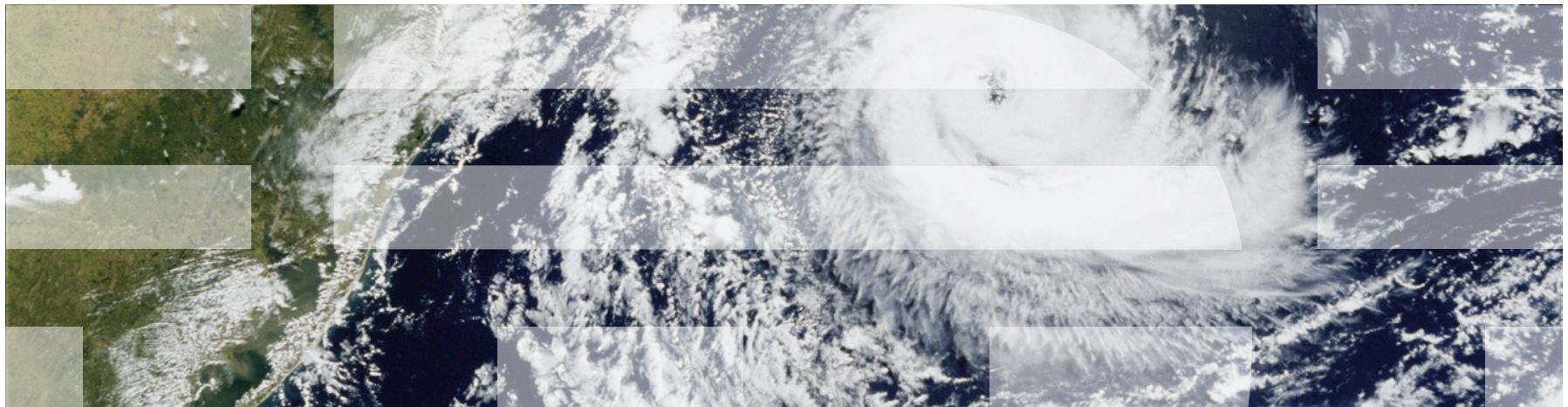PIs: Michael Hind, Yuqing Gao
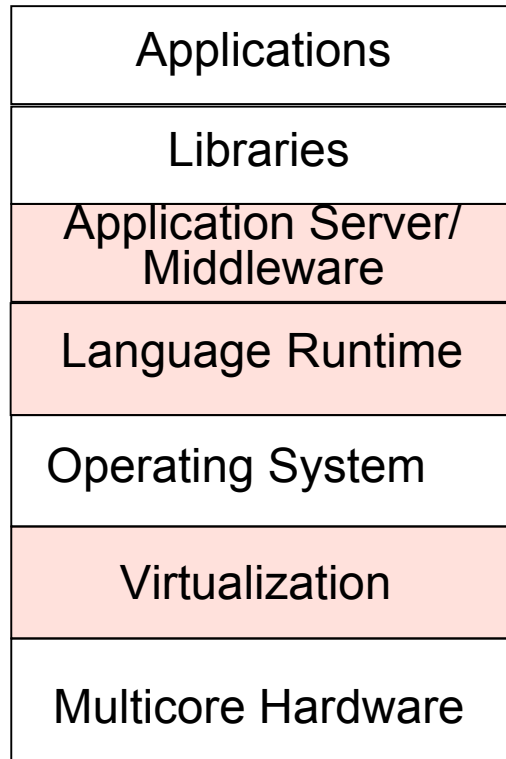Execs: Brent Hailpern, Toshio Nakatani, Kevin Nowka

