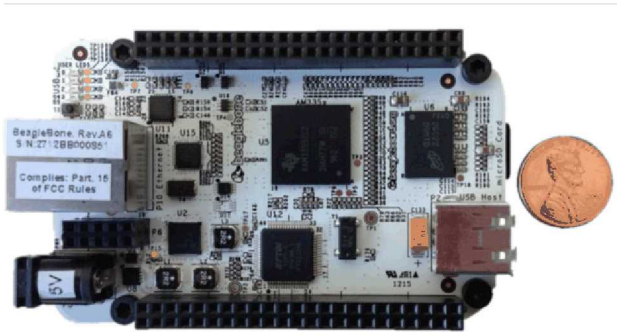
SAND2015-7838C



Re-evaluating Network Onload vs. Offload for the Many-Core Era

Matthew G. F. Dosanjh, Ryan E. Grant,
Patrick G. Bridges, Ron Brightwell

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Onload vs Offload

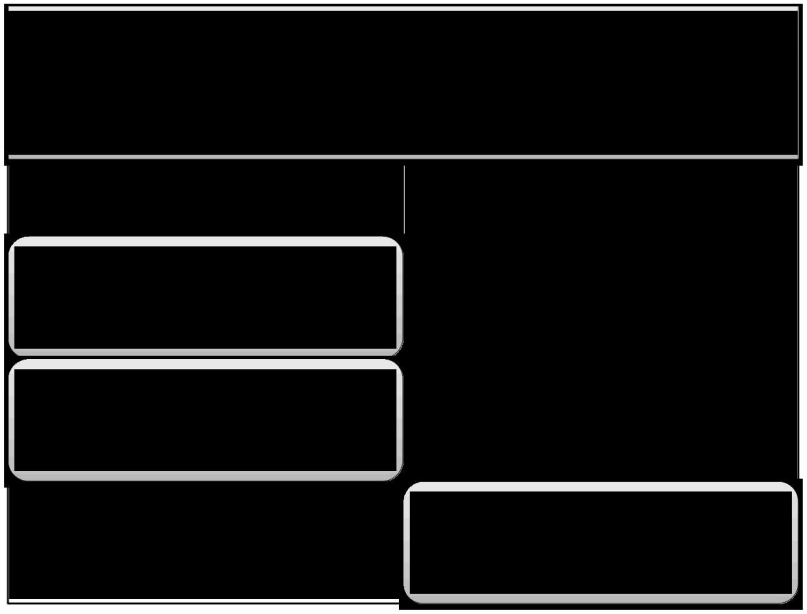
Sandia National Laboratories

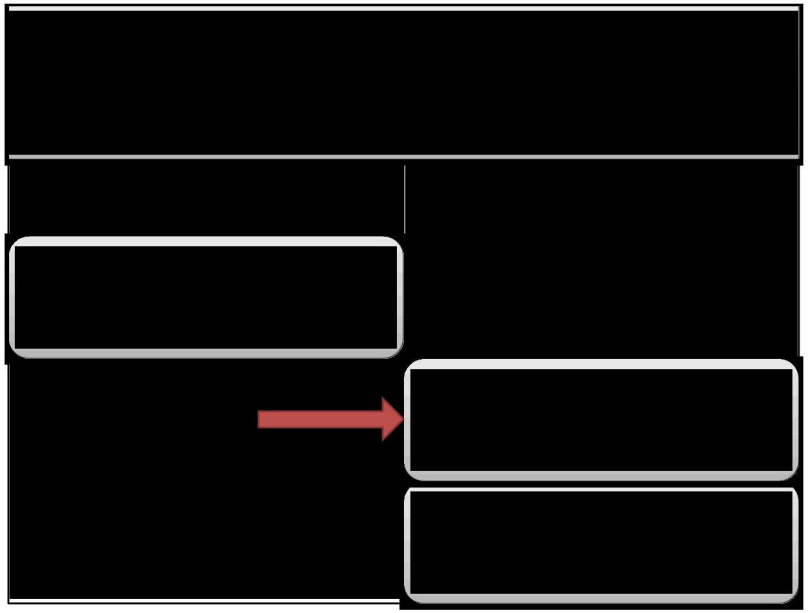
- A highly contentious debate
  - Proponents of Onload
    - Onload works well with Modern Cores
    - Can't we just burn a core?
  - Proponents of Offload
    - Less interference with the CPU
- Many opinions but this is the first quantitative study comparing onload vs offload on the same underlying hardware
- Future systems
  - Many core systems
  - Power and Energy

Network Onload vs Offload

Sandia National Laboratories

Simplified version





There are varying degrees of offload, each doing a subset of the message processing on a dedicated ASIC

Slide 3

RG3 An animation might be useful here too. Note that you're only showing complete onload vs complete offload, when in reality there is typically something in the middle happening (especially with MPI)

Ryan Grant, 6/29/2015

Networking in the Many-Core Era

Sandia National Laboratories

- HPC is trending to many-core systems
- Thin Cores
  - Lower Clock Speed
  - Lower Complexity (In-Order or Limited Out-of-Order)
  - More Parallelism
- Network Processing
  - Often Serialized
  - Onload is Sensitive to Processor Speed & Complexity
- Can Network Offload Alleviate This?
  - Current Offload Cards Use a Dedicated Low Frequency ASIC
- How Do Onload and Offload Compare in Performance?

