#SHAREorg

# Fit For Purpose Platform Architecture Selection and Design

Joe Temple
IBM (jliitemp@us.ibm.com)

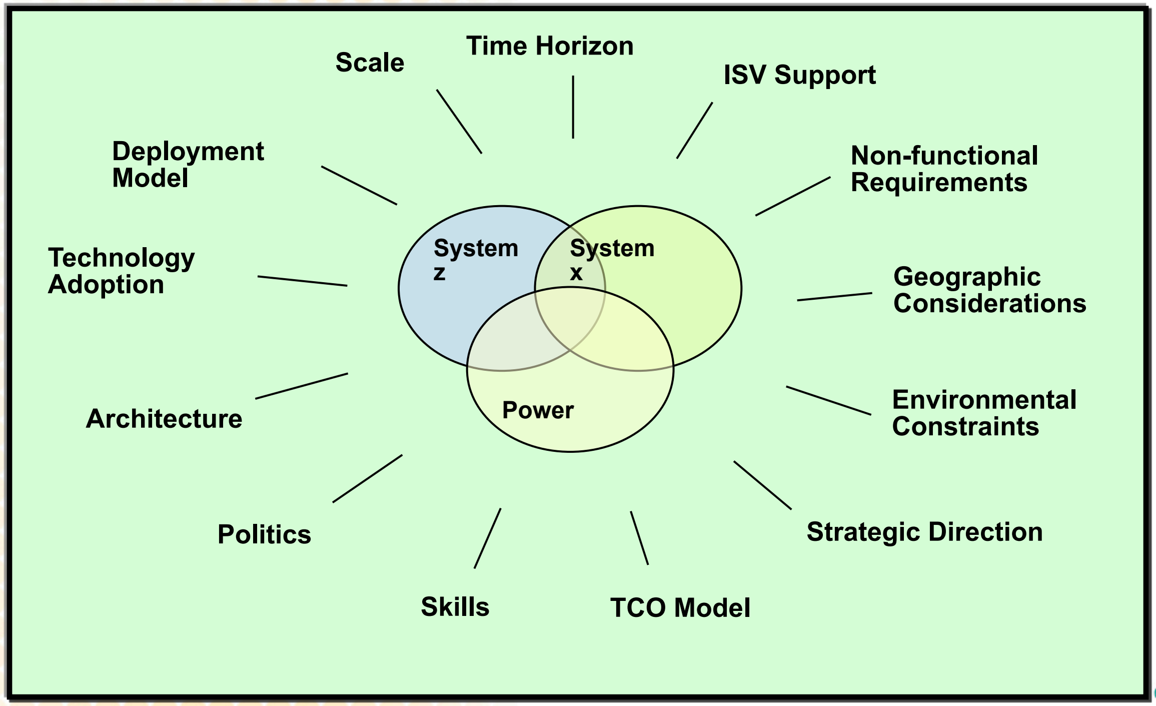
Aug 6, 2012 AM Session
11853

SHARE in Anaheim 2012

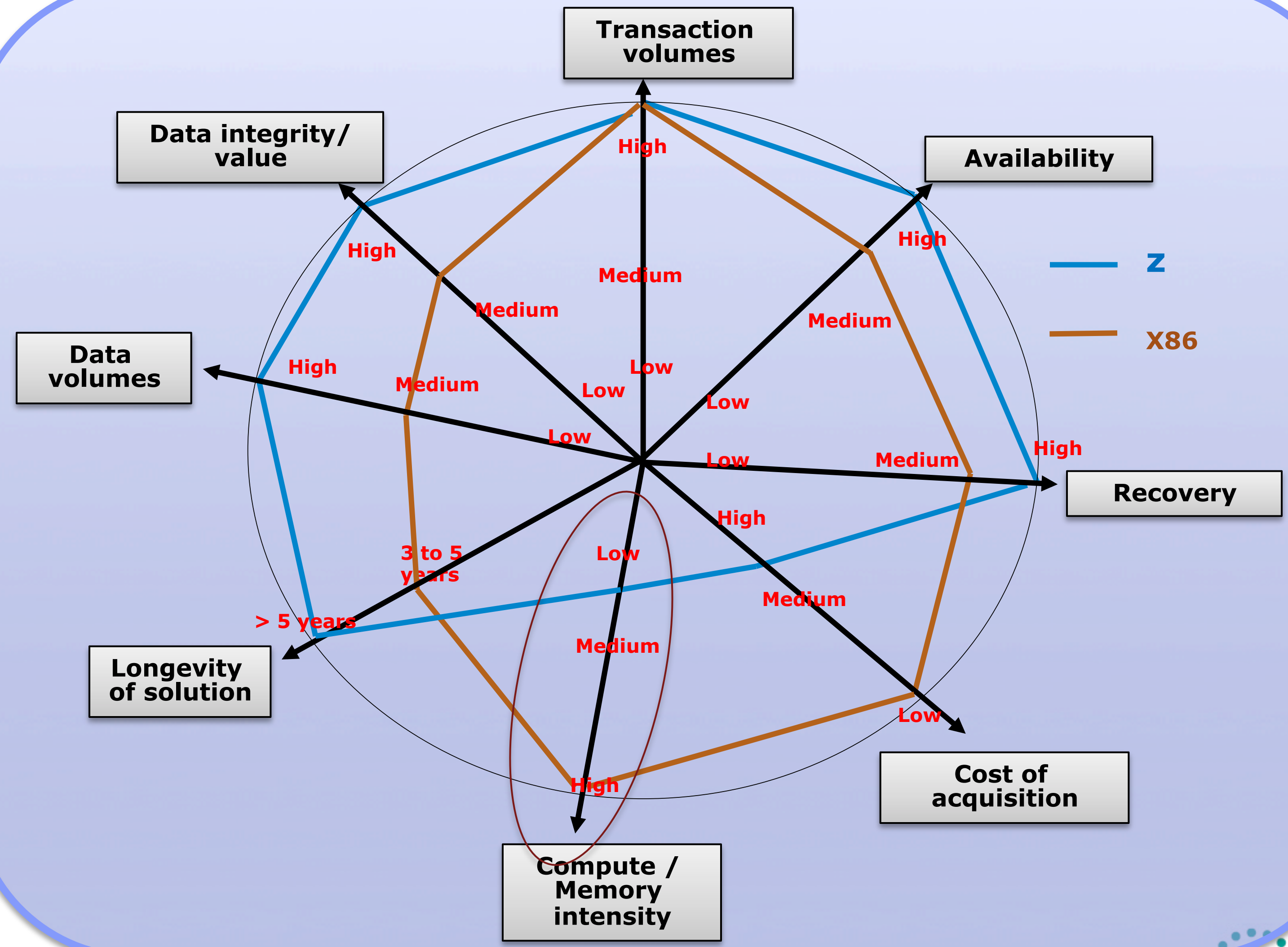
# What is Fit for Purpose (F4P)?

