Microsoft Research

Microsoft

Agile Co-Design for a Reconfigurable Datacenter

Shlomi Alkalay, Hari Angepat, Adrian Caulfield, Eric Chung, Oren Firestein, Michael Haselman, Stephen Heil, Kyle Holohan, Matt Humphrey, Tamas Juhasz, Puneet Kaur, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Andrew Putnam, Raja Seera, Rimon Tadros, Jason Thong, Lisa Woods, Derek Chiou, Sitaram Lanka, Doug Burger

Acknowledgments: CSI Team, Altera
FPGA'16 Designer Session

Microsoft Research

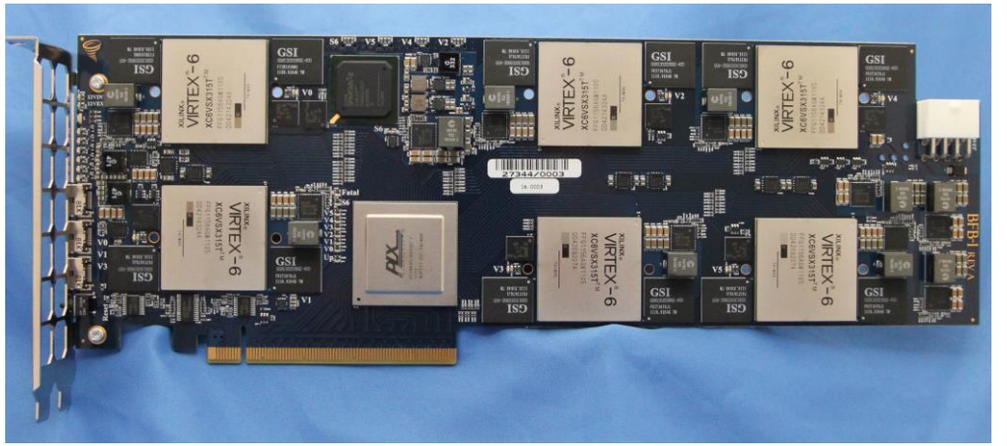
Microsoft

Or: 8 Requirements for Deploying and Developing Algorithms for FPGAs at Scale

Shlomi Alkalay, Hari Angepat, Adrian Caulfield, Eric Chung, Oren Firestein, Michael Haselman, Stephen Heil, Kyle Holohan, Matt Humphrey, Tamas Juhasz, Puneet Kaur, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Andrew Putnam, Raja Seera, Rimon Tadros, Jason Thong, Lisa Woods, Derek Chiou, Sitaram Lanka, Doug Burger

Acknowledgments: CSI Team, Altera
FPGA'16 Designer Session

Project Catapult History

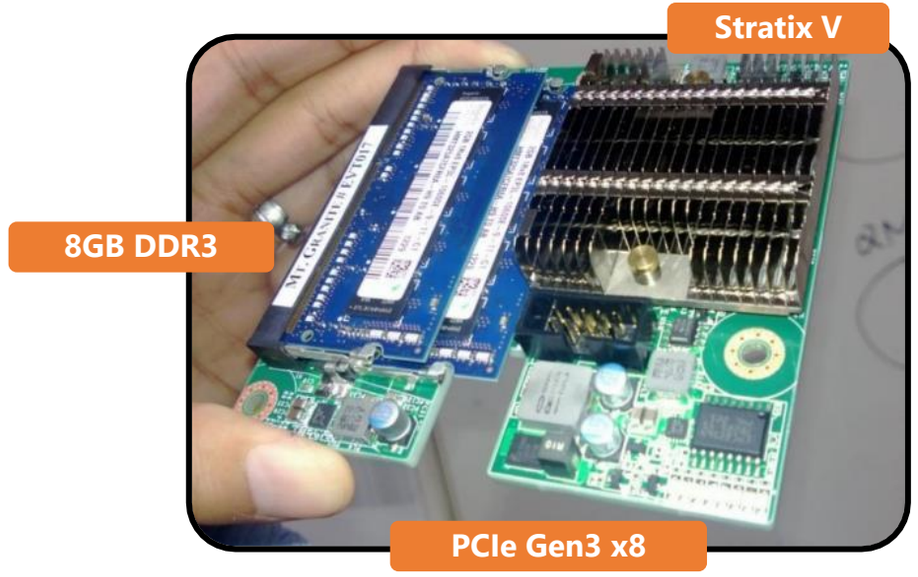


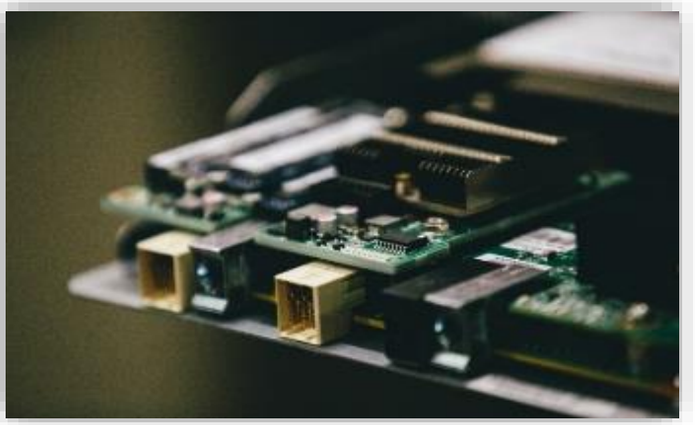
- December 9, 2010 – initial meeting
  - Christmas break 2010: feasible to accelerate ranking?
  - January 12, 2011: meeting with Bing leadership
- 2011 – v0: ported Bing ranking stack, built BFB board
- 2012 – v1: developed distributed architecture
- 2013 – Took v1 to scale, Bing pilot
- 2014 – v2: developed new architecture
- 2015 – Production