2011 IBM Research Strategic Initiative: Workload Optimized Systems
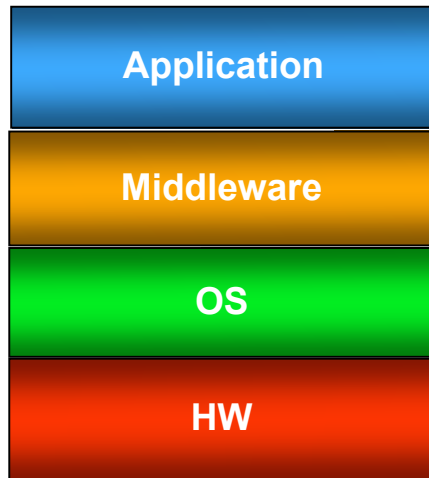
Yuqing Gao

IBM

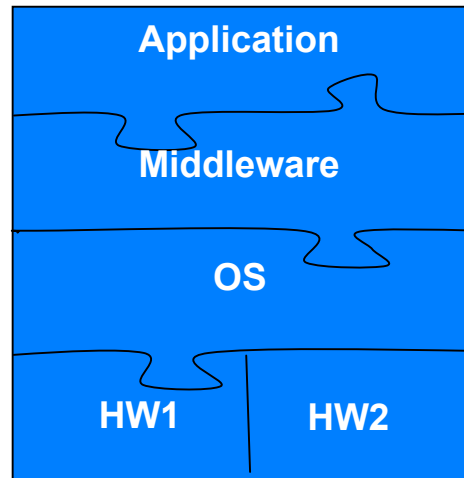| |
|---|
| Applications |
| Libraries |
| Application Server/ Middleware |
| Language Runtime |
| Operating System |
| Virtualization |
| Multicore Hardware |

- The industry has created a layered runtime stack
  - Each layer provides an abstraction, hides details
- Widely successful
  - Addresses complexity, enabling sophisticated runtimes
  - Threat of customer migration forces vendor innovation
- But creates performance issues
  - Too many layers
    - 1 level of indirection ➔ Brilliance
    - N levels ➔ **???**
  - "Layers of Liars"
    - Abstraction sometimes hides too much information
      - ex) GC wants to know # of real processors, not virtualized
  - Thin interfaces between layers impedes synergy and duplicates functionality
    - *AppServer, JVM, OS, Virtualization, HW* all present a thread abstraction, but do not communicate semantics among layers
- A WOS can address these performances issues, but at the risk of
  - increasing development and management complexity
  - raising customer concerns about vendor lock in
- Why now?
  - Performance free lunch (frequency scaling) no longer exists
- The HW design point that leads to frequency scaling stagnation also provides multicores, scaleout, and heterogeneous systems
  - Great opportunity, but provides significant challenges

Workload Optimized Systems Definition and Components   IBM

Traditional System

Application
Middleware
OS
HW

Workload Optimized System

| |
|---|
| Application |
| Middleware |
| OS |
| HW1 | HW2 |

Software and hardware are optimized for a specific workload, resulting in higher performance, reduced power consumption, and/or simpler management/use

Three Needed Components for Workload Optimized Systems

- Programming models and tools to ease software development of WOS ("Build")
  - Help determine which workloads, and which parts of workloads, to target
  - Help to productively develop the system
- Optimized middleware and hardware systems to realize performance ("Run")
  - How to create specific SW and middleware systems that realize WOS value
  - How to create fit-for-purpose appliances to accelerate customer's time-to-value
  - How to optimize connections and synchronization between subsystems
- Software to manage systems in a simplified manner ("Manage")
  - How do we ensure overall system and resource management and provides appropriate value of heterogonous systems