***Fit for Purpose is a client centric thought process that when applied, yields infrastructure architecture decisions which are in line with the client's requirements and local conditions.***

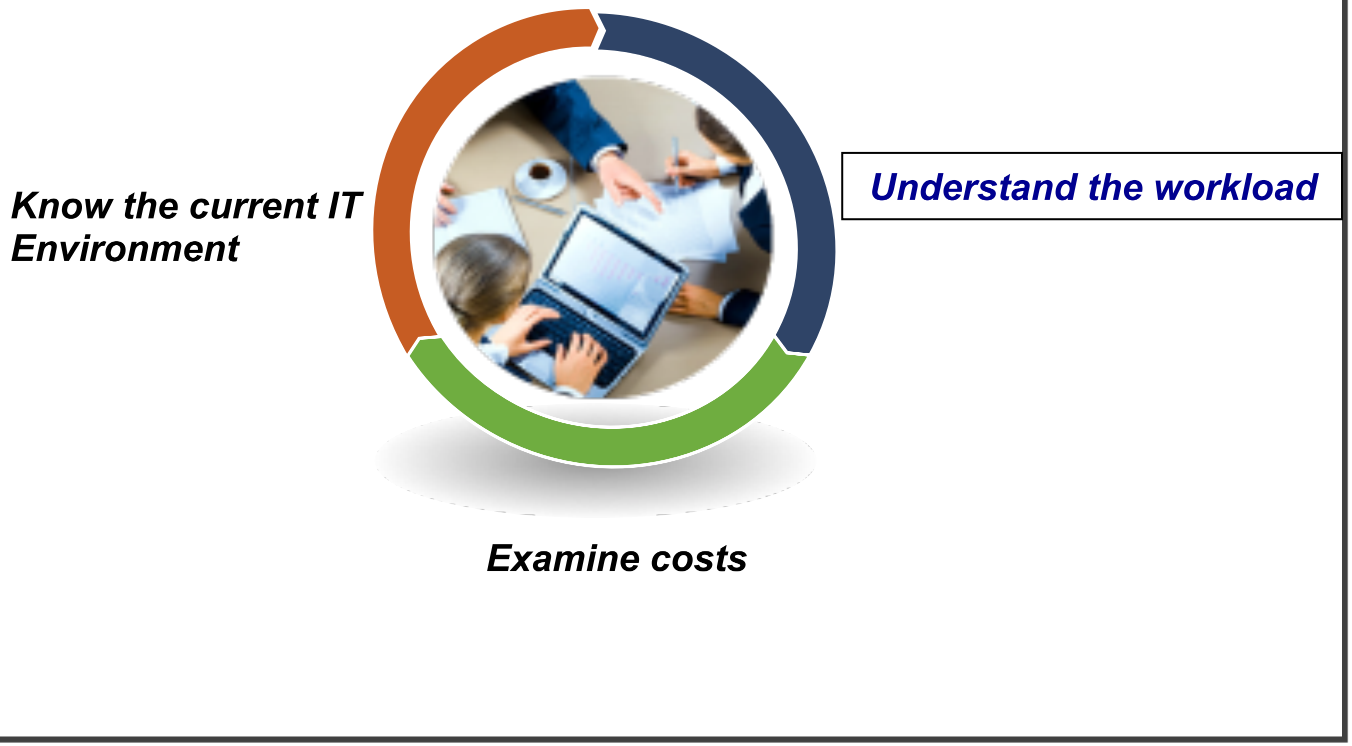
***It is based on the fundamental principles that "one size does not fit all" and that "local factors matter."***



# A Client's Decision Matrix



# Know the legacy, workload, and costs



# Fit for Purpose Categorized Workloads

## Mixed Workload – Type 1

- Scales up
- Updates to shared data and work queues
- Complex virtualization
- Business Intelligence with heavy data sharing and ad hoc queries

## Parallel Data Structures – Type 3

- Scales well on clusters
- XML parsing
- Buisness intelligence with Structured Queries
- HPC applications

*Application Function   Data Structure*   ***Usage Pattern   SLA   Integration   Scale***

## Highly Threaded – Type 2

WebSphere.