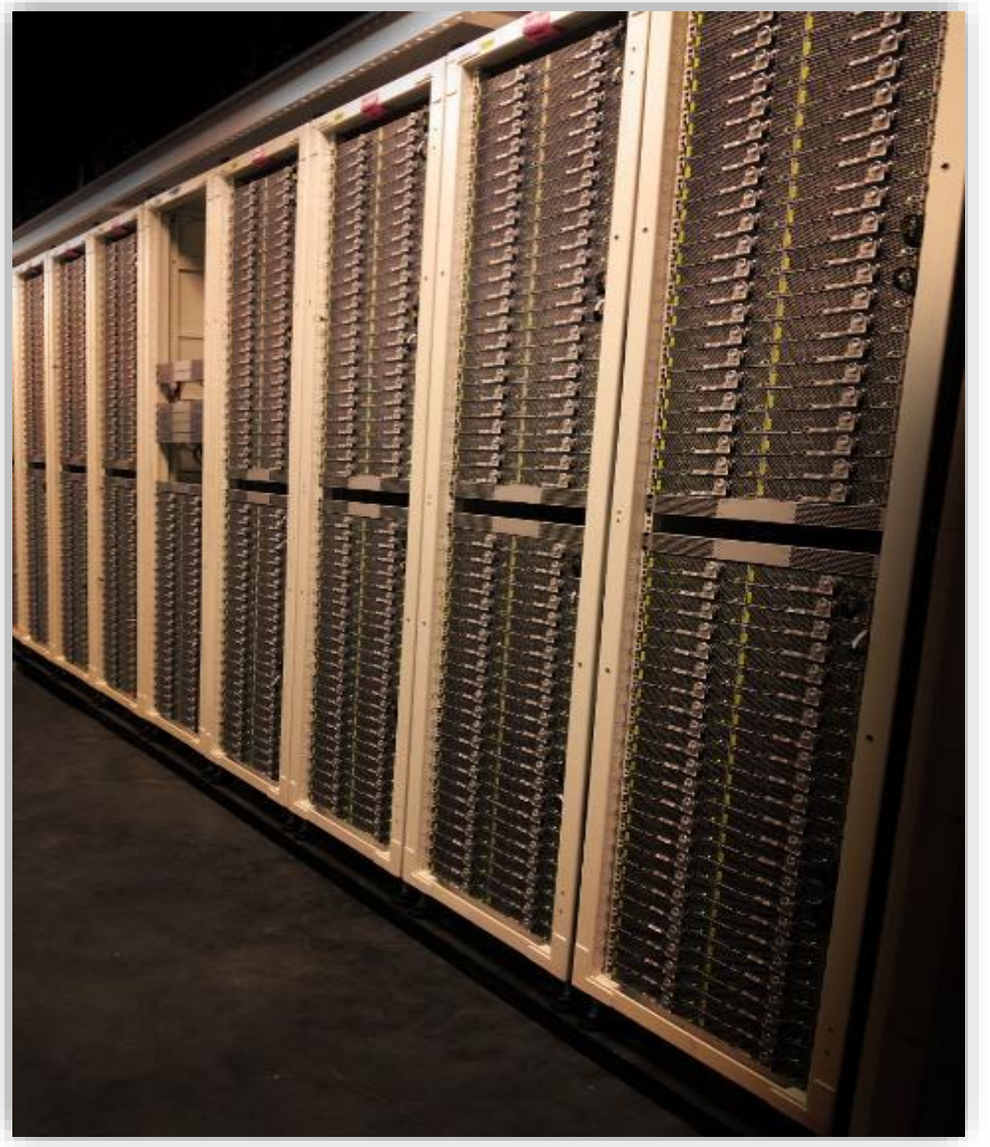
Putnam et al., “A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services”, ISCA’13

Catapult FPGA Accelerator Card

- Altera Stratix V D5
- 172,600 ALMs, 2,014 M20Ks, 1,590 DSPs
- PCIe Gen 3 x8
- 8GB DDR3-1333
- Powered by PCIe slot
- Torus Network

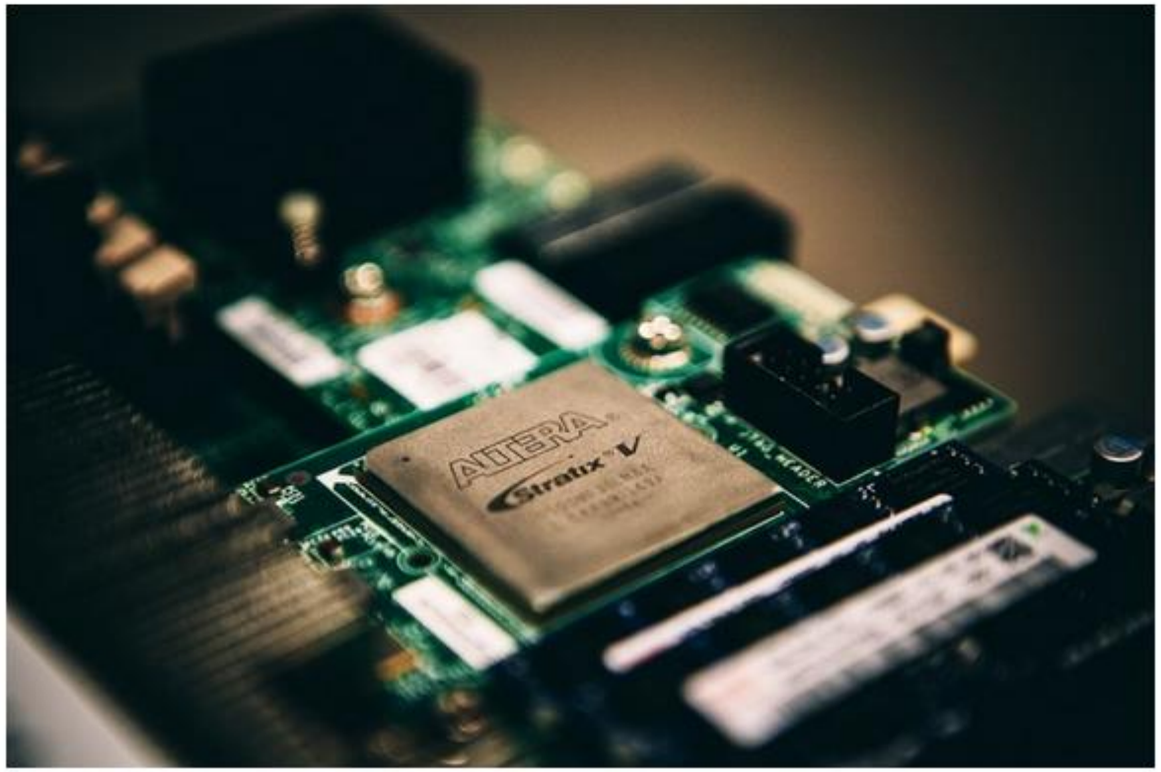






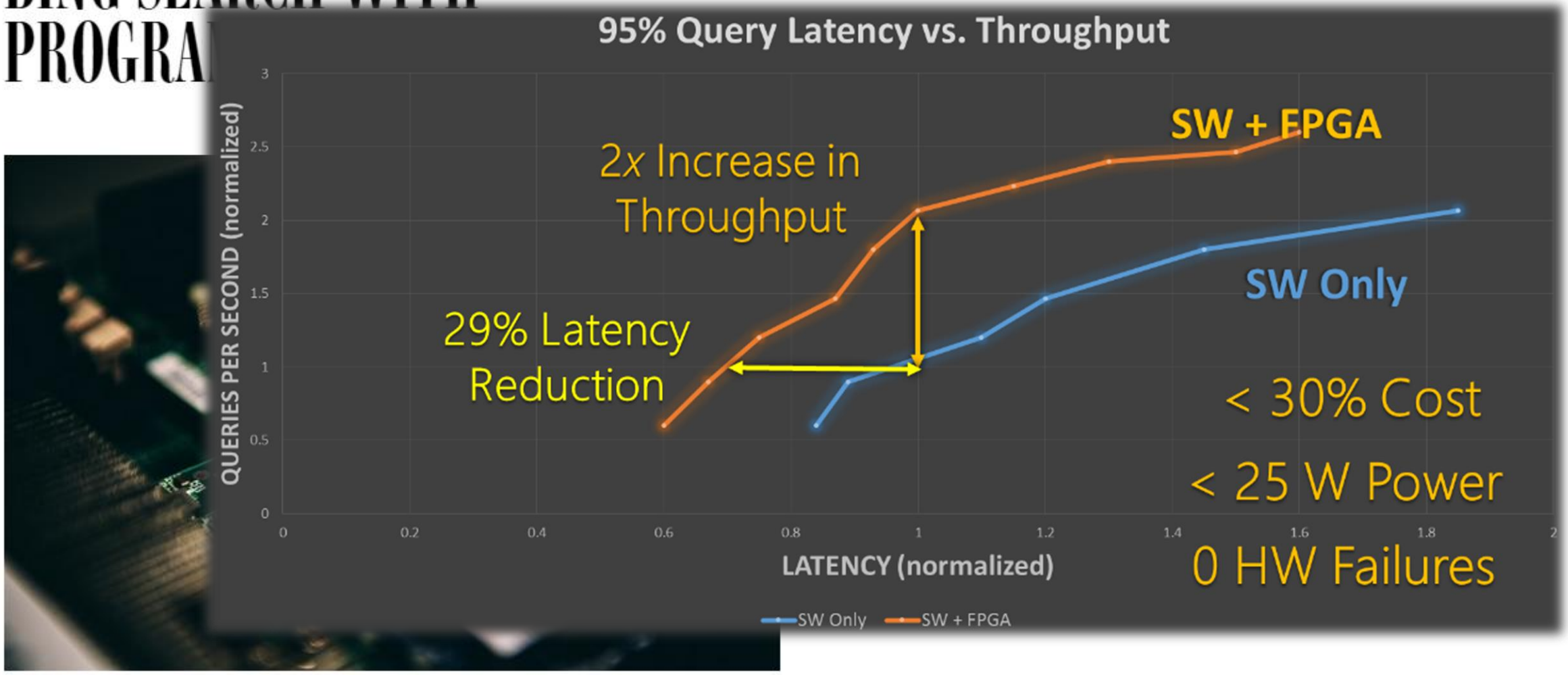
1,632 server pilot deployed in production BN datacenter

MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRAMMABLE CHIPS



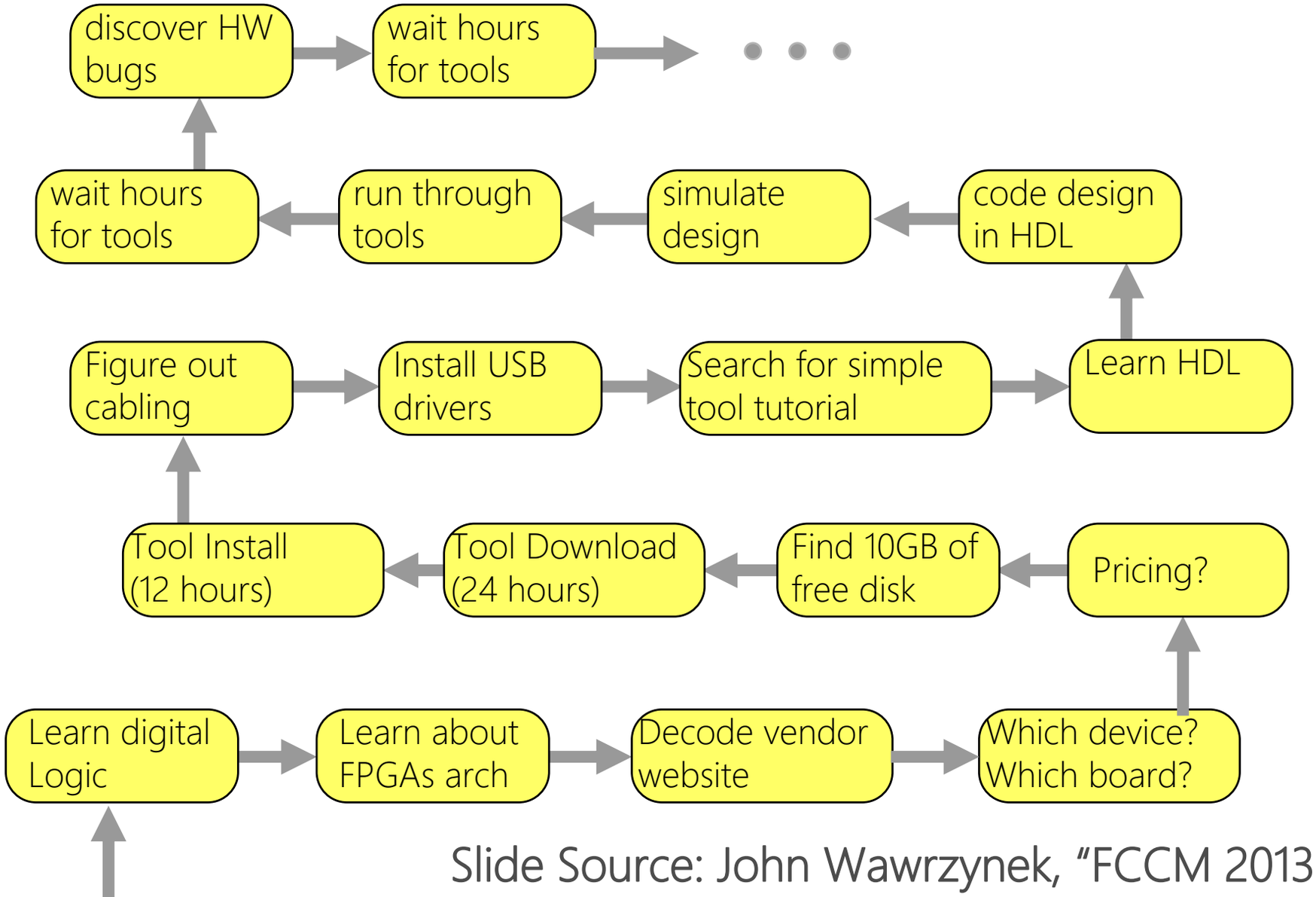
http://www.wired.com/2014/06/microsoft-fpga/

MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRA



http://www.wired.com/2014/06/microsoft-fpga/

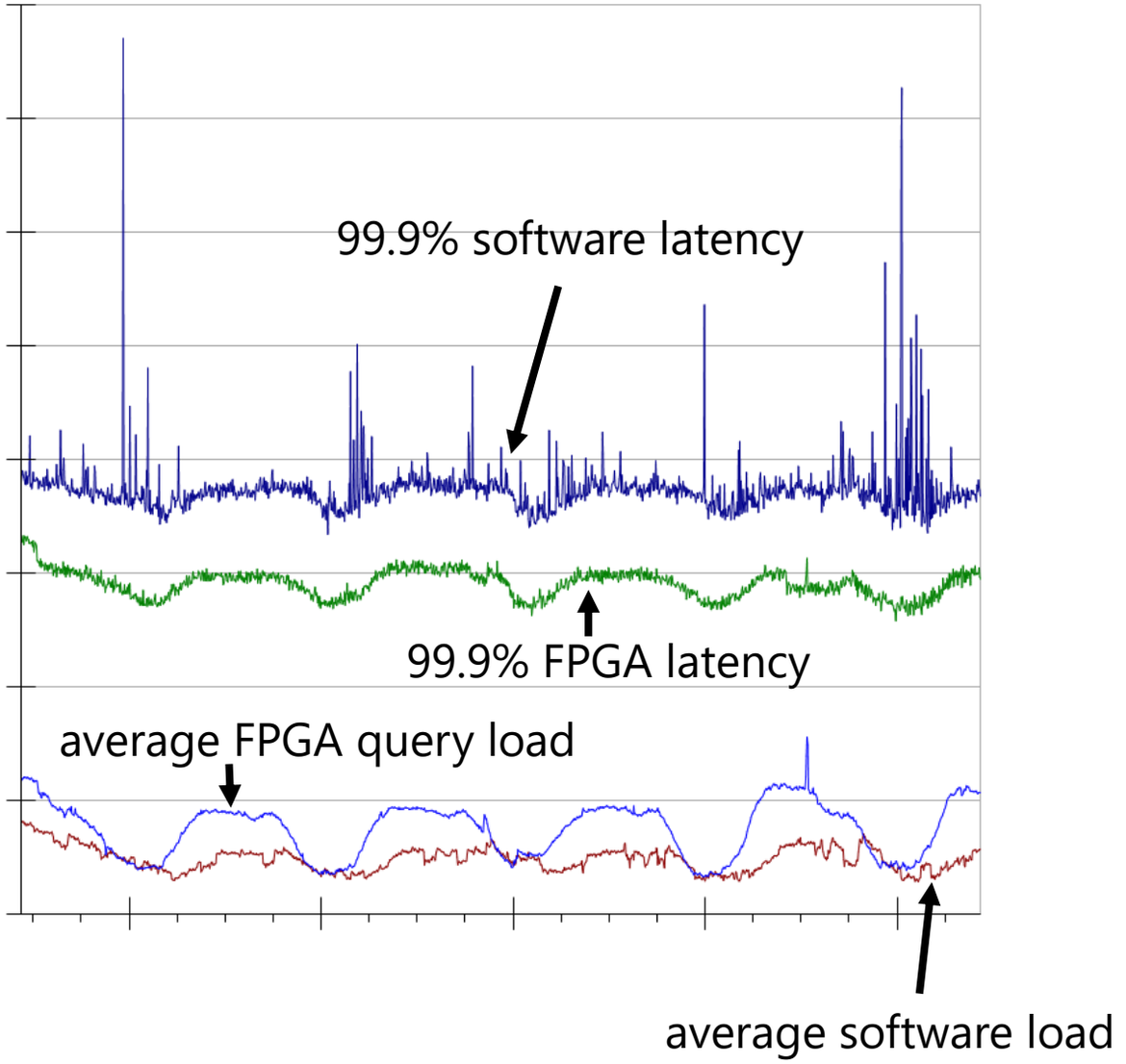
So You Want to Compute With FPGAs?



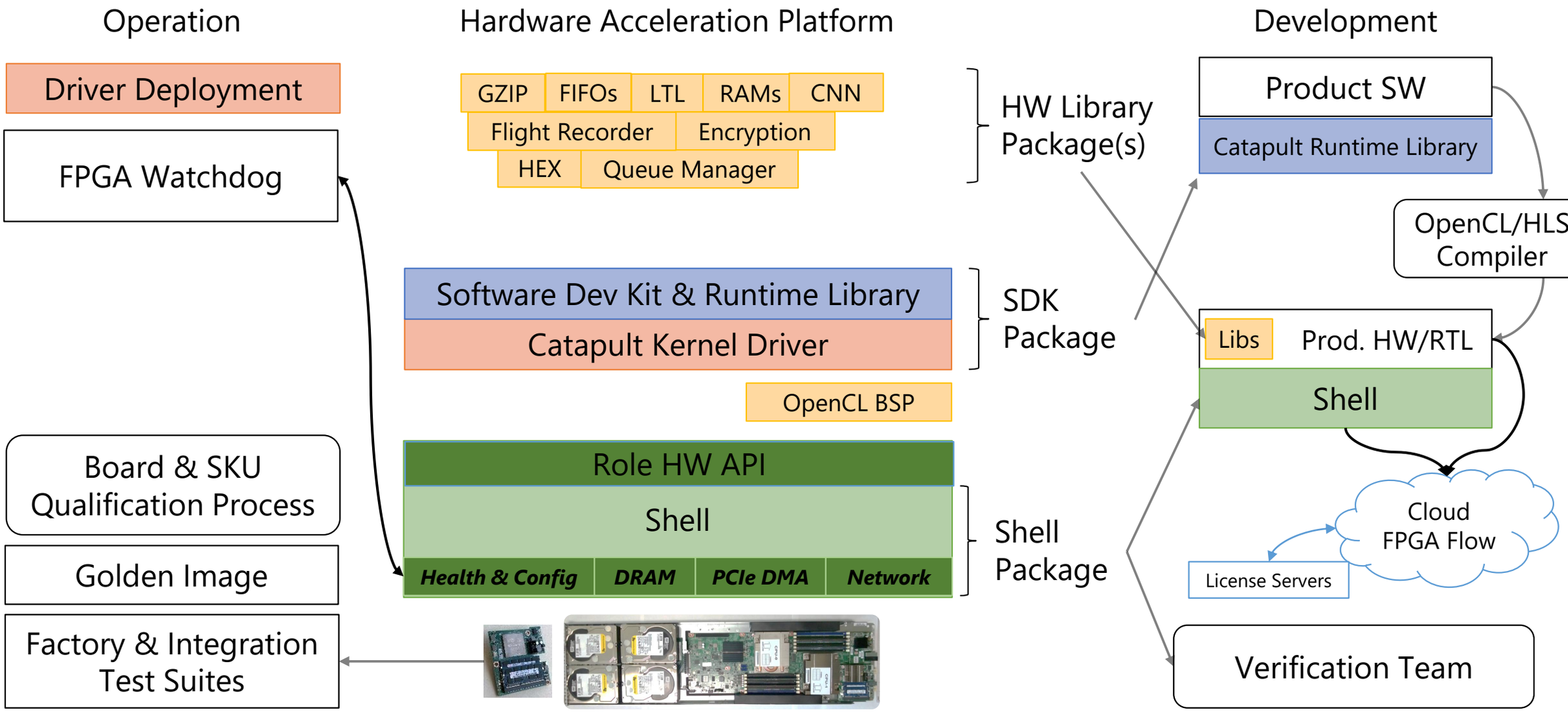
Slide Source: John Wawrzynek, "FCCM 2013 Panel", 2013

Anecdote

- Newly hired FPGA developer assigned to port critical Bing feature to FPGA
- Using Catapult platform, developer built and deployed new feature with >2X net gain in 1 month