Energy and Power Concerns

Sandia National Laboratories

- Power Caps
  - Limit on Instantaneous Power Usage
  - Capability-Class Supercomputers
  - Is There Enough Power to Run This Application?
- Energy Budgets
  - Financial Constraint Over Time
  - Capacity Systems
  - How Can We Save Money?
- How Much Power Do Onload and Offload Cards Use?
- What is The Trade Off?

Experimental Methodology

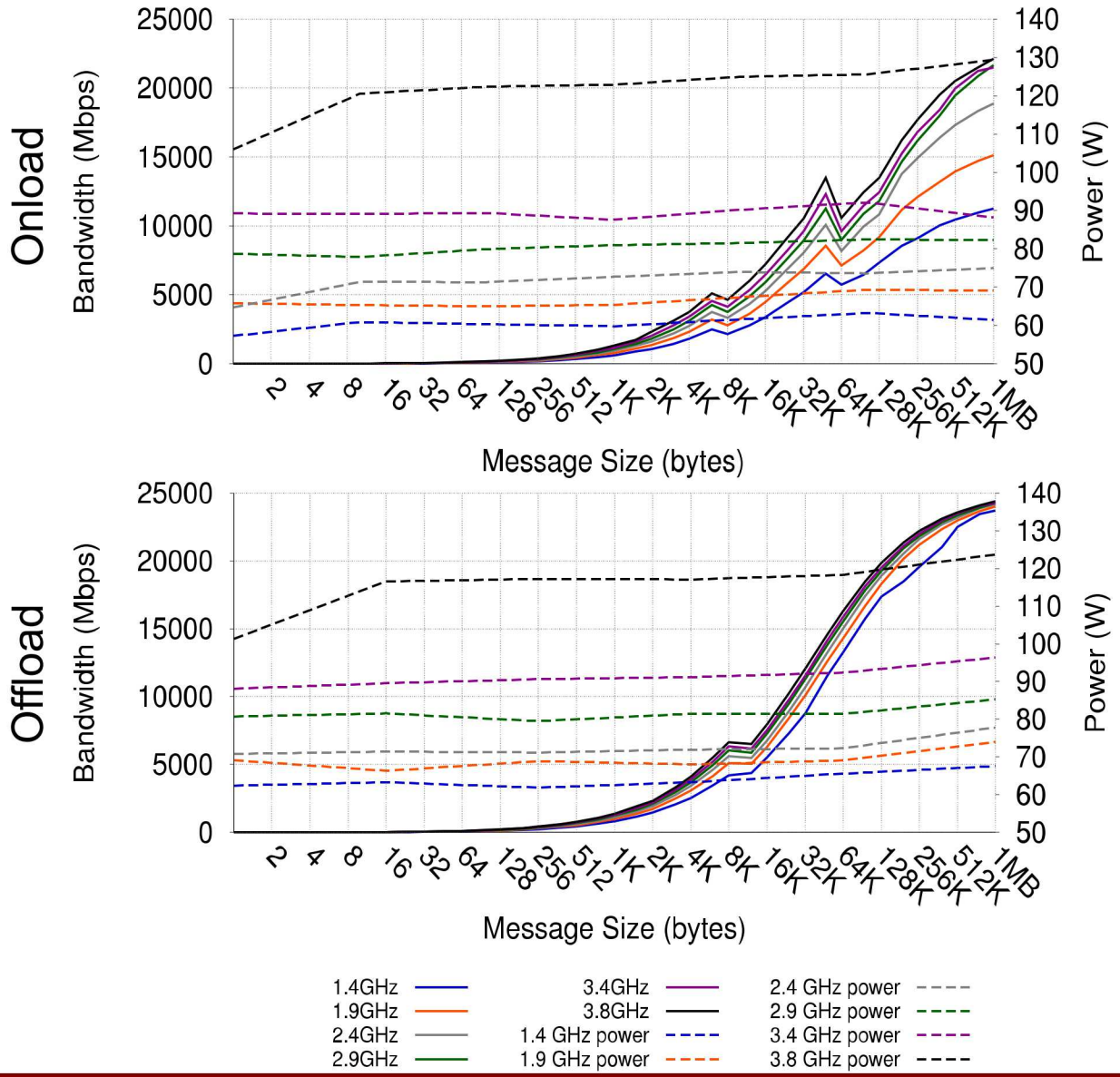
Sandia National Laboratories

- How Do We Empirically Test This Trade Off?
  - Different Network Cards
    - Controlled for Other Hardware
  - Many Core: Look at Varying CPU Speeds
  - Energy and Power Measurement
- Power Insight
  - Custom Power Measurement Devices
  - High Resolution, Out of Band Data Collection
- Teller Testbed Cluster
  - 3.8 GHz AMD Fusion APU
  - Dynamic Voltage and Frequency Scaling
  - Inifiniband QDR
    - Both Melonox Offload and Qlogic Onload