- Scales well on large SMP
- Web application servers
- Single instance of an ERP system
- Some partitioned databases

## Small Discrete – Type 4

- Limited scaling needs
- HTTP servers
- File and print
- FTP servers
- Small end user apps

Black are design factors     Blue are local factors

# Using Pfister's paradigm we can map the workload types to our existing platforms and new platforms we build as a result of a WOS study

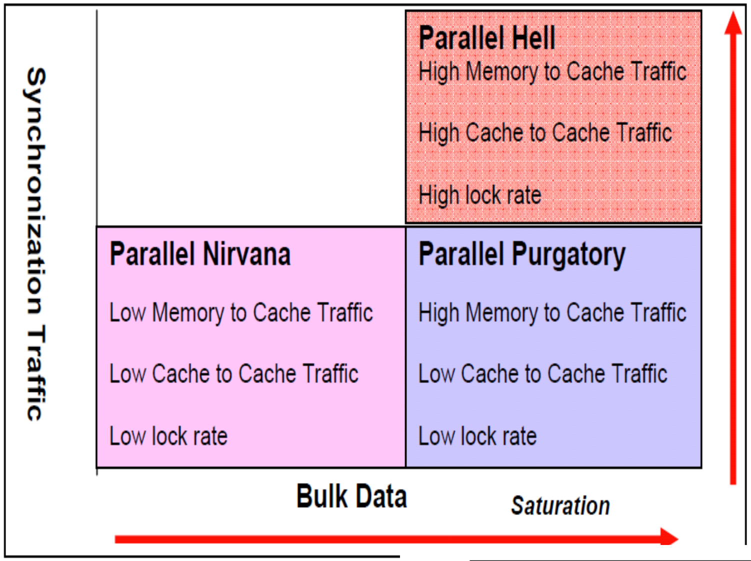
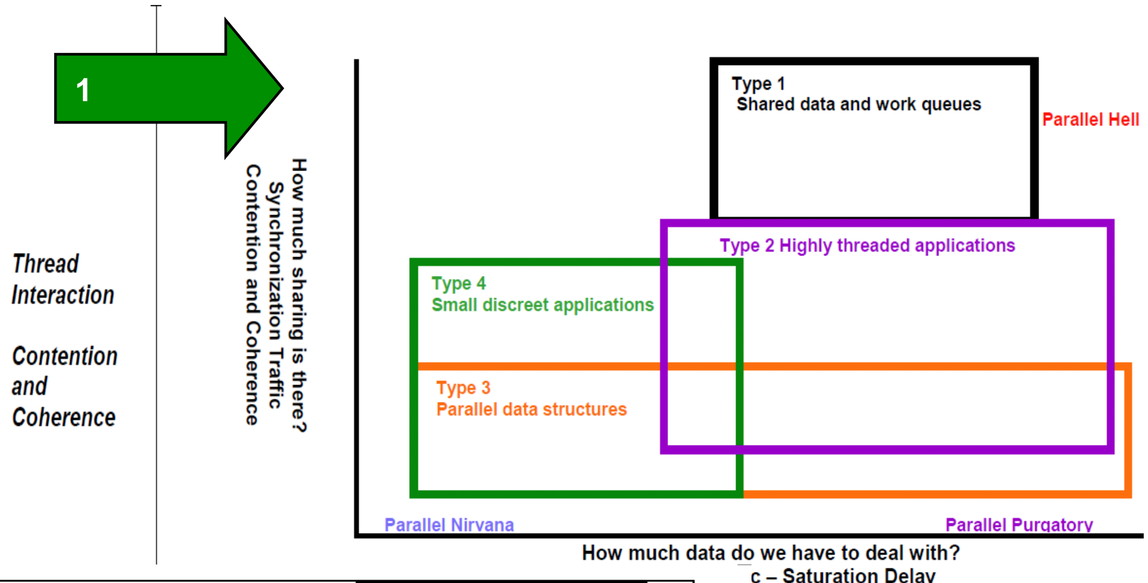