IBM zEnterprise System

Consolidation

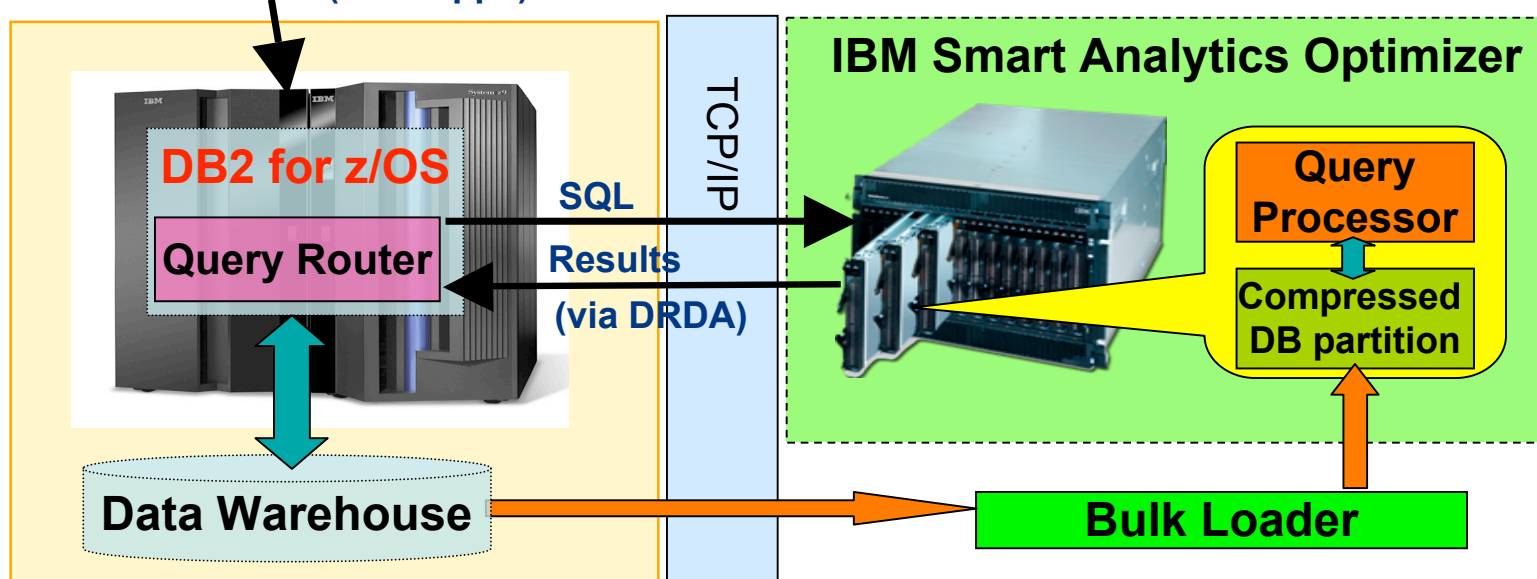|	Application	
System Z stack	System X stack	System P Stack

(TCP/IP) link
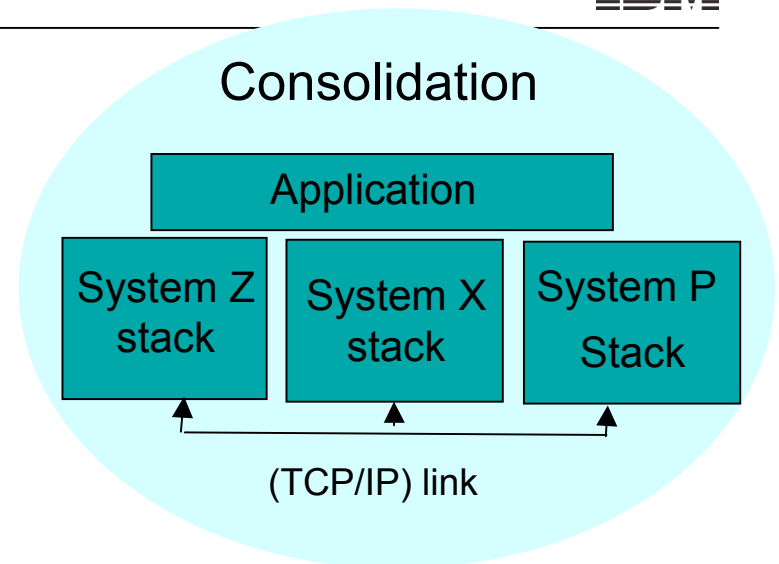
- The revolutionary new design of the zEnterprise System addresses the complexity and inefficiency of today's multi-architecture data centers by giving you the ability to integrate and unify IBM System z, Power and System x resources as one complete system

- IBM zEnterprise Unified Resource Manager (Unified Resource Manager)
  - Integrated System z management facility for zEnterprise platform management
    - Enables clients to install, monitor, manage, optimize, diagnose, and service resources and workloads from a single point of control while extending System z qualities of service across the entire infrastructure.