Sandia National Laboratories

MICROBENCHMARK RESULTS

NETPIPE Stream Bandwidth (Put) With Power



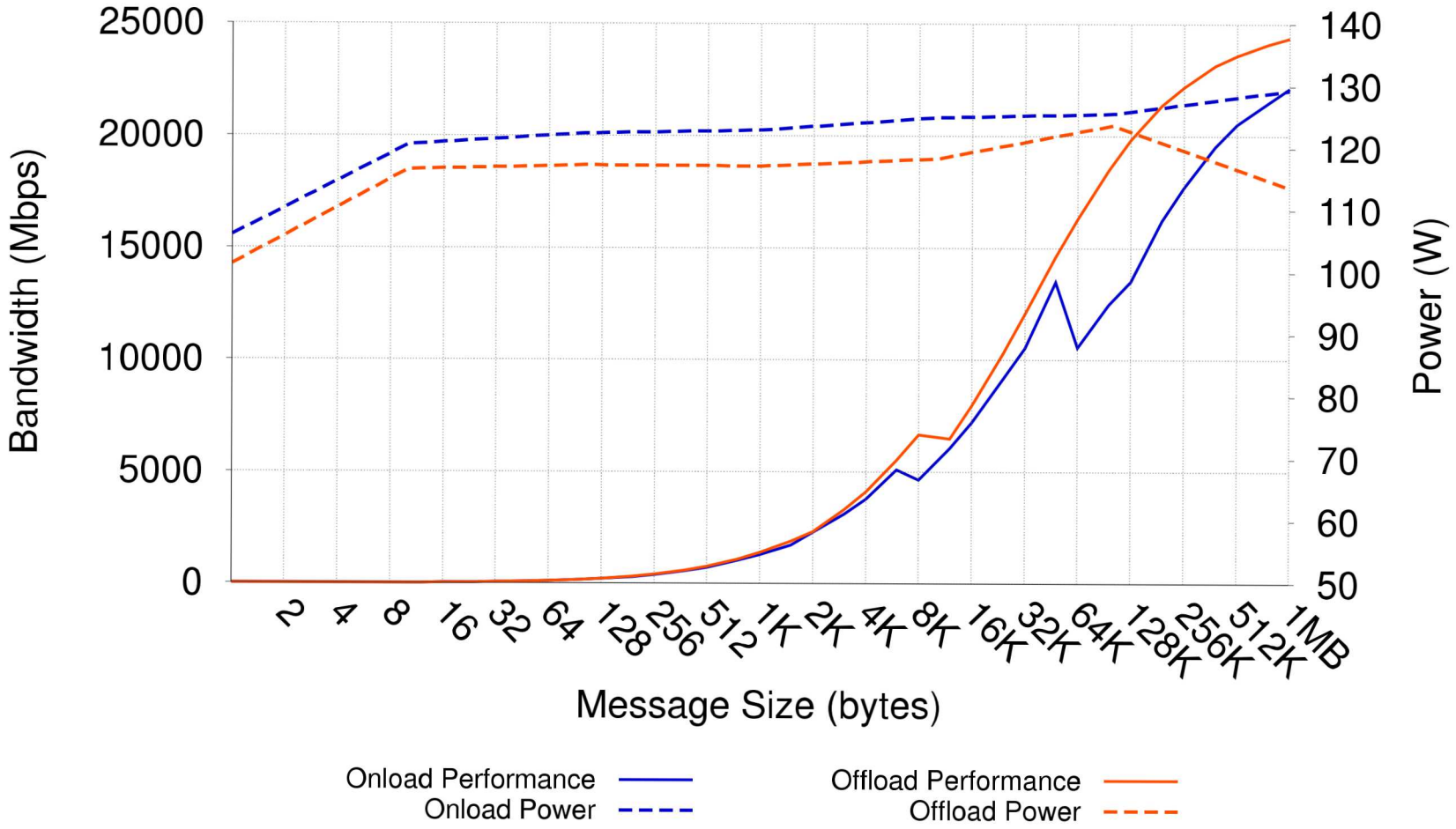
At low frequency offload outperforms onload by up to 50% in bandwidth

