Westat®

Improving Lives Through Research®

# Westat Scaling Experience

Mangal Subramanian, Arthur Menis, Kathleen O Reagan

September 15, 2020

# Introduction

- This paper is about our experiences fielding a large scale web survey.
- The project consisted of a Web Survey and  a CATI follow-up.
- Web survey had two instruments, a screener and an extended.
- The Sample Size was approximately 360k.
- Westat's role was focused on sampling, data collection, and weighting

# Blaise Instrument Characteristics

- The instrument was programmed in Blaise 5 version (5.6.5.2055)
- The screener and extended surveys were programmed as one to allow for a smooth transition for the user from screener to extended.
- Randomized questions - did not have the option to implement the new randomization feature in Blaise 5.7
- Multiple questions on the same page. - Had to set up with individual templates.

# Performance Requirements

- 2500 users accessing survey each day.
- Expectation of 400 concurrent users.
- Only 10% of the sample was expected to qualify for the extended.
- Page load times of <= 3 seconds

# Management Application

- In-house management system that managed loading and launching of cases.
- Token based security system to ensure secure launching and completion of the instrument.
- The management system also handled authentication, using pins.

# Initial Load Testing Experience

- Visual Studio Web load testing tool combined with Azure load testing service.
- VS load testing tool didn’t Work!
- Custom test script to be created to handle the ‘sessionID’ parameters during the automated Test script runs.
- SN team gave us a custom script, but we did not have the time to try it out and get it to work.

# Our Solution to the Load Testing Problem

- Low tech, Low risk and dependable testing tool - Human Testers!
- Only the screener had to handle the large load.
- Around 70 total testers
  - Each Tester was provided 5-6 cases and asked to complete at least 3
  - Management application was not used.
- Performance Profiling on the Web and DB servers.

# So What Did the Load Test Reveal…….?

# Manual Load Test Results

- A total of 333 completes were validated in the Blaise DB.
- This included mostly screeners, and a handful of extended interviews.
- Server performance monitors were happy.
  - No CPU spikes, indication of memory leaks etc.
  - DB profiling, showed no issues like slow queries, locks etc.

# Manual Load Test Results

- A total of 333 completes were validated in the Blaise DB.
- This included mostly screeners, and a handful of extended interviews.
- Server performance monitors were happy.
  - No CPU spikes, indication of memory leaks etc.
  - DB profiling, showed no issues like slow queries, locks etc.

# Issues Found During the Load Test

- One page was slow to load for all users - it contained multiple questions where question order was randomized.
- After consulting with Blaise team, we learned that the additional load time was due to the client rendering time.
- Slow page had a layout with several enumerated type questions