- IBM Smarter Analytics Optimizer

Workload driven approach

- Characteristics of potential acceleration opportunities
  - Existing/legacy code (middleware platform or application software) is suboptimal on a host HW platform – need to identify bottlenecks
    - Too slow, doesn't scale, or too costly in some other sense
  - Sub-tasks or functions are separable from the main code
    - Changes to legacy code is minimal
    - Changes to legacy code is non-disruptive (can't afford to break the legacy code)
  - Sub-tasks or functions can exploit special capabilities of different HW (commodity or special HW: e.g., GPU, FPGA)
    - Faster CPUs
    - Abundantly and cheaply available multicores
    - Abundantly and cheaply available memory and caches
    - Network-speed processing

IBM Smart Analytics Optimizer Configuration

SQL Queries (from apps)

DB2 for z/OS

Query Router

Data Warehouse

TCP/IP

SQL

Results (via DRDA)

IBM Smart Analytics Optimizer

Query Processor

Compressed DB partition

Bulk Loader

DB2 for z/OS:

- Routes SQL queries to accelerator
- User need not change SQL or apps.
- Can always run query in DB2, e.g., if
    - too complex SQL, or
    - too short an est. execution time

IBM Smart Analytics Optimizer:

- Multiple blades in blade center
- Connects to DB2 via TCP/IP & DRDA
- Analyzes, compresses, and loads
    - Copy of (portion of) warehouse
    - Partitioned among nodes
- Processes routed SQL query and returns answer to DB2

