Too Big to Test:
Breaking a production brokerage platform without causing financial devastation

Kyle Parrish
Director
Performance Architecture
Fidelity Investments
Kyle.Parrish@fmr.com

Dave Halsey
Vice President
Performance Testing and Engineering
Fidelity Investments
David.Halsey@fmr.com
Agenda

- Who we are
- How we got into this
- The planning process
- Building the beast
- Running tests
- What we have learned
- Where we are heading
- Questions
First, A little About us…

- We both …
  - Have backgrounds in consulting
  - Bring different skill sets and personalities to projects – which is a really good thing
  - Work at Fidelity Investments in performance engineering and capacity management
  - Have been involved in the full-scale, end-to-end testing of brokerage systems for about 5 years

#velocityconf
A little bit about where we work - Fidelity Investments

➢ Our systems support over...

➢ 24 million individual investors
➢ 20.0 million brokerage accounts
➢ 506,000 commissionable trades per day
➢ $5.4 trillion in total customer assets

➢ Our technology groups support...

➢ 10 major businesses, all relying on shared infrastructure
➢ Two major data centers, each with massive mainframes
➢ Many different front-end channels
➢ Strong regulatory and compliance controls

#velocityconf
Disclaimer

- All of that being said…

Nothing we say or show during this presentation represents the opinions of our employer. Everything said here results from our own individual experiences and development of best practices throughout our careers in IT.

- So sayeth us….
The History
Where were you on the mornings of...

- August 24th, 2015?
- On May 18th, 2012?
- May 6th, 2010?
- All of September of 2008?
- Black Monday, 1987?
- October 24th, 1929?
They were not “good” days...

Dow plummets as much as 1,000 pts.

- China’s ‘Black Monday’: 2015 stock-market gains swamped by panic
- MARKET EXTRA: Exactly how panicky should we be about China?
And they will not be the last…

“A recent spur in volatility across the global equity market is driving system failures at major financial institutions.”
The Bottom Line…

- There is no “IF” when discussing the possibility of another significant market event…only when

- The question is…

Can we be prepared when it happens?

We believe we can and this is what we are here to talk about

#velocityconf
The Event
Big events evoke big changes, and the Flash Crash of May 6, 2010 raised some eyebrows.

*The market dropped 600 points in a span of minutes*

Finding a new way to test was born out of this event, and has only been reinforced by events that followed.

It was a wakeup call - While technology is creating great opportunity, at the same time, it is creating great risk.
On May 7th, we came to work and the management asked....

“What are YOU going to do to protect our clients?”
The Approach
What is the problem we are trying to solve?

- May 6\textsuperscript{th} proved that...
  - Market spikes can happen at any time
  - All systems and all channels need to be flexible enough to adapt to those changing conditions
  - We could not simulate possible scenarios with the current testing protocols
  - Management wanted to pinpoint weak spots in order to mitigate the risk before any events and was, at the time, unable to identify those risks

So, we needed to design a test protocol that...

- Found application and infrastructure limits anywhere throughout the system
- Enabled true end-to-end testing
- Could repeat market open events
- Would stress the system using many different market scenarios
- Had the veracity of production and the flexibility of statistical models
We began thinking…

- Should we build a new environment?
- Should we work with a vendor?
- Should we repurpose an existing test environment?
- How much data do we need?
- How do we make it realistic?
- How do we test like our customers work?
And, what are we actually dealing with?

System Under Test
Ultimately, we settled on DR+prod

- DR provides a production scale mainframe with all data and applications
- Production is a “routed-out” segment of true production front-ends for a site
- Customers can be simulated from the cloud, getting closer to the real user experience
- We can performance test End-to-End!
- Tests simulate real production customer patterns and volumes on production systems
- Tests are repeatable!
Using the combination of DR+Prod creates opportunity

- Test at x times prod scale
- Replay production traffic
- Recreate windows of production activity
- Build repeatable test patterns/workload profiles
- Ensure veracity of tests through use of production data and application
- Break things with impunity – no messy cleanup!
The Process
So how do you make this happen?

Ultimately, it always boils down to the same three things…

*Unsolicited Observation – people are the hardest part…*
First, we needed a team...

- Small number of specific resources
- Key characteristics
  - Firm knowledge
  - Process knowledge
  - Data knowledge
  - Executive presence
  - Assertiveness
  - Team oriented, but with independent spirit
- Fearless of failure
And an approach... no templates existed

The complexity of the End-to-End testing requires that the team adopt a clear process for analyzing, planning, designing, and delivering the test models, data, and execution plans.
We needed to understand the systems and processes involved

Brokerage systems are much more complex than the average application stack, and often have code and processes which have existed for decades.