Figure 12: Pfister's Paradigm



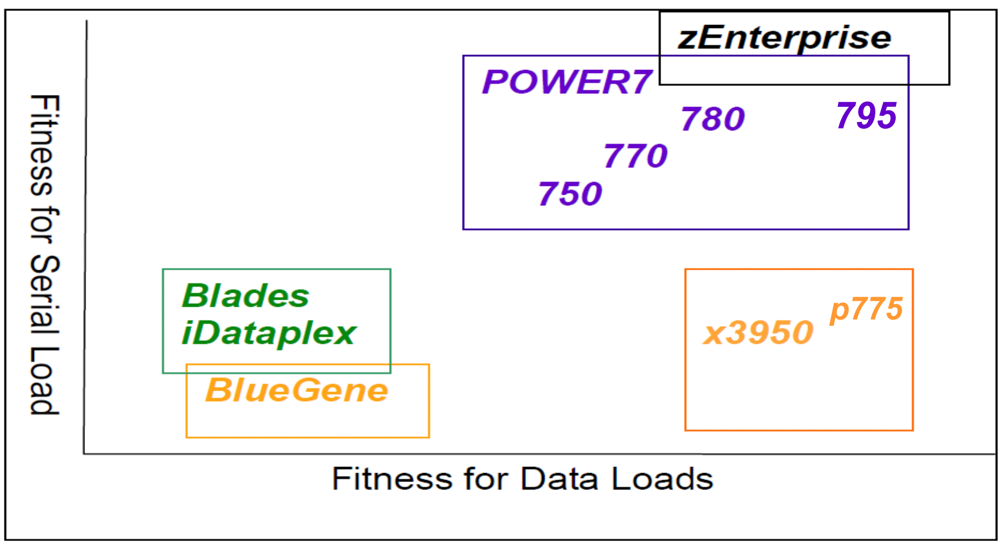
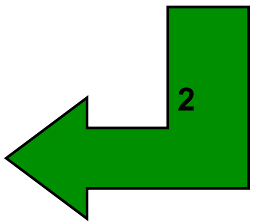
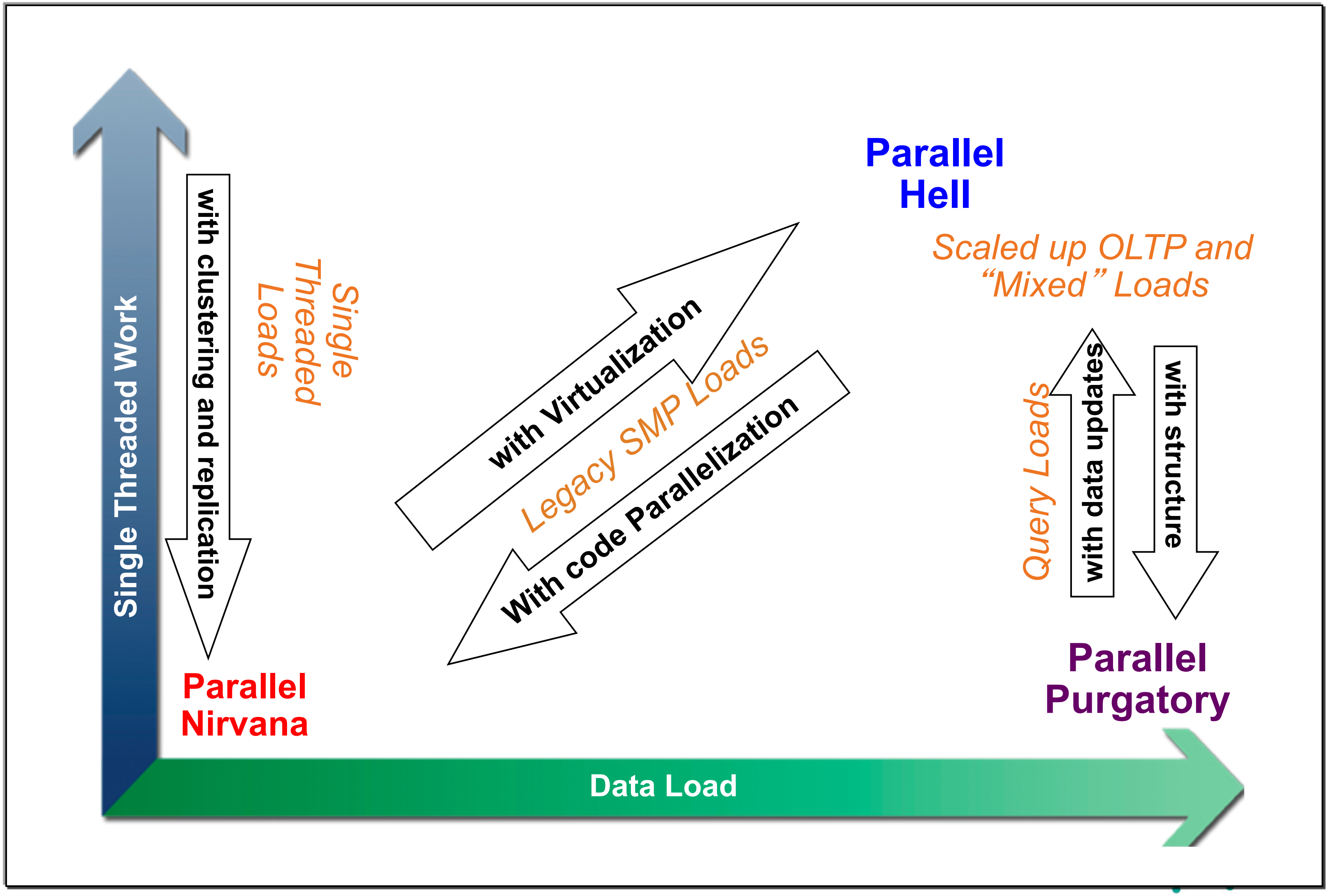



