

CASE STUDY

Dark Launch of Facebook Chat (2008)

For nearly a decade, Facebook has been one of the most widely visited Internet sites, as measured by pages viewed and unique site users. In 2008, it had over seventy million daily active users, which created a challenge for the team that was developing the new Facebook Chat functionality.^{34†††}

Eugene Letuchy, an engineer on the Chat team, wrote about how the number of concurrent users presented a huge software engineering challenge, “The most resource-intensive operation performed in a chat system is not sending messages. It is rather keeping each online user aware of the on-line-idle-offline states of their friends, so that conversations can begin.”³⁶

Implementing this computationally intensive feature was one of the largest technical undertakings ever at Facebook and took almost a year to complete.^{†††} Part of the complexity of the project was due to the wide variety of technologies needed to achieve the desired performance, including C++, JavaScript, and PHP, as well as their first use of Erlang in their back-end infrastructure.³⁷

Throughout the course of the year-long endeavor, the Chat team checked their code into version control, where it would be deployed into production at least once per day. At first, the Chat functionality was visible only to the Chat team. Later, it was made visible to all internal employees, but it was completely hidden from external Facebook users through Gatekeeper, the Facebook feature toggling service.

As part of their dark launch process, every Facebook user session, which runs JavaScript in the user browser, had a test harness loaded into it—the chat UI elements were hidden, but the browser client would send invisible test chat messages to the back-end chat service that was already in production, enabling them to simulate production-like loads throughout the entire project, allowing them to find and fix performance problems long before the customer release.

The Chat release and launch required only two steps: modifying the Gatekeeper configuration setting to make the Chat feature visible to some portion of external users, and having Facebook users load new JavaScript code that rendered the Chat UI and disabled the invisible test harness. If something went wrong, the two steps would be reversed.

When the launch day of Facebook Chat arrived, it was surprisingly successful and uneventful, seeming to scale effortlessly from zero to seventy million users overnight. During the release, they incrementally enabled the chat functionality to ever-larger segments of the customer population—first to all internal Facebook employees, then to 1% of the customer population, then to 5%, and so forth. As Letuchy wrote, “The secret for going from zero to seventy million users overnight is to avoid doing it all in one fell swoop.”³⁸

In this case study, every Facebook user was part of a massive load testing program, which enabled the team to gain confidence that their systems could handle realistic production-like loads.