Offload is also less sensitive to the protocol switches at 8k and 64k

Offload bandwidth is not significantly impacted by cpu frequency

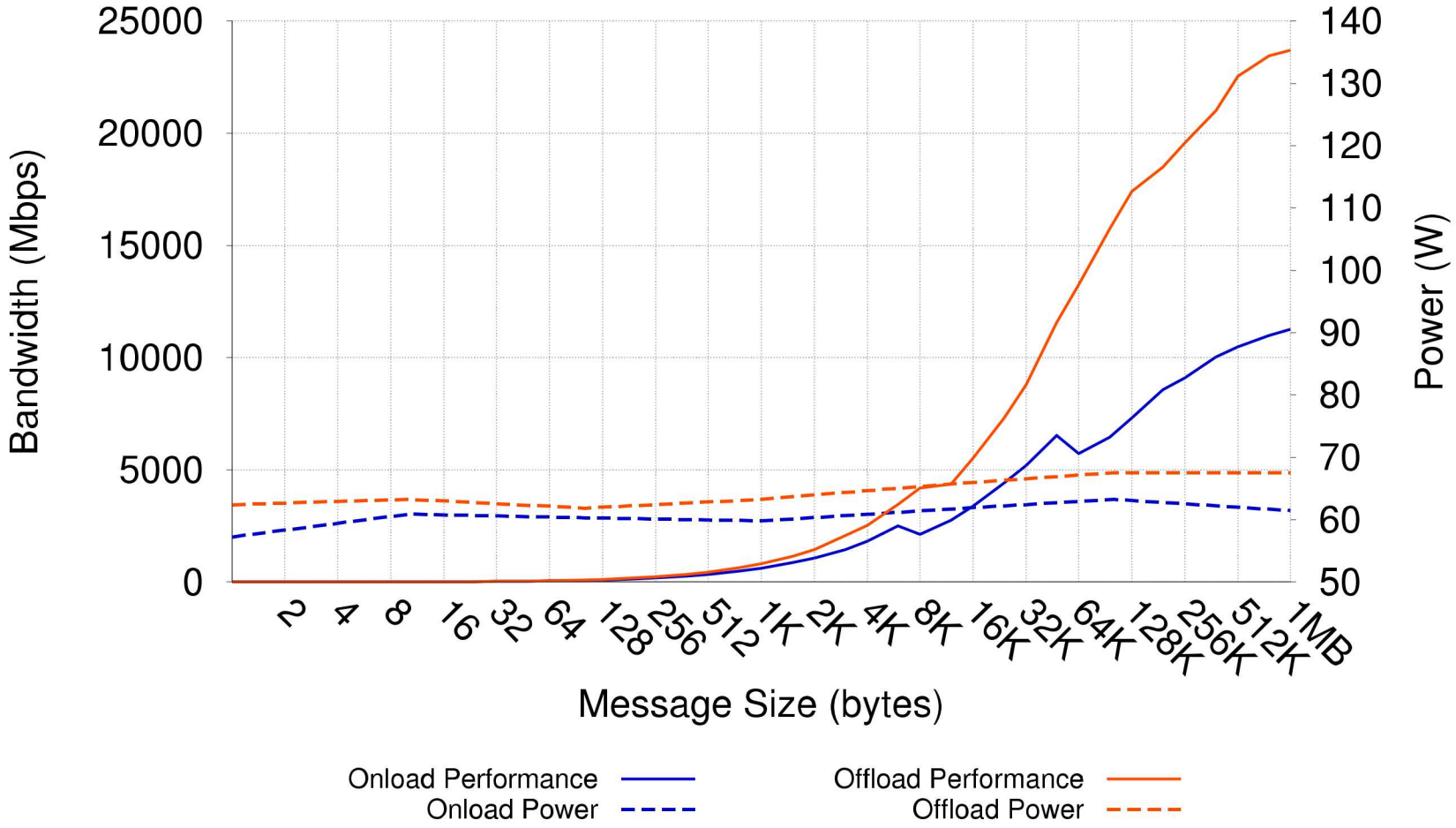
Stream (Put) Comparison – 3.8 GHz

Sandia National Laboratories

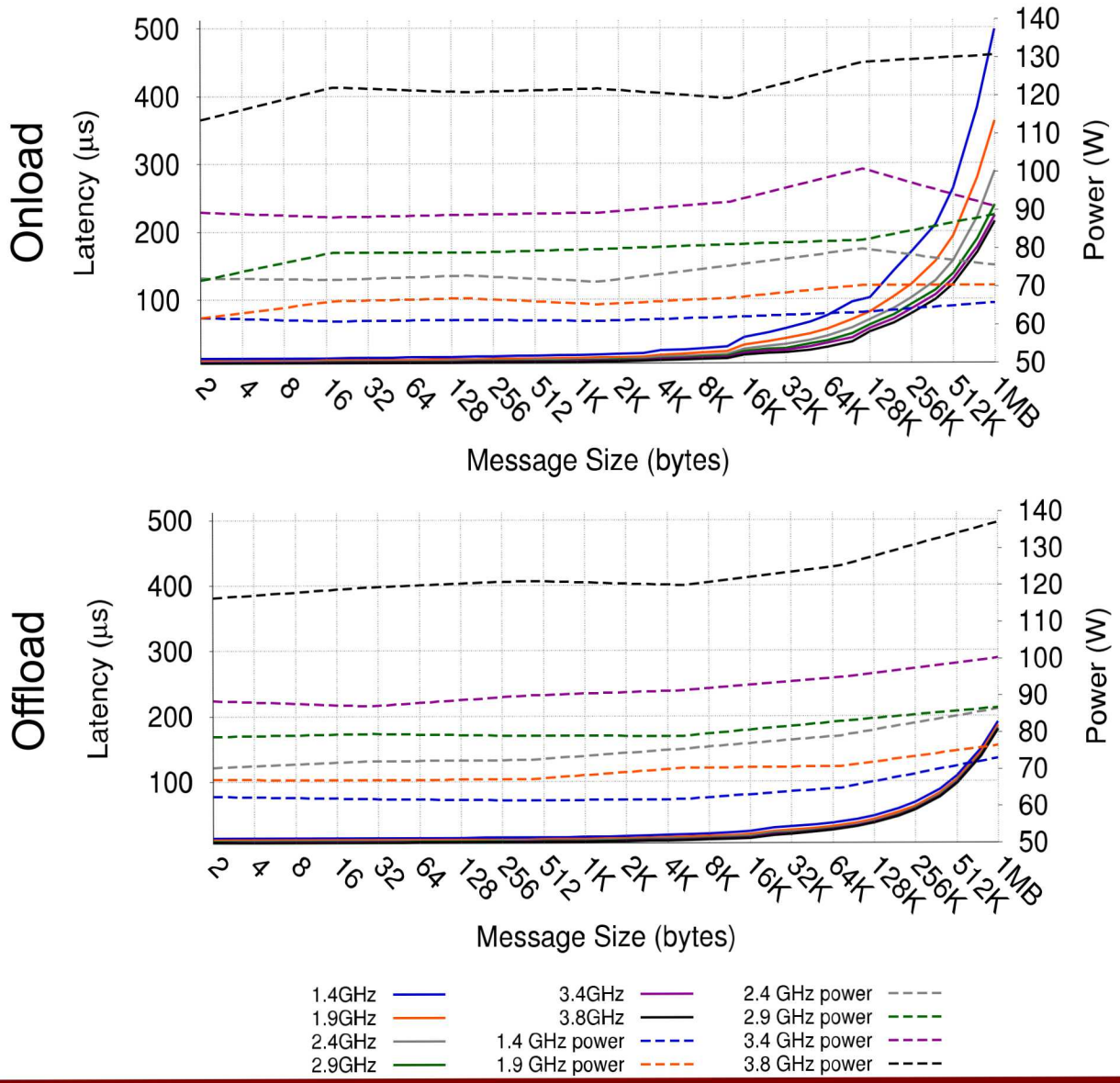


Stream (Put) Comparison – 1.4 GHz

Sandia National Laboratories



Ping Pong (Latency)