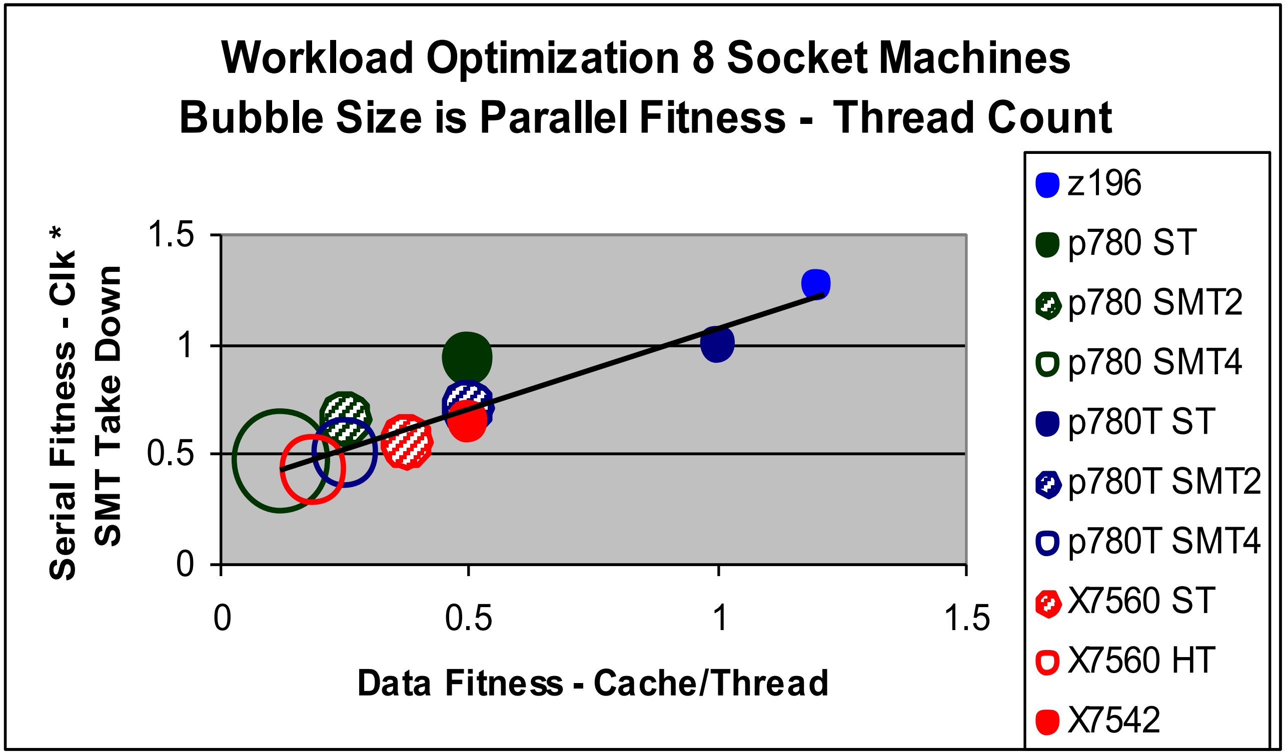

Figure 14: Temple's Assertion



# Pfisters Paradigm and the Trends



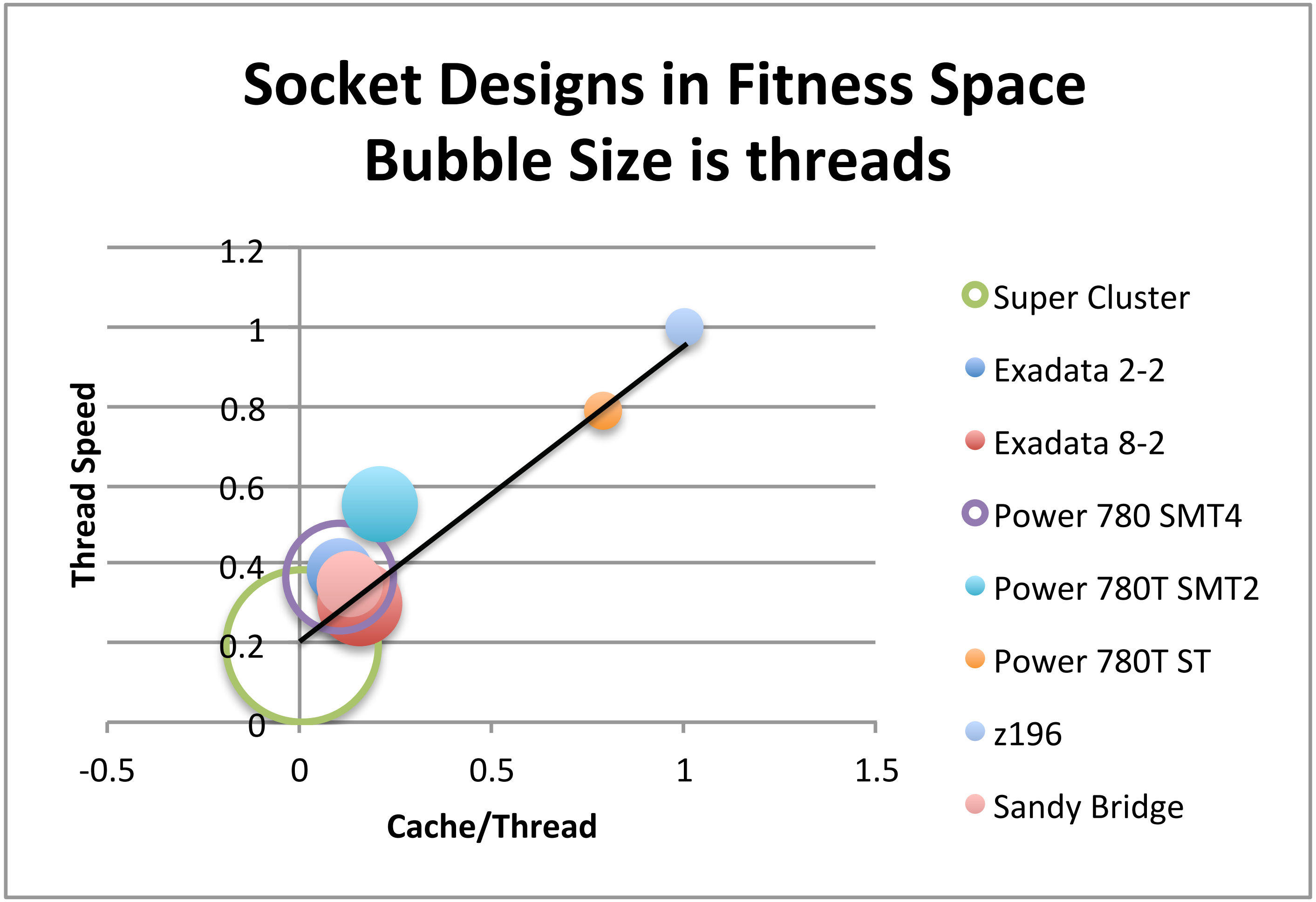
# Workload Optimization




"

There is a clear trade off at work here:

- To have more threads you must give up thread speed and cache/thread
- Machine capacity metrics govern how that tradeoff is made. In turn the metrics are designed for the "style" of computing used by each machine's base market.



Socket Designs in Fitness Space
Bubble Size is threads