# Load Test Results

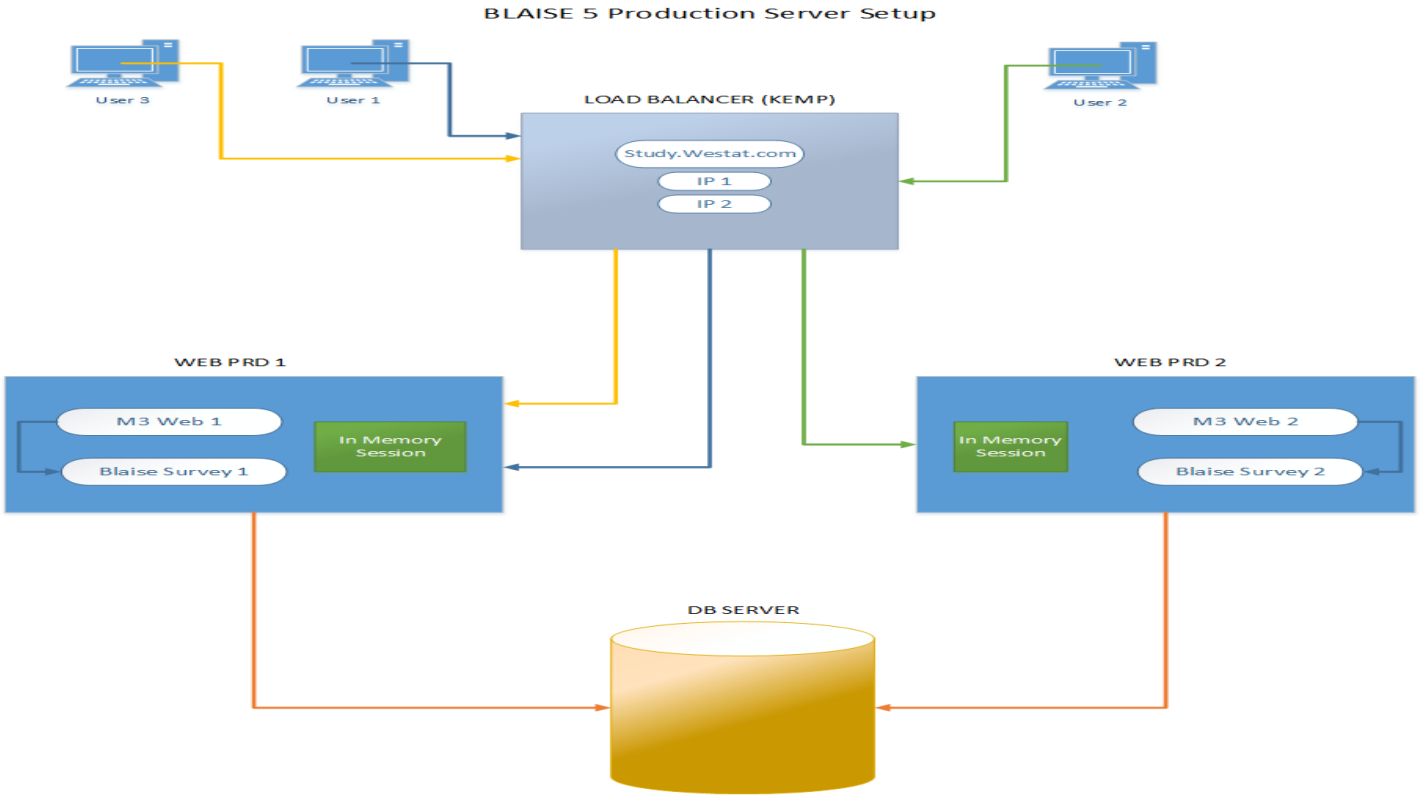
- 300 + completes in less than an hour with a single server configuration.
- The response times were less than 2 seconds. (for most of the pages)
- Actual production load will probably be less intense.

"I hope the jumper cables work. I've got to start her up and finish my report."

# So Now What?.... Making Sense of the Results

- Dedicated Virtual Windows Server for the Database.
- Two Windows application servers running IIS.
- Load balancing done with a KEMP network device.
- Quad Core Servers with 32GB of memory.
- Windows 2016 OS

# Server Set-up Diagram



# Blaise 5 Installation Details

- Each application server was configured with a serverpark with all the roles.
  - Management, Audittrail, Data, DataEntry, Resource, and Session.
- A local server (logical server) was created to allow the serverpark to be mapped to the website and the physical location of the installed survey

# Production Set-up Validation

- Need to validate if the production server set-up was working properly.
- Blaise look-up performance validation, related to security implementation.
  - Blaise search (TokenModel.SEARCH(pToken))
  - Blaise Read (TokenModel.READ).
  - To test this we prefilled the table with 200k records and manually tested running cases – No slowness in response times were found.
- Initial smoke test went off smoothly.

# How did we fare…..Did our efforts pay off!

- Overall the project implementation was a big success.
- Very few issues reported.
- During peak load times, we got a maximum of 2400 completes in a day.
- Around 200 completes an hour during peak load.

# Can We Better This?

- One consideration is to adopt the recommended server park set-up from the Blaise team.
  - Application Server-1 – Web, Data Entry, Resource Roles
  - Application Server-2 – Web, Data Entry, Resource, Session, Audit Trail, Data.
  - Database Server.
- Important Configuration - machineKey should be the same for both servers in a multiserver park.
- The scaled up two server configuration better manages sessiondata than two independent servers.

# Takeaways

- The Blaise 5 architecture definitely seems capable of handling large scale applications.
- Its distributed Services architecture lends itself to set-up a multi server environment.
- The Blaise team’s support through this process was invaluable.
- Some areas that we need further collaboration & discussions are.
  - Better handling of layouts.
  - More clarity around Record locking and best practices for using it.
  - Standardized approaches to support automated testing tools.

# Future Considerations & Conclusion

- The ability to run Blaise on Cloud will open up lot of possibilities for scaling.
- Containerizing Blaise - This would be a game changer, if we as a community can figure this out.

# Thank You

*See You At the Next IBUC Without a Mask*

Contact Information;


- Mangal Subramanian – YOMan@Westat.com
- Arthur Menis – ArthurMenis@Westat.com
- Kathleen O Reagan –KatOi@Westat.com