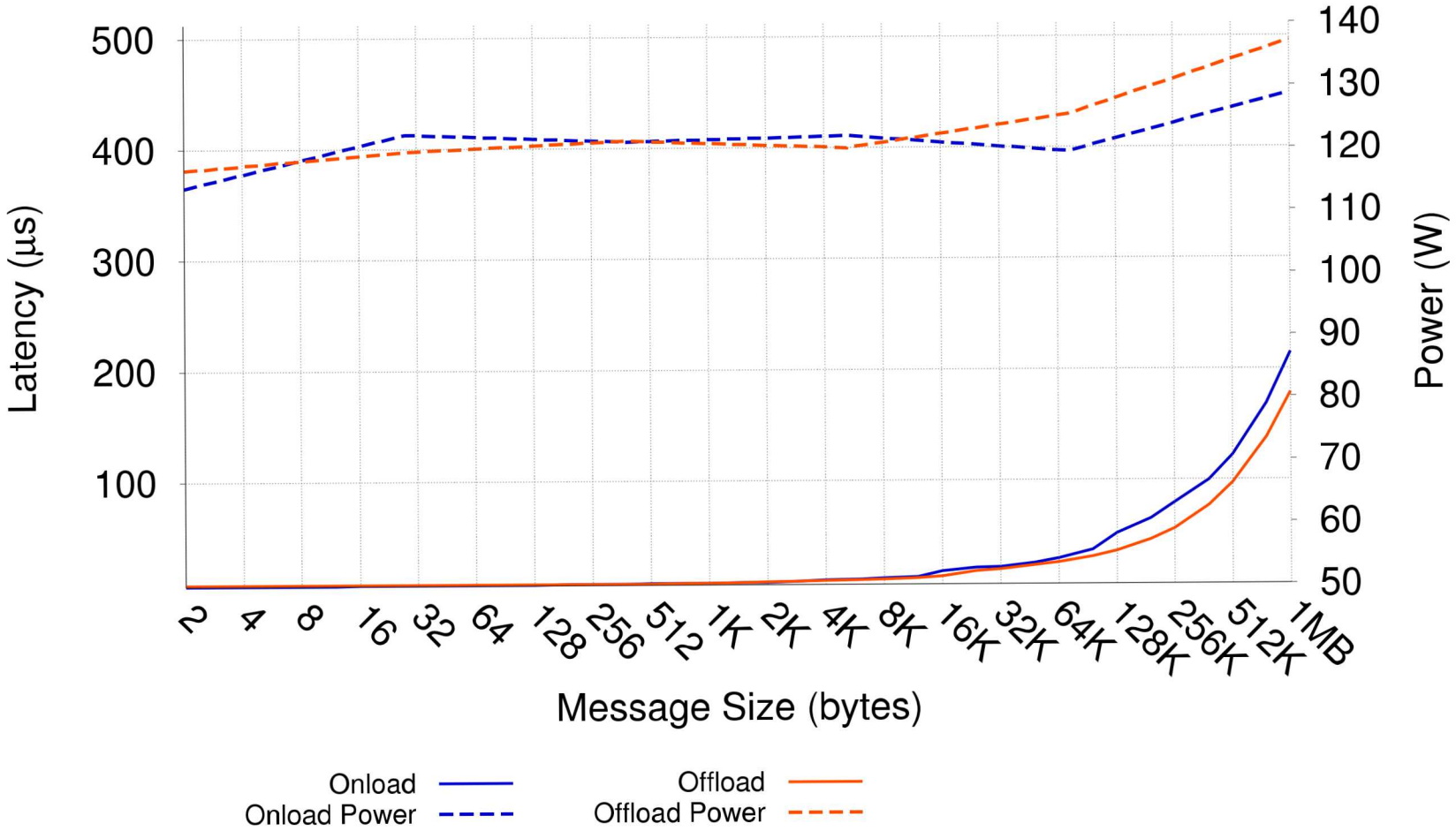
At high frequency there is little difference between latencies

At low frequencies Offload has 40% the latency of onload at 1MB Messages.