Thread Speed
1.2
1
0.8
0.6
0.4
0.2
0
-0.5
0
0.5
1
1.5
Cache/Thread
Super Cluster
Exadata 2-2
Exadata 8-2
Power 780 SMT4
Power 780T SMT2
Power 780T ST
z196
Sandy Bridge

# Throughput and Capacity

***Relating Fitness to workloads***



- We observe:

  Throughput ~ Thread Count x Thread Speed

- Also:

  Thread Capacity ~ Cache / Thread x Thread Speed

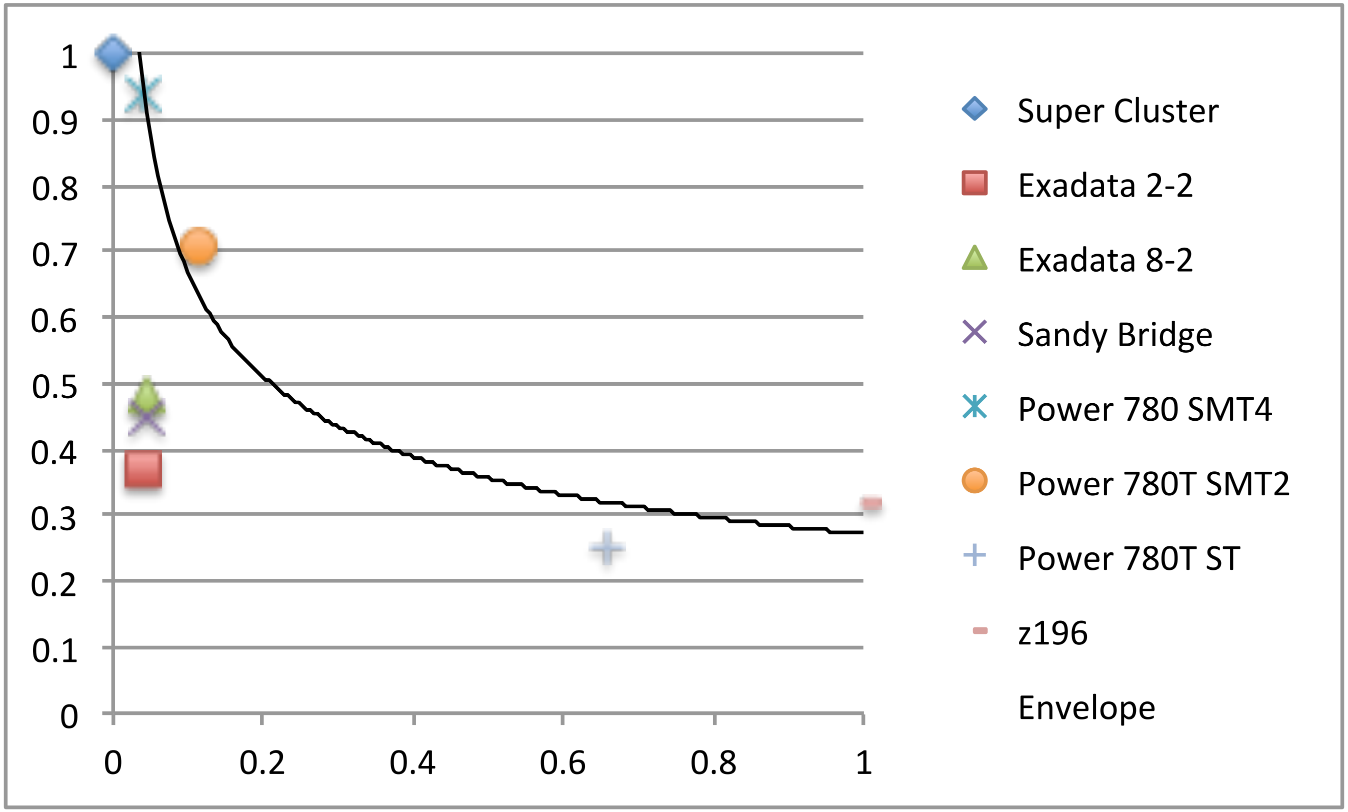
- We Assert:

  Performance ~ Thread Capacity and Throughput

- We Define:

  Performance = w(Thread Capacity) +(1-w)(Throughput)

