

행정업무 자동화 챗봇 개발 및 성능 향상에 관한 연구

박준수 · 김영준 · 정윤교*

공군사관학교

A Study on the Development and Performance Improvement of Chatbot for Office Automation

Junsoo Park · Youngjun Kim · Yoonkyo Jung

Republic of Korea Air Force Academy

E-mail : ykjung.rokafa@gmail.com

요 약

조직에서 이루어지는 반복적인 행정업무는 비효율적인 방법으로 진행되는 경우가 많다. 본 연구에서는 반복적인 작업을 자동화하기 위해 사용자의 접근성이 좋은 카카오톡 기반의 챗봇을 개발하고 실제 업무 환경에 사용했다. 그 결과 챗봇을 실제 운영하는 과정에서 여러 명이 동시에 접속할 때 서버가 다운되거나 응답이 되지 않는 문제를 확인했다. 이를 개선하기 위해 챗봇 서버에서 사용하는 프로그램의 언어를 변경하는 코드 고도화를 진행했으며, 데이터베이스 구조를 새로 설계하고 로드 밸런싱을 통해 서비스의 부하를 분산시키는 등 성능 향상을 위해 여러 방법을 시도했다. 각각의 방법이 얼마나 성능 개선에 영향을 주는지 확인하기 위해 초당 처리량, 평균 레이턴시 등의 지표를 통해 비교 후 챗봇을 실제 업무 환경에서 사용하면서 발생한 문제에 대한 개선방안을 제시했다.

ABSTRACT

Many office workers spend a lot of time performing repetitive office tasks in inefficient ways. We developed a user-friendly chatbot system based on Kakaotalk to automate repetitive tasks and used the chatbot in the real workplace. In the process of operating the chatbot, if several people use the chatbot at the same time, the server was down or could not respond. To address these issues, we performed code migration of programs used by chatbot back-end servers and tried several ways to improve server performance such as database redesign and load balancing. To determine how much each method affects performance improvement, we measured total request per second and average latency. After that, we proposed ways to improve the problems of using the chatbot in the work environment.

키워드

Chatbot, Office Automation, Performance Analysis, Golang, Database

1. 서 론

최근 정보화 기술의 발전에 따라 여러 기관에서 반복적인 업무를 자동화하고 사용자 편의성을 높이기 위해 노력하고 있다[1]. 하지만 이러한 노력에도 불구하고 행정업무가 비효율적으로 이루어지는 경우가 많다. 예를 들어 조직 구성원의 특정 정보를 조사하는 업무의 경우, 정보를 엑셀 파일에 입력하는 단순한 작업은 비효율적이고 많은 업무 소

요가 있다. 업무 부담을 줄이기 위해 조직원 모두가 볼 수 있는 게시판을 이용하게 되면 개인정보가 노출될 수 있으며, 구글 폼과 같은 도구들은 기능이 제한적이고 외부 서버에 정보가 저장된다는 단점이 존재한다. 따라서 이러한 문제를 해결하기 위하여 행정업무 자동화를 도울 수 있는 카카오톡 챗봇을 개발했다.

개발한 챗봇을 실제 운영하는 과정에서 다수의 이용자가 동시에 챗봇에 접속할 때 성능상의 문제로 응답을 하지 못하거나 서버가 마비되는 현상이 자주 발생하였고, 이에 대한 개선방안을 찾아 실용

* corresponding author

적이며 범용성이 높은 챗봇 시스템을 설계했다.

II. 챗봇 개발 및 성능 개선

2.1 연구 목표

본 연구에서 설계한 시스템은 서버 자체의 물리적인 사양을 높이는 방안은 고려하지 않으며, 프로그램의 최적화를 통해 서비스의 성능을 개선하는 것을 목적으로 한다. 카카오톡 챗봇 정책상 서버가 기준시간 이내에 응답이 없으면 요청에 대한 응답이 없는 것으로 간주하기 때문에, 다수의 이용자가 동시에 서비스를 이용할 때 초당 처리량과 레이턴시를 개선하는 것을 목표로 했다.

표 1. 기능 구현 요구사항

Index	Function Requirement
1	Chatbot has to identify user's name and authority.
2	Users have to be allowed to participate, check and cancel survey.
3	Administrator has to be allowed to make, check and delete survey.
4	Administrator has to be allowed to download csv file that contains results of survey.

챗봇의 기능 구현 요구사항은 표 1과 같다. 챗봇은 사용자의 이름과 권한을 확인하고, 일반 사용자와 관리자의 기능을 구분하여 설계해야 한다.