IBM Smart Analytics Optimizer Architecture
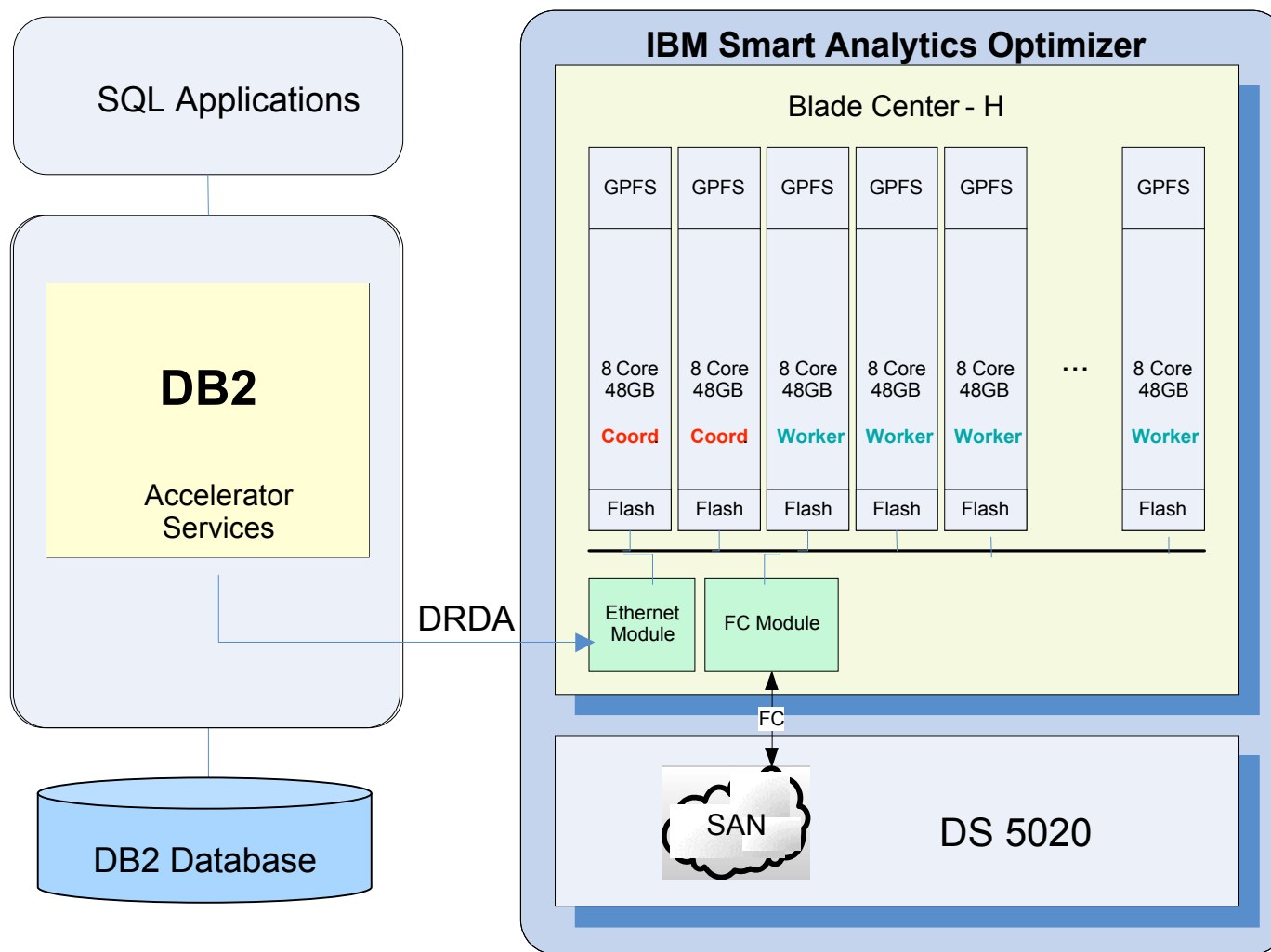
ISAO Accelerates Most the Longest-Running DB2 Queries
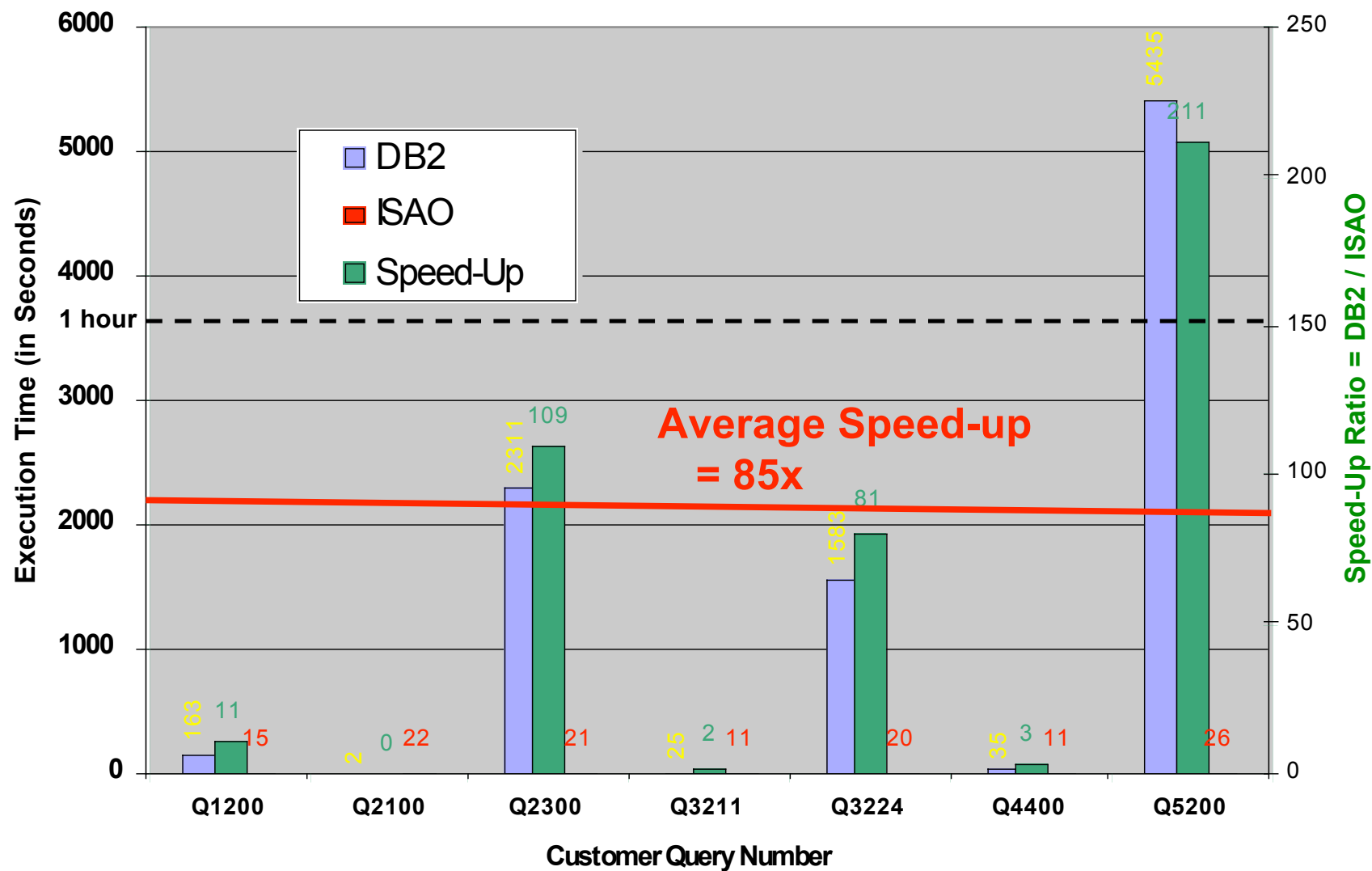