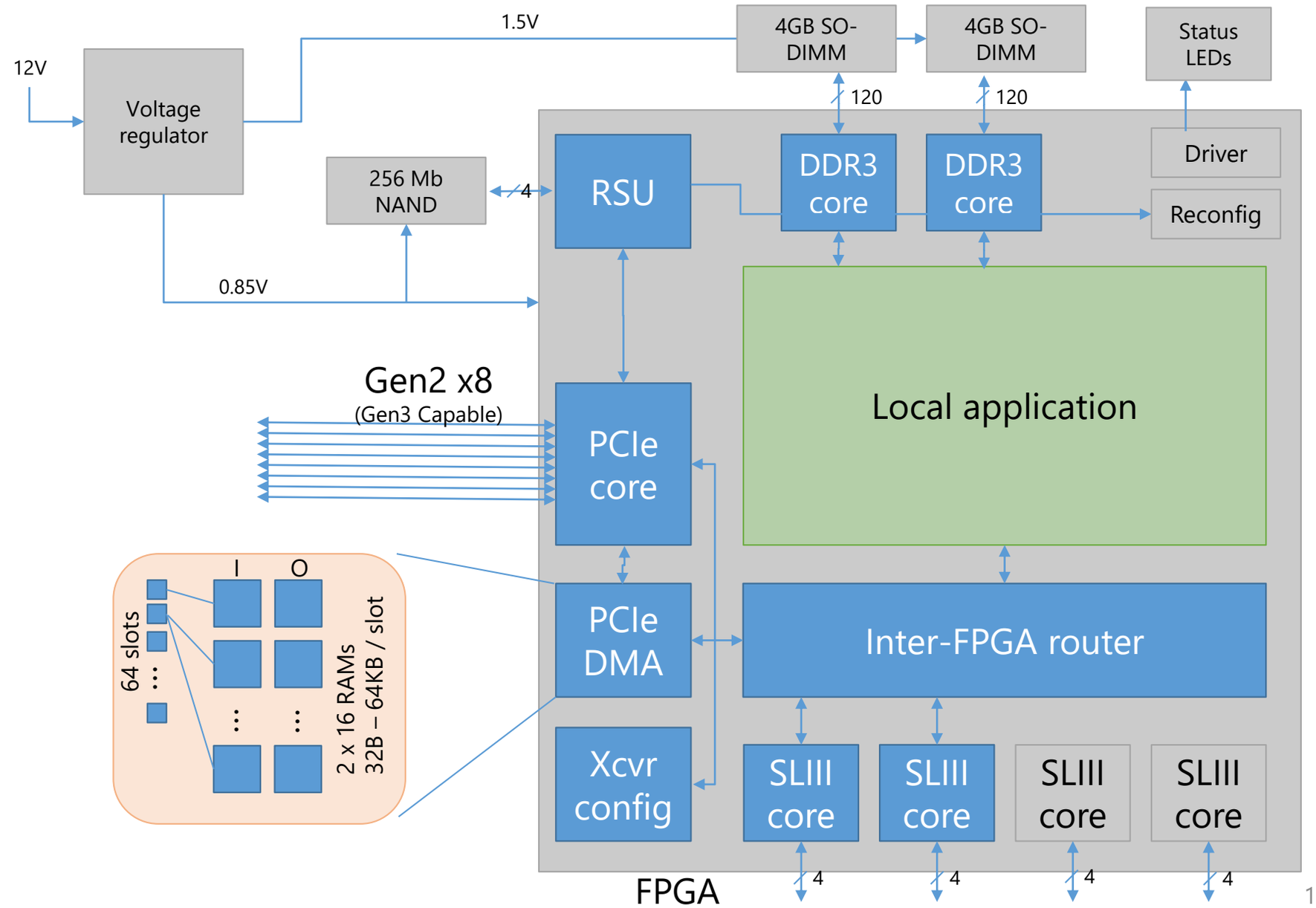


Requirement 1: Build Production-Class Platform for Developers, Developers, Developers, …



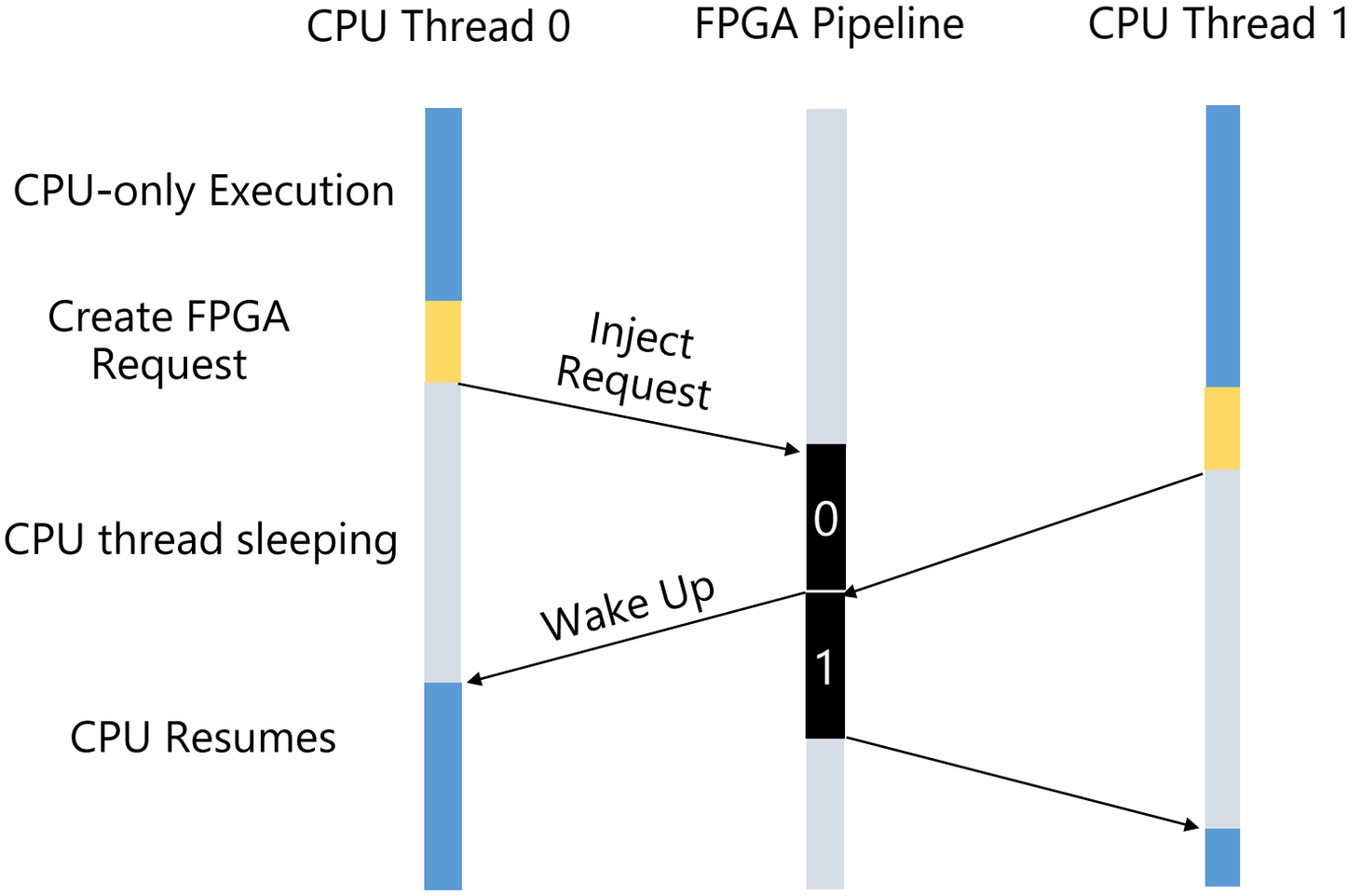
Catapult Shell Architecture

- Common I/O
  - Network
  - DRAM
  - PCIe DMA
- Verified to ASIC standards by verification team
- Published as stable packages to developers
- OpenCL BSP support