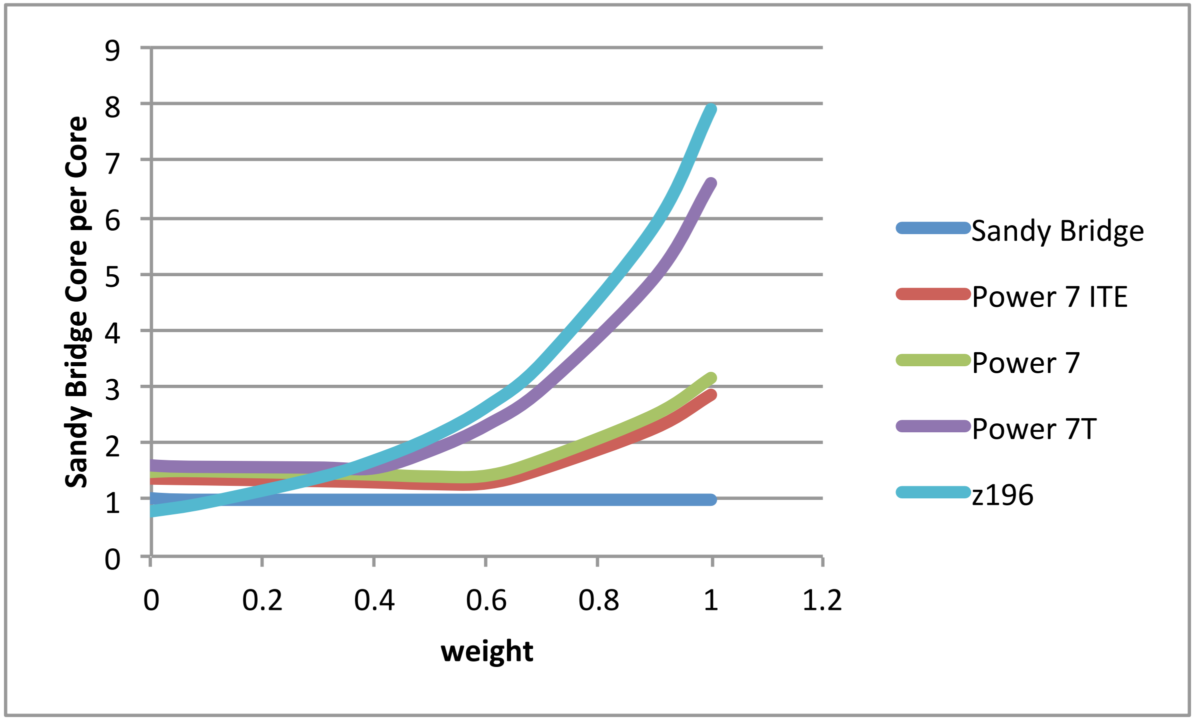
# Machines have different Throughput and Thread Capacity




Note that Very High Throughput → Very Low Thread Capacity (and Vice Versa)

Therefore to achieve Very High Throughput workload must have Very Low Weight

# Single Core Relative Capacity




Note that relative capacity *is not linear with weight.*

# Local factors take the form of Operational Trade offs

- Operations governed by "Normalized Headroom"

  - $HR = (1-u)/u = c^2 N t_0 / t_{wait}$
  - $HR(avg) = kcN^2$ => (SLA)(Variability)(Scaling)
  - $U = 1/(1+HR)$
  - $t = t_0 + t_{wait}$
  - $t_{wait} = (t_0)(c^2N)(u/(1-u)) = (t_0)(c^2N)/HR$
    = (1/weighted capacity)(variability)(scaling)/HR
    M/G/1 system