Our testing requires deep process understanding
We needed to understand the data and entities

User Types
- Retail
  - .Com - user ID pass
  - Frequent Traders - user ID pass
  - Others - ??
- Brokers/Dealers
  - Broker - user ID pass
  - Administrator
  - Client - user ID pass
- Advisors
  - Channel1 - user ID pass
  - Channel2 - user ID pass
  - 3270 Manual - user ID pass

Security
- Web Challenge Response
  - PIN/User ID
  - Security questions
  - Password
  - Account mappings
- 3270
  - UserID
  - password
  - TSO/Top Secret permissions

Middleware
- Account number
- Requestor ID
- View Request
- Physical Routing

Internal Trade Routing
- Product/Securities Info
  - CUSIP
  - Symbol
  - Pricing
  - Update process
- Corporate Actions
  - CUSIP and Action
- Streaming Data/Quotes
  - Symbols/CUSIPS
  - Reuters
  - Activ Financial
  - Exchanges

Mainframe Applications
- Order Management
  - Account
  - Requestor ID
  - Security
  - View
- Customer Name and Address
  - Account
  - Customer
- Functional Applications
  - View
  - Conditional orders
  - Market data

Exchanges
- Symbols
- Prices
- Updates/changes

Institutions

Brokerage Mainframe
- CICS, WAS on Z, zLinux, VSAM, DB2, COBOL, Java, etc

Downstream Systems
- Solaris, AIX, WebSphere, TomCat

Market Data
- Streaming
  - Quotes
  - News/Charts
- Alerts
  - Updates to users

Clearing
- Account number
- Intraday positions

Custody and Clearing

Front End Apps

#velocityconf
While always assessing risk

When testing in Production…

Risk management is KEY…

1. Preparation of transaction loads for the load generation scripts will require that all transactions are submitted to clearly identified securities which are designated as test only securities
2. Physical and system access to the load generation data and facilities will be restricted to team involved directly in the test
3. Test transactions will only be submitted through users and into accounts which are clearly identified as test only users and accounts
4. Test users and test accounts will only be attached to specific branch codes designated as test only branches
5. The only securities which will be included in any transaction during test are those securities created/assigned specifically to the test
6. External reporting and financial impacts will be identified and isolated, and tested prior to test
Navigating the enterprise presents challenges

- Three different brokerage groups and all of their friends
  - Technology
  - Account Setup
  - Firm Setup Team
  - Deployment
  - Product Management
  - Risk and compliance
  - Functional, Integration, and Performance QA
- Corporate Risk and Compliance
- Clearing Services
- Order Management
- Name and Address

- Capital Markets
- Security Master/Product Master
- Fees and Commissions
- Electronic Notifications
- Corporate Actions
- Streaming Data and News
- Market Data
- Operations
- Printing and reporting
- Offshore QA Teams
- Purchase and sale/Books and Records
- Security
- Third Party providers

#velocityconf
We needed help from a lot of people…

Participation by Business Group

Participants by Division

330+ Unique Test Participants since 2011
Building the test
Systems are the vehicle, but data is the fuel...

- Approximately 110,000 user IDs
- Approximately 300,000 investment accounts
- Synthetic marketplaces
- UserID and account combinations to cover multiple business channels
- Full coverage of all registration/account types
- Creation of very diverse account holdings
Data needs to be accurate and SAFE, so masking DR is critical

DR is a copy of real customer data

We use that, +masking to deliver realistic, YET Safe, users and accounts

### PRODUCTION

<table>
<thead>
<tr>
<th>RealCustID</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealCustID1</td>
<td>$1,097,230</td>
<td>5</td>
<td>36</td>
<td>220</td>
<td>John Doe</td>
<td>30 Elm St</td>
</tr>
<tr>
<td>RealCustID2</td>
<td>$10,234,400</td>
<td>22</td>
<td>135</td>
<td>600</td>
<td>Jane Doe</td>
<td>20 Maple St</td>
</tr>
<tr>
<td>RealCustID3</td>
<td>$65,340</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>Tom Smith</td>
<td>40 Main St</td>
</tr>
<tr>
<td>RealCustID4</td>
<td>$113,434</td>
<td>2</td>
<td>5</td>
<td>32</td>
<td>Sue Jones</td>
<td>10 River St</td>
</tr>
</tbody>
</table>