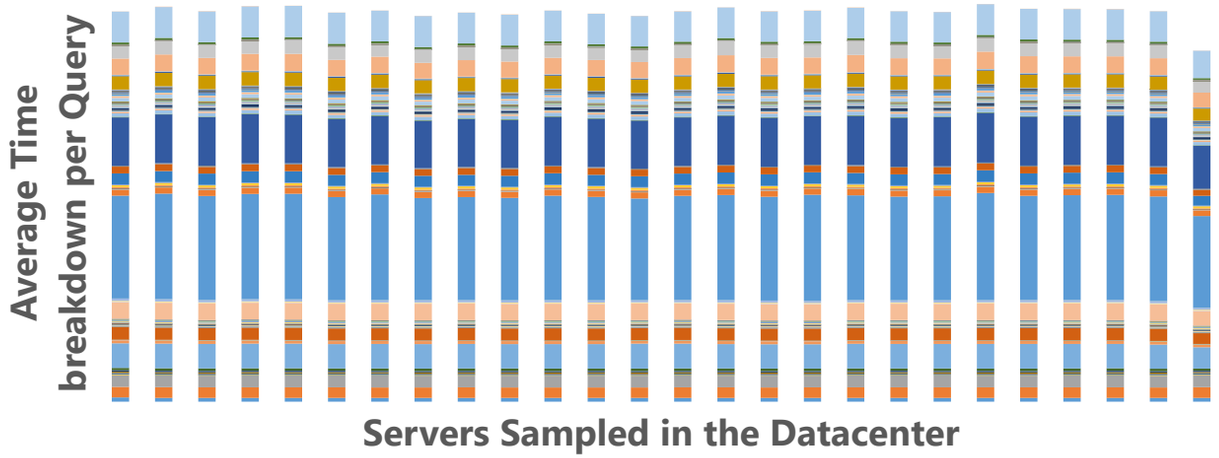
Catapult Windows SDK & PCIe Driver

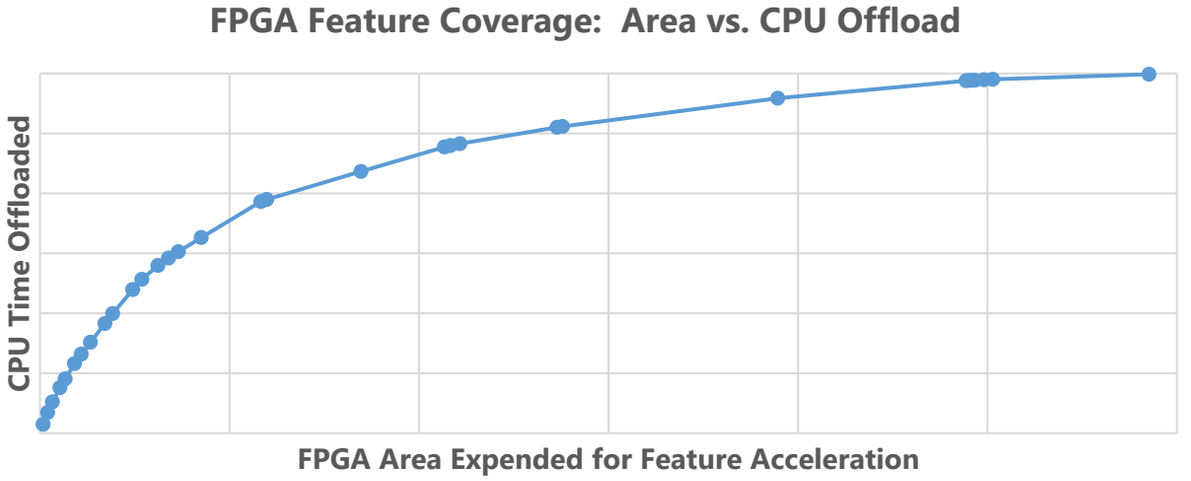
- SDK: thread- and multi-process safe library for CPU-FPGA message passing
- Driver: handles FPGA reconfiguration without server reboot (~3sec)
- Shells, SDKs, and Drivers are compatible across versions



Requirement 2: Be Obsessively Data-Driven When Selecting Features To Accelerate

- Real cloud workloads extremely diverse and complicated
- Invest in in-situ performance profiling of target application
- Optimize on “return per LUT”



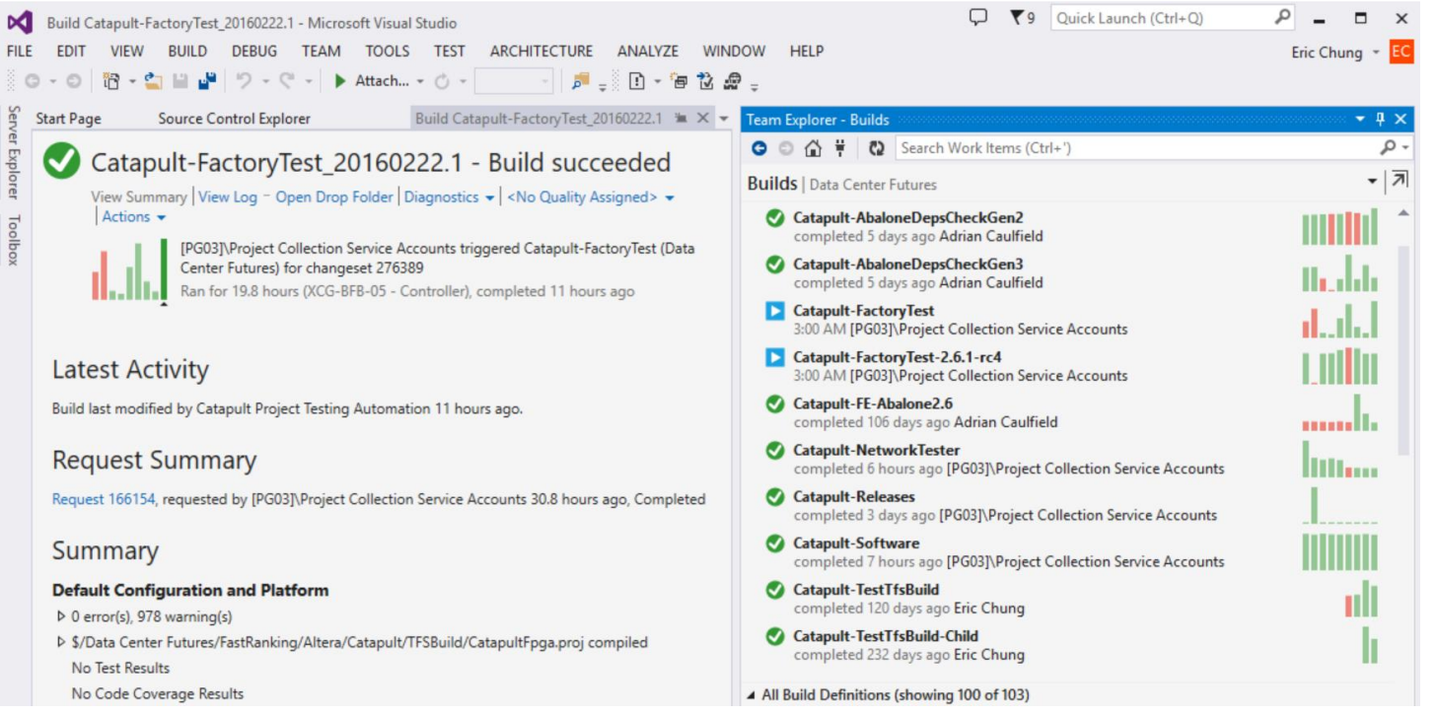


Requirement 3: Don’t Underestimate Software

- Invest properly in architecting efficient, correct HW-SW contract
  - Ignoring this can lead to performance pathologies, hangs, parity issues
- Establish through formal HW-SW contracts
  - Formalize and evolve through APIs and header definitions
  - Closely monitor SW overheads
  - Run-time check of HW-SW versions and data types
  - Tight coordination with software teams
- Build tools to automate code generation from precise specifications