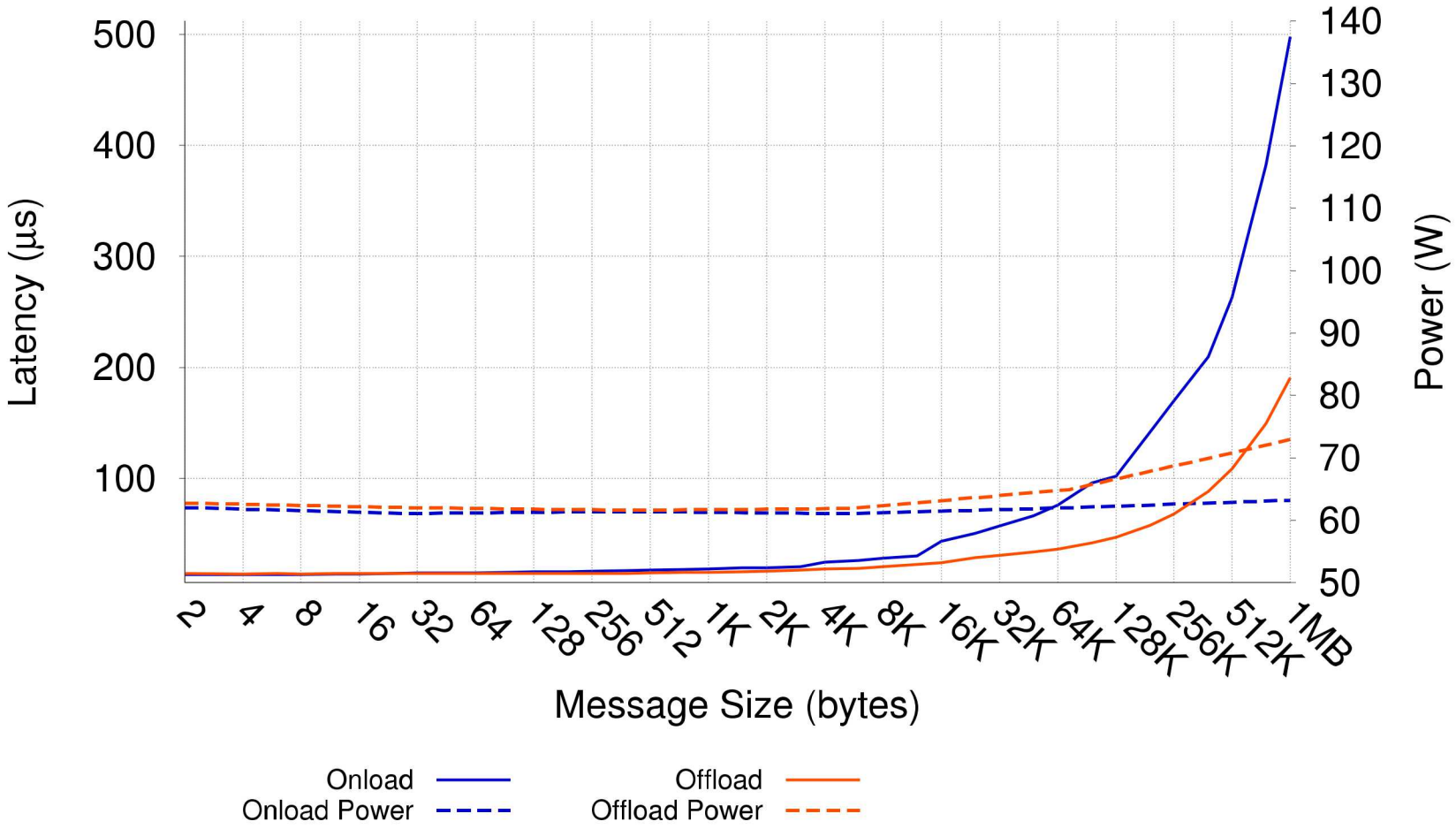
For large messages offload is a clear winner