$T_0 \sim 1/\text{Capacity}$

# What are you Optimizing?

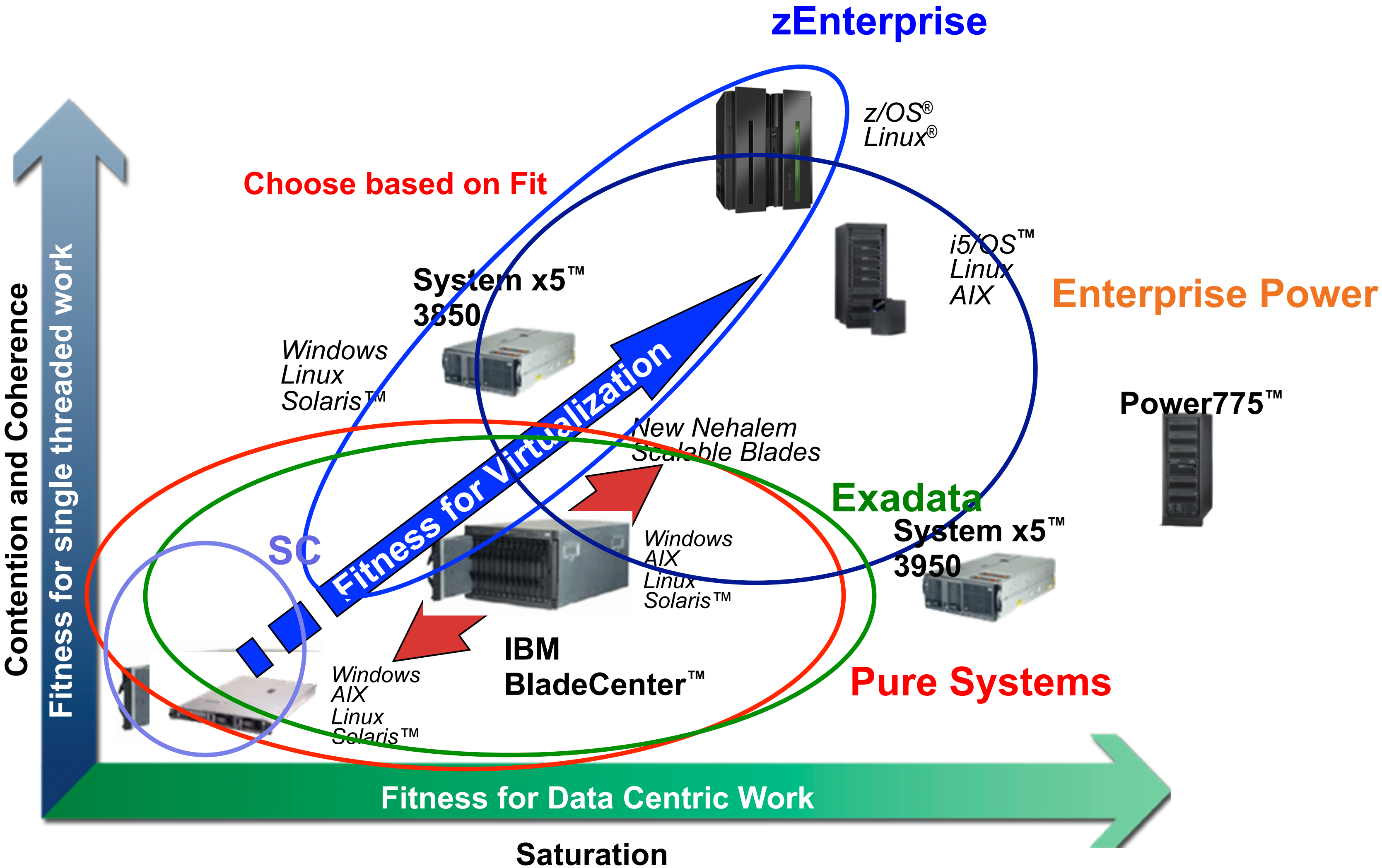
Is t measured at completion of each user thread?Look at Thread Capacity v Throughput

Want efficiency with good enough response time? Look for Thread Capacity

Or is t measured at completion of many threads? Look for Throughput

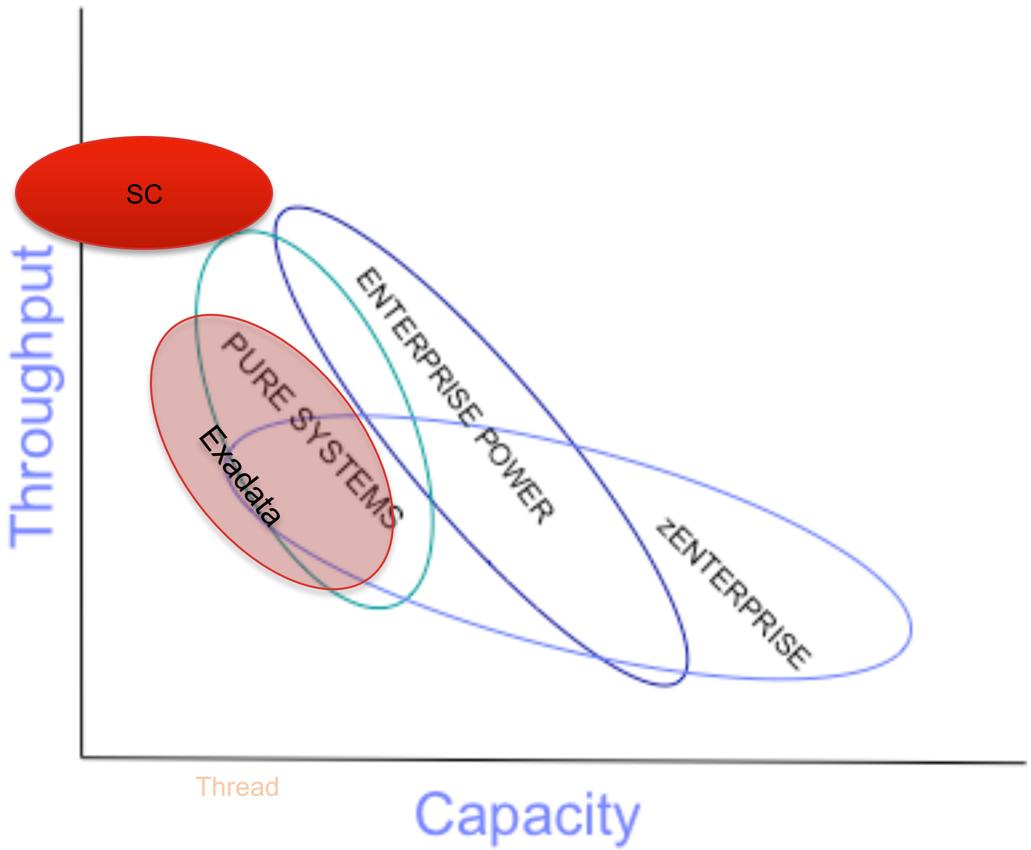
Can “wait time” tolerance be bought by reduction of network latency? Trade off wait time for efficiency; look for Capacity

# IBM Integrated Systems cover the fitness space and key legacies




Federation bridges gaps and also avoids stretching processor brands

# Positioning with Throughput and Thread Capacity




Today Pure Systems is a match for workloads with lower weight.