層獲得認可，擴展成一個完整的軟體開發項目

達成這一點

- 創建並且宣揚一個明確的信息
 - 定義和宣揚戰略目標和計畫非常重要
 - 該軟體可以持久地產生加速作用
 - 減少同樣任務的執行方式
- 評估組織的能力
 - 需要找人來彌補 SRE 團隊中缺少的角色和技能
 - 通過內部招聘是個好策略
- 發布並迭代
- 不要降低標準
 - 抵抗走捷徑的誘惑
 - 問自己如果是其他團隊開發的, 會不會想採用
 - 可靠性很重要

結論

- 成功和失敗的項目為以後的項目鋪平道路
- SRE 驅動的軟體開發項目對公司建立可持續的維運團隊很有幫助
 - 不斷開發工具簡化流程或自動化
 - 不需要隨著服務的規模呈線性成長
- SRE 花在軟體開發的經歷最終會對整個公司、團隊以及個人得到回報