Power increases with offload for larger messages

Ping-Pong – 3.8 GHz



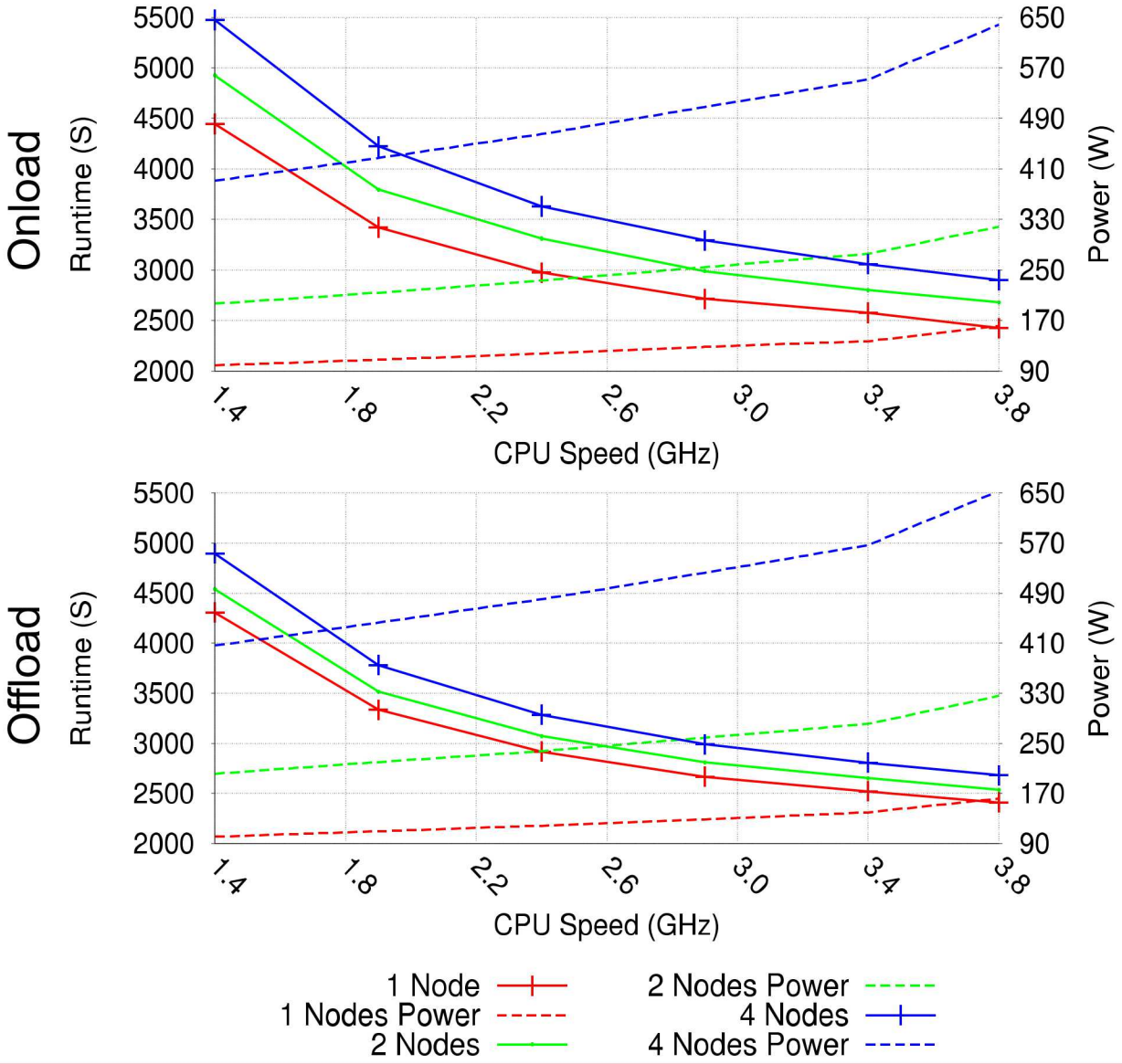
Ping-Pong – 1.4 GHZ



Sandia National Laboratories

APPLICATION RESULTS

MILC Runtime With Power



Lattice computation

Weak Scaling Problem Size

Offload reduces runtime by 7.7% to 10.6% at 4 nodes

PoweRG13esn't significantly change

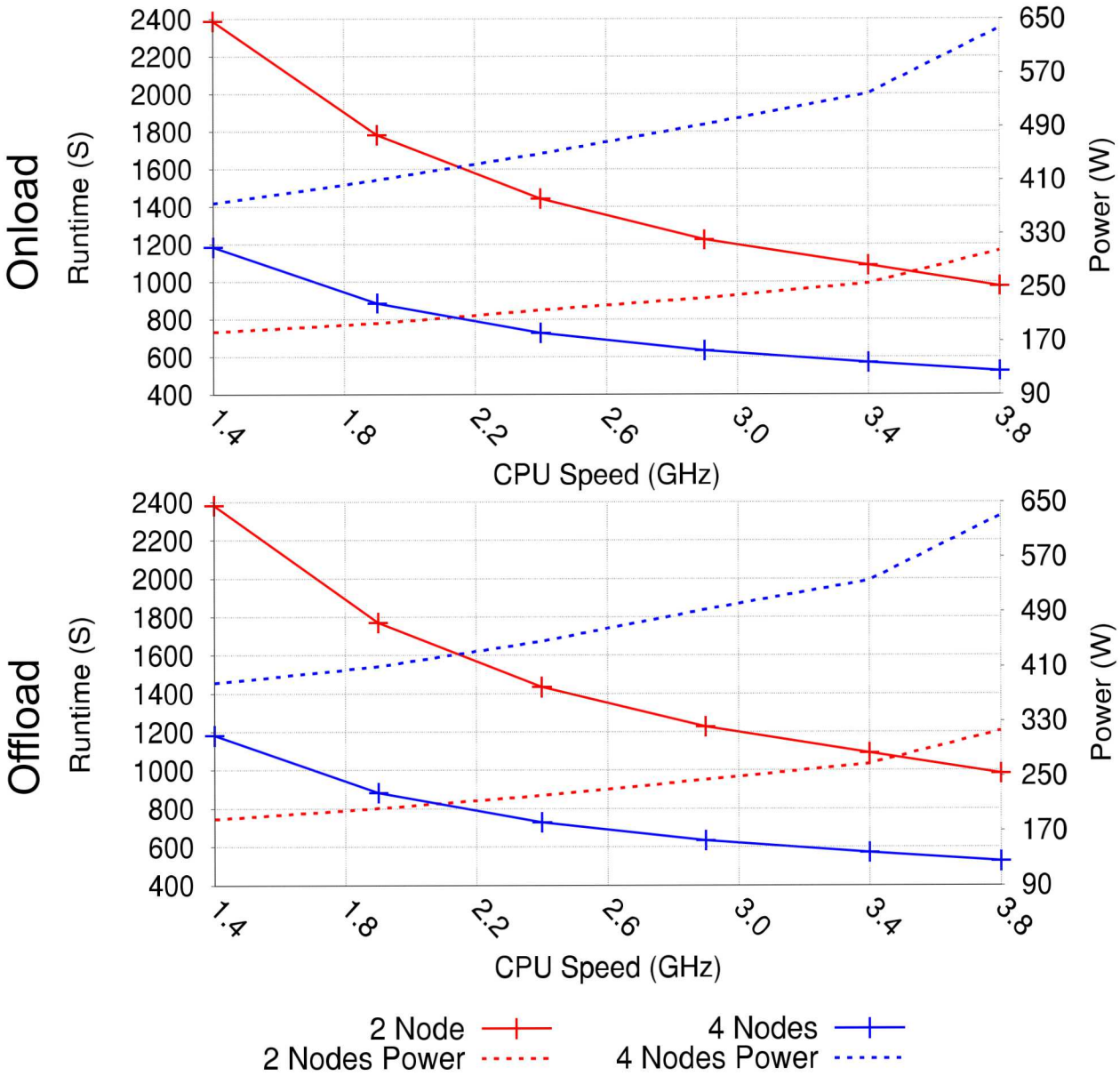
Slide 15

RG13 Do you have any more insight? At least include the percentage differences. Here you need a teaser for the upcoming explanations.