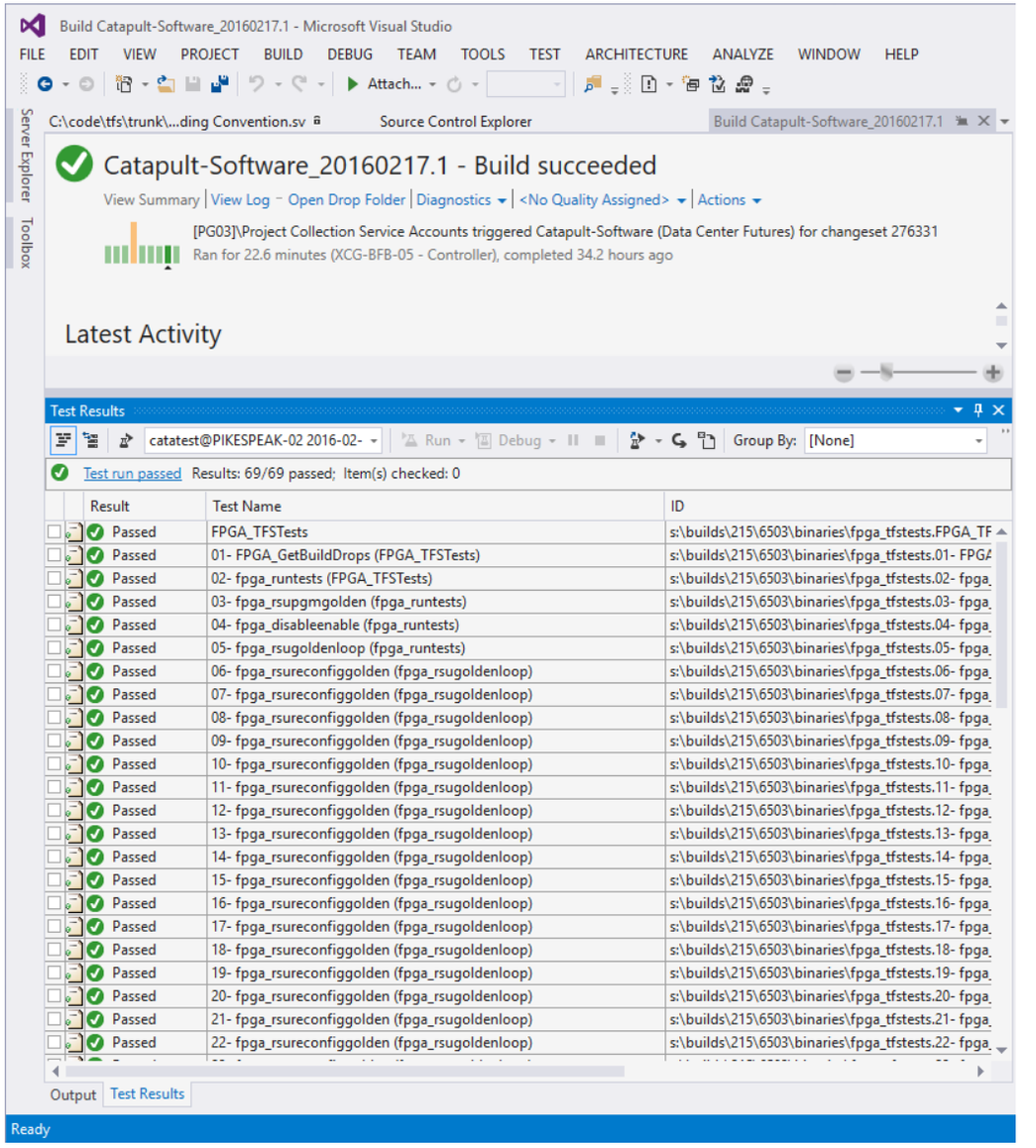
Requirement 4: Scale Out FPGA Tools in Cloud

- MS Cloud Tools
  - Run place-and-route and simulation tools on scalable resources
  - Speed up timing closure with parallel seeds
  - Must support determinism
  - Build caching shared amongst multiple users
  - Visual Studio integration
  - Partial reconfiguration
  - Static lock regions



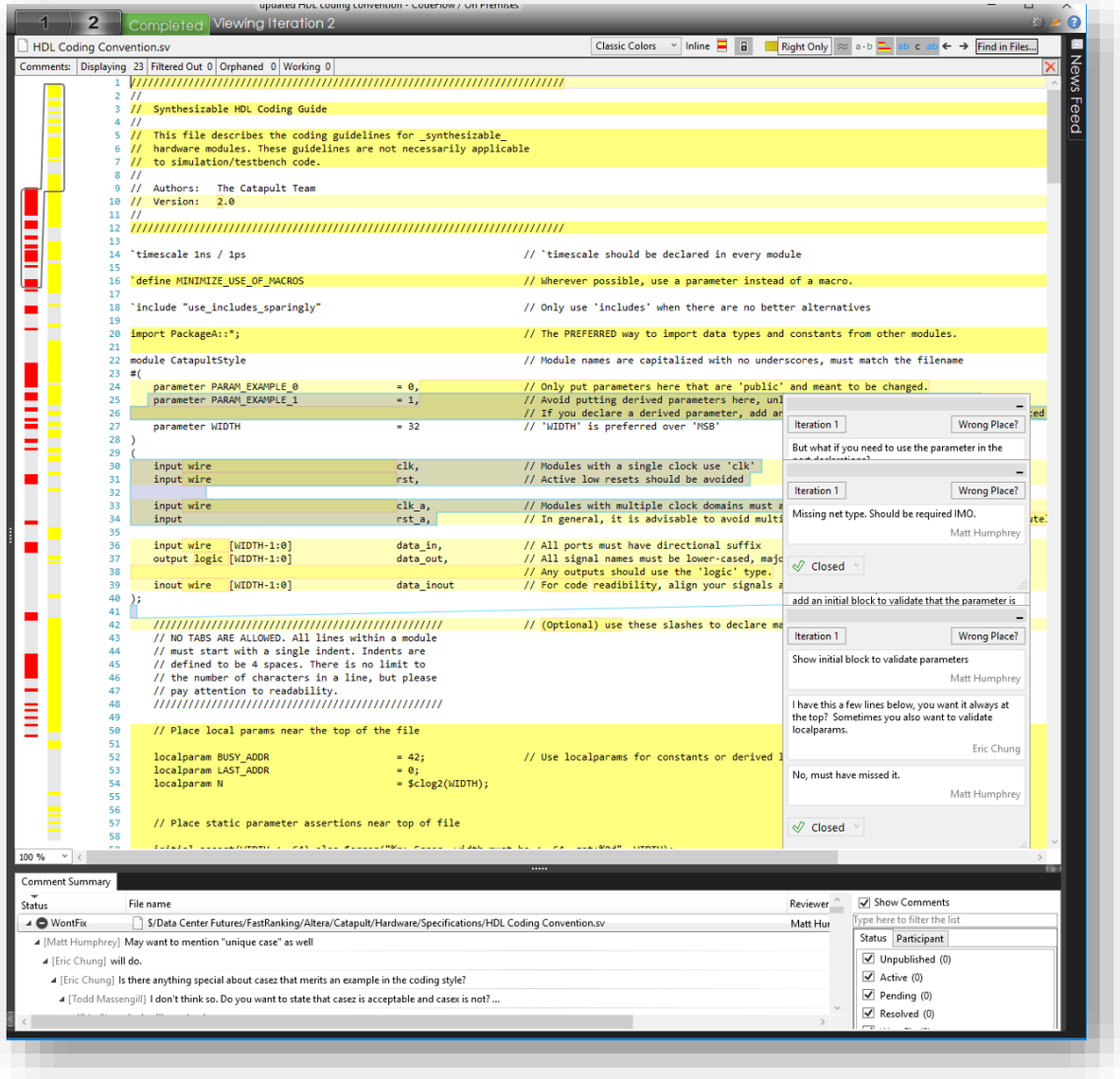
Requirement 5: Automate Hardware Testing