### DR

<table>
<thead>
<tr>
<th>RealCustID</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealCustID1</td>
<td>$1,097,230</td>
<td>5</td>
<td>36</td>
<td>220</td>
<td>John Doe</td>
<td>30 Elm St</td>
</tr>
<tr>
<td>RealCustID2</td>
<td>$10,234,400</td>
<td>22</td>
<td>135</td>
<td>600</td>
<td>Jane Doe</td>
<td>20 Maple St</td>
</tr>
<tr>
<td>RealCustID3</td>
<td>$65,340</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>Tom Smith</td>
<td>40 Main St</td>
</tr>
<tr>
<td>RealCustID4</td>
<td>$113,434</td>
<td>2</td>
<td>5</td>
<td>32</td>
<td>Sue Jones</td>
<td>10 River St</td>
</tr>
</tbody>
</table>

### Masking batch jobs

<table>
<thead>
<tr>
<th>Num</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>S3PEED 1</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>2</td>
<td>$10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>S3PEED 2</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>3</td>
<td>$10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>S3PEED 3</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>4</td>
<td>$10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>S3PEED 4</td>
<td>1 SPEED ST</td>
</tr>
</tbody>
</table>

x100,000

etc.
"Splicing" allows us to safely look like customers

Real DR copies real customer data, so we wipe it

But even fake users need liquidity, so we provide it

```
<table>
<thead>
<tr>
<th>Real CustID1</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealCustID</td>
<td>$1,097,230</td>
<td>5</td>
<td>36</td>
<td>220</td>
<td>John Doe</td>
<td>30 Elm St</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RealCustID</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealCustID1</td>
<td>$65,340</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>Tom Smith</td>
<td>40 Main St</td>
</tr>
<tr>
<td>RealCustID1</td>
<td>$113,434</td>
<td>2</td>
<td>5</td>
<td>32</td>
<td>Sue Jones</td>
<td>10 River St</td>
</tr>
</tbody>
</table>
```

Masking batch jobs

“splicing” batch job assigns cash and positions to fake users

```
<table>
<thead>
<tr>
<th>DR</th>
<th>Portfolio $</th>
<th>Acct</th>
<th>Pos</th>
<th>Trades/yr</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>FakeUser1</td>
<td>$1,097,230</td>
<td>5</td>
<td>36</td>
<td>220</td>
<td>S3PEED1</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>FakeUser2</td>
<td>$10,234,400</td>
<td>22</td>
<td>135</td>
<td>600</td>
<td>S3PEED2</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>FakeUser3</td>
<td>$65,340</td>
<td>2</td>
<td>12</td>
<td>10</td>
<td>S3PEED3</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>FakeUser4</td>
<td>$113,434</td>
<td>2</td>
<td>5</td>
<td>32</td>
<td>S3PEED4</td>
<td>1 SPEED ST</td>
</tr>
<tr>
<td>x100,000</td>
<td>$10,000</td>
<td>5</td>
<td>36</td>
<td>220</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

#velocityconf
Additional Key Processes

- Capture replay data
- Process data to the load drivers
- Establish all security perimeters
- Build the DR environment

#velocityconf
Running the test
The final test platform integrates Prod, DR, and Test...
We can test in “production” and never impact customers

[Diagram showing test and production environments with arrows indicating connectivity between Test/Prod Middleware, DR Mainframe (Masked Accounts), Prod Site 1, Test, Prod Site 2, Prod Middleware, Prod Site 3, Clients, and Production Mainframe.]
The Results

#velocityconf
And testing leads to fixes before our customers find them…

1. Load Balancer bottleneck
2. Firewall limit of 1 Gbps found
3. Application Server CPU saturated at 1350 TPS
4. Slow roundtrip from a transaction slowed a related transaction at 1600 TPS
5. Contention for 15 slots by 18 hosts – Fixed
6. String Wait on VSAM file
7. Transaction error consuming Alerts
And because of these tests, there has been much joy!!

The Ultimate Result:
We quickly adapt to changing conditions with relatives ease and know our systems limits.
But there is always more to do, as management is still asking…

- When are you going to…
- Build pre-prod testing
- Design selectable workloads
- Improve infrastructure alignment
- Include additional channels
- And a bunch of things that they have not yet imagined
Takeaways

“*It is not the strongest of the species that survives, nor the most intelligent. It is the one that is most adaptable to change*“. Leo Megginson

- Testing like this takes patience. It is a journey, and a long one at that.
  - This is not a “I wrote a script and ran a test” scenario
- Relationships are critical
  - To be successful, you need help from a lot of people
- You need to be ok with failure
  - In complex testing, with dynamic requirements and targets, you will fail
    - The upside – you will learn an amazing amount
- There are many ways to “skin a cat”
  - Be creative – Stretch your imagination – Try to have fun!

#velocityconf
Questions?
Thank you!