표 2. 성능 개선 요구사항

Index	Performance Requirement
1	Chatbot has to reply all request at least 100 per second.
2	Chatbot has to reply all request at least in 1 second under situation of 100 concurrent users.

표 2는 실제 업무 환경을 바탕으로 작성한 성능 개선 요구사항이다. 챗봇은 초당 100개의 요청에 응답하고, 동시접속자가 100명인 경우에도 1초 이내에 응답할 수 있어야 한다.

2.2 시스템 설계

챗봇을 만들기 위해 카카오톡 채널 서비스 생성 및 카카오 i 오픈빌더[2] 설정을 진행 후 챗봇 서비스를 실행할 서버와 연결했다.



그림 1. 챗봇을 사용한 사용자의 요청 및 응답 과정

사용자의 요청 및 응답 과정은 그림 1과 같다. 사용자가 카카오톡을 통해 챗봇 채널로 메시지를 보내면 카카오톡 채널을 통해 챗봇 서버로 사용자의 ID, 발화 등의 정보가 담긴 JSON 형식의 데이터가 전송된다. 챗봇 서버는 이를 처리하여 다시 JSON 형태로 응답을 보내면 카카오톡 채널이 사용자에게 답장을 보내는 형태이다.

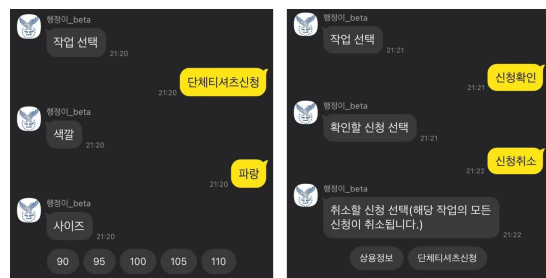


그림 2. 사용자 기능 예시 화면

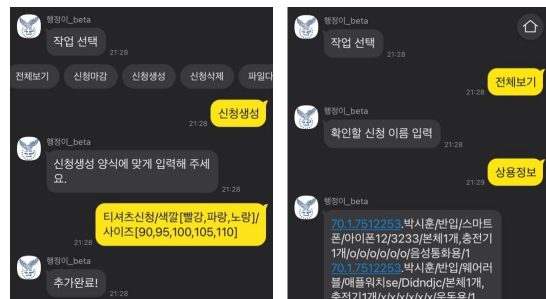


그림 3. 관리자 기능 예시 화면

주요 기능은 사용자와 관리자의 기능을 구분하여 구현했다. 먼저 사용자는 그림 2의 예시처럼 챗봇에 등록된 설문조사에 참여하고 이를 확인 및 취소할 수 있다. 관리자는 그림 3의 화면과 같이 직접 조사를 생성하고 삭제할 수 있으며, 전체 신청 내역 확인과 파일 다운로드가 가능하다.

2.3 성능 개선방안 및 비교

초기에 제작한 챗봇을 800여 명의 사용자를 대상으로 운용하는 과정에서 동시접속자가 많아질수록 응답이 지연되고 서버가 마비되는 문제가 발생했다. 따라서 성능을 개선하기 위해 코드 고도화, 데이터베이스 설계, 로드 밸런싱을 적용했고 각 성능을 측정하여 결과를 비교했다. 성능 확인은 요청 메시지와 동일한 JSON 형식의 POST를 반복적으로 보내면서 오픈소스 성능 테스트 도구인 locust를 사용하여 초당 처리량과 응답시간을 측정했다.