Average Speed-up = 85x

Legend:
- DB2
- ISAO
- Speed-Up

Y-axis (left): Execution Time (in Seconds) — 0, 1000, 2000, 3000, 4000, 5000, 6000
Y-axis (right): Speed-Up Ratio = DB2 / ISAO — 0, 50, 100, 150, 200, 250
1 hour (dashed line at ~3600)

X-axis: Customer Query Number — Q1200, Q2100, Q2300, Q3211, Q3224, Q4400, Q5200

Data labels:
- Q1200: 163, 11, 15
- Q2100: 2, 0, 22
- Q2300: 2311, 109, 21
- Q3211: 25, 2, 11
- Q3224: 1583, 81, 20
- Q4400: 35, 3, 11
- Q5200: 5435, 211, 26

WebSphere Datapower XC-10:
Accelerates Time-to-value and Reduces Total Cost of Ownership

Caching Appliance…
- ✓Enhanced DataPower 9004 Platform
- ✓Expanded cache capacity to 160GB (SSD), 32GB RAM
- ✓Market proven distributed cache technology

- Software
- WebSphere eXtreme Scale
- Derby DB without SQL engine
- Bedrock platform
- Systems Management
- Linux, Java 1.6

Core XC-10 (underlying cache infrastructure) features:
- ✓**Scale-out with ease -** Quickly plug-in more appliance to increase capacity and throughput needs
- ✓**Flexible and simple User Management interface** for monitoring and administration
- ✓**Drop-in Use Cases** – focused on speedy deployments and simplicity
  - ▪Offloaded Session Management for HTTP requests
  - ▪Dynacache support
  - ▪Support for generic non-java based clients (MemCacheD applications & .NET Rest, Php)
  - ▪Additional drop-in adaptors based on market/business opps
- ✓**Fast disk offload support** for object grid cache

Different Flavors of Hybrid Systems

Consolidation

Application

System Z stack | System X stack | System P Stack

(TCP/IP) link

Acceleration

Application

Host system

Accelerator

offload

Functional unit	Memory bus-attached	I/O attached	Network attached
Vector units	Crypto	Cell, GPU	XML, ISAO
Type 1	Type 2	Type 3	Type 4
1-50	500	10^3	10^4-10^6

tightly coupled

Invocation latency/granularity (cycles)

loosely coupled