- Invest in framework for automatic regression testing with FPGAs
- Test HW automatically on check-ins
  - Cross-compatibility between software/drivers/shell
  - Reconfiguration stress test
  - Factory test
  - Applications
  - Golden images



Requirement 6: Adopt Software Best Practices

- Establish high bar for coding
- No “lone-wolf” check-ins
- Rigorous code reviews
- Strict coding guidelines
- Automate linting on check-ins
- Track tasks systematically (e.g., Visual Studio TFS)



Requirement 7: Prepare for Bugs in Deployment

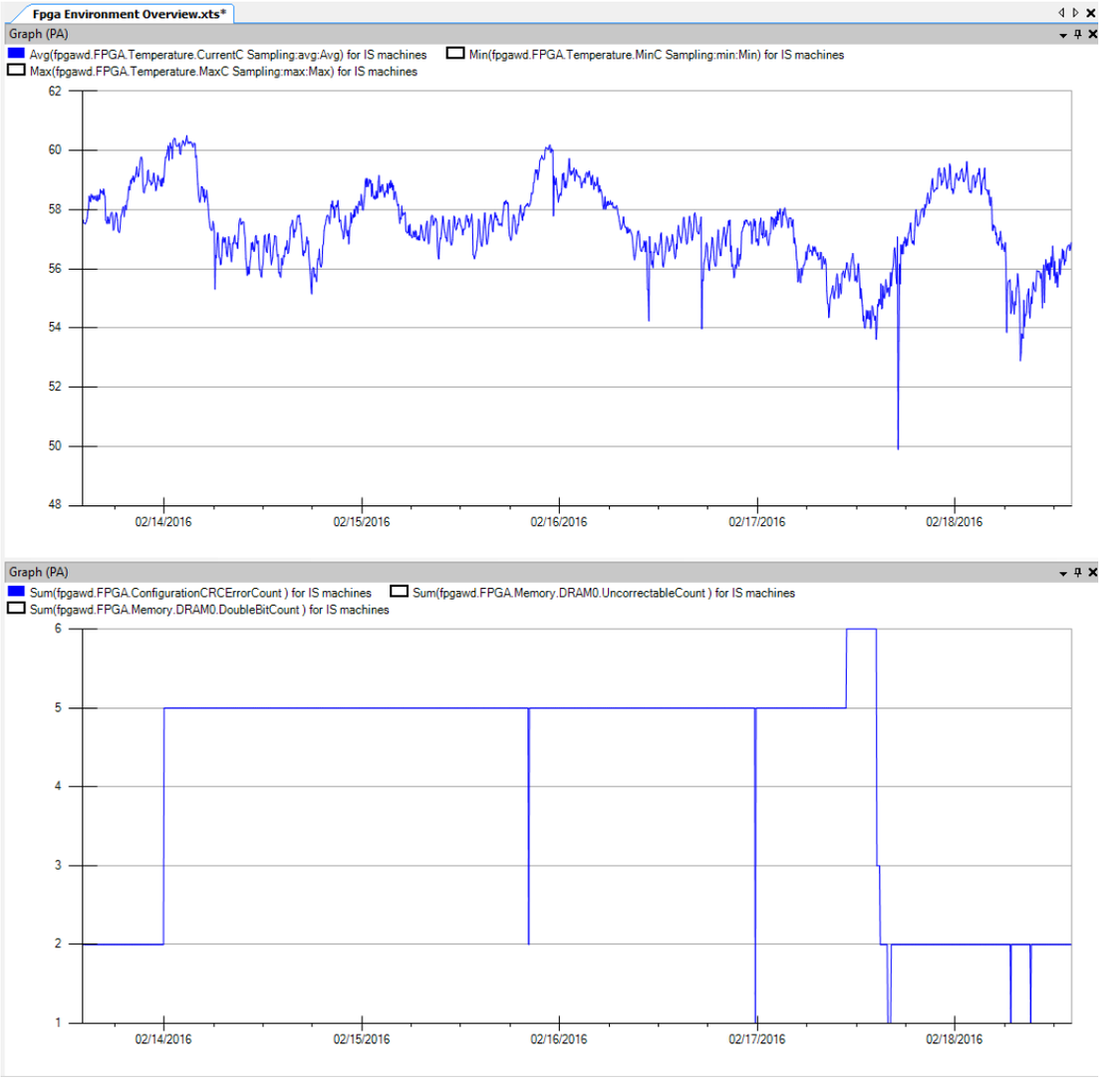
- Challenges
  - Bugs inevitably slip into deployment ➔ lead to stalls or hangs in software
  - *No digital logic analyzer (e.g., Signaltap)*
- Goals
  - Minimize service disruption during bug-induced failure
  - Minimize time-to-recovery
  - Collect information for off-line repro and debug
- Solutions
  - Lightweight in-situ monitoring and mitigation of failures in the FPGA itself
  - Examples: Flight Data Recorder, Hardware Exceptions, Role self-reset

  ➔ *Enabled successful debug of rare FPGA hang (once every few days) in datacenter*

Requirement 8: Support Fault Tolerance at Scale

- Challenges in Large-Scale Deployments
  - How to minimize server disruption during FPGA reconfiguration?
  - How to track FPGAs' health across datacenter?
  - How to distinguish FPGA failures from others?
  - How to log FPGAs without disrupting service?
  - How to respond to failures?

- Solutions
  - Datacenter-scale telemetry of FPGAs using AP
  - Tuned drivers minimized impact during reconfiguration
  - Continuous monitoring using watchdogs ➔ automatic triage steps
  - DRI for 24/7 monitoring of livesites



Closing Remarks

- Programmable logic can play major role in datacenter architecture
  - Will enable new applications, services to be cost effective
- It is viable to support a high rate of innovation with robust, scalable, & productive development platform
- erchung@microsoft.com