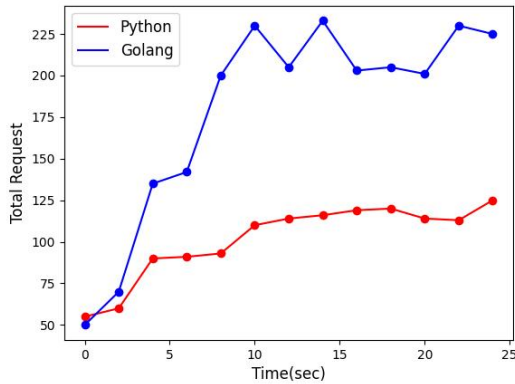


그림 4. 코드 고도화 전·후 초당 처리량 비교

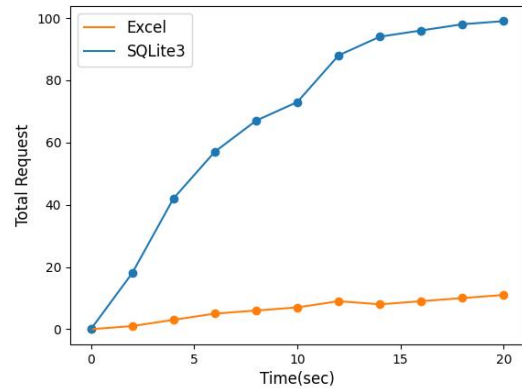


그림 6. 데이터베이스 설계 전·후 초당 처리량 비교

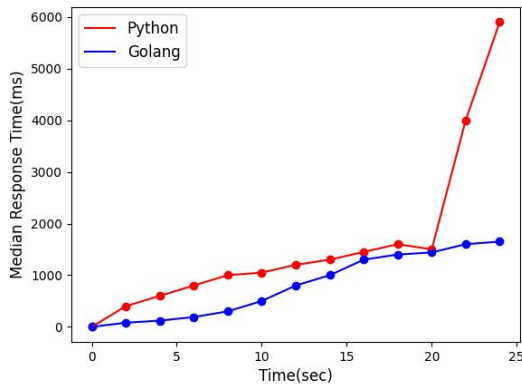


그림 5. 코드 고도화 전·후 평균 응답시간 비교

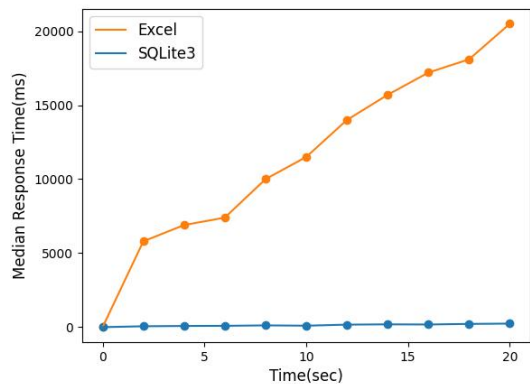


그림 7. 데이터베이스 설계 전·후 평균 응답시간 비교

그림 4, 5는 챗봇 서버 프로그램의 언어 변경을 통한 코드 고도화 결과를 보여준다. 기존에는 개발의 편의성을 위해 Python3와 Flask를 이용하여 서버를 개발하였으나, 성능 개선을 위해 HTTP 서버 성능이 더 좋은 Golang의 net/http 모듈을 이용하여 서버를 다시 개발하였다. 발화 내용을 반복하여 응답하는 프로그램을 Python과 Golang으로 작성하여 비교한 결과 동일 환경에서 Golang으로 작성한 프로그램이 초당 처리량은 2배 이상 많고 응답 시간은 최대 4배까지 줄어드는 것을 알 수 있다.

그림 6, 7은 데이터베이스 설계 후 성능 향상 결과를 보여준다. 기존에는 사용자가 신청 내역을 간단히 확인할 수 있도록 데이터를 엑셀파일로 저장했으나, 데이터에 접근하는 과정에서 레이턴시가 증가하여 SQLite3로 데이터베이스를 변경하였다. 사용자로부터 메시지를 받았을 때, 사용자의 이름을 답장해주는 프로그램을 데이터베이스와 언어만 다르게 하여 개발한 결과 SQLite3를 사용하였을 때 초당 처리량은 40배 이상 많고, 응답 속도 또한 100배 이상 빠르다는 사실을 알 수 있다.

또한 로드 밸런싱을 위해 한정된 서버 자원에서 여러 개의 백엔드 프로그램을 구동시켰으나 단일 서버에서는 성능 개선에 큰 영향이 없었다.

III. 결 론

본 연구는 반복적인 작업을 자동화하기 위해 챗봇을 설계했고, 실제 운용 후 성능 개선 과정을 통해 연구 초기에 설정하였던 4가지 기능, 2가지 성능 개선 요구사항을 모두 충족했다. 성능 개선 방법으로 사용한 코드 고도화와 데이터베이스 설계는 큰 효과가 있으나 로드 밸런싱은 성능에 큰 차이가 없었다. 실제 업무 환경에서 사용자와 관리자 모두 문제없이 사용했으며, 사용자 응답에 대한 데이터베이스 암호화 등의 작업은 추가로 필요할 것으로 예상된다.

References

- [1] H. Y. Yeen, Y. H. Shin, W. J. Ko, S. J. Lee, and M. W. Koo, "Coversational major recommendation chatbot that allows general conversation and major recommendation," in *Proceeding of HCI Korea 2022*, pp. 764-767, 2022.
- [2] Kakao i Open Builder. Available: <https://i.kakao.com/>.