Ryan Grant, 6/29/2015

LULESH Runtime With Power

RG26



A hydrodynamics proxy

Strong Scaling Problem Size

Little performance difference

Power doesn't significantly change

RG26 You mean LULESH

Ryan Grant, 6/29/2015

What's the difference?

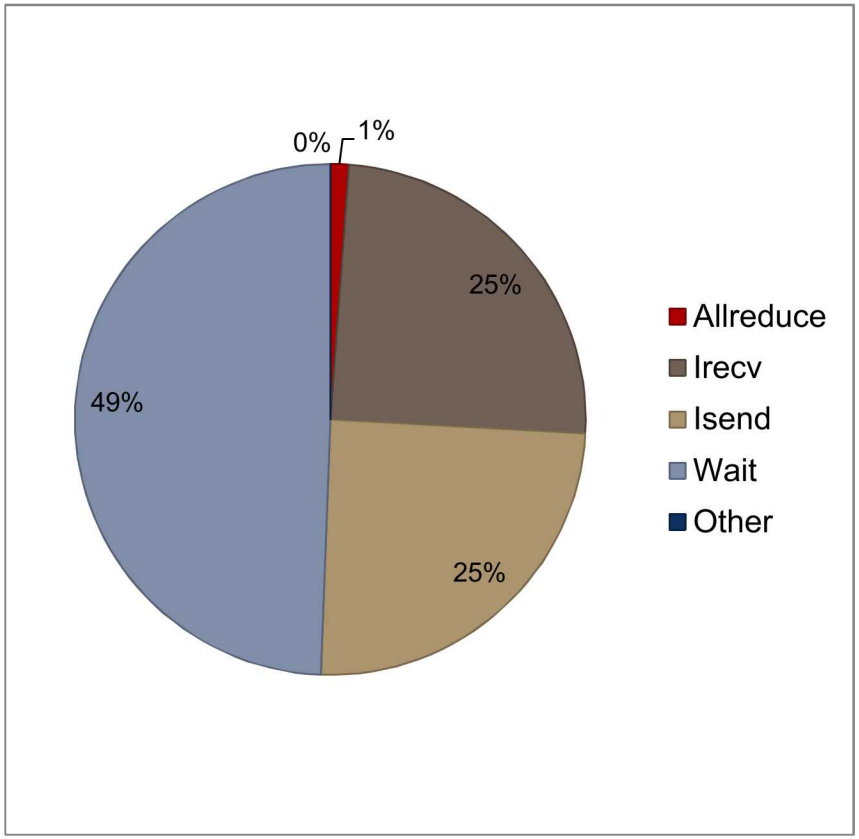
Sandia National Laboratories

- Why does MILC's performance differ while LULESH's does not?
  - Does one spend more time communicating?
  - Does one send more messages?
  - Can we gain insight here by profiling the apps?
- We gathered MPI profiling data using MPIP to gain look into probable
  - Callsites
  - Visit rate for the callsites
  - Communication function of each callsite
  - Time spent at each callsite

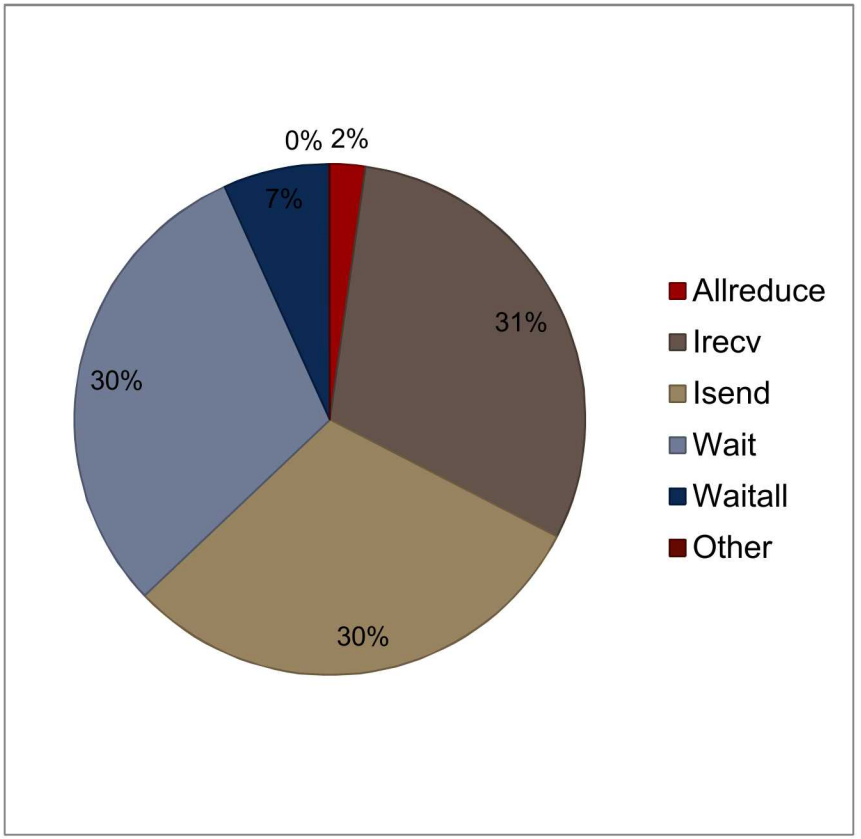
Distribution of MPI Calls

MILC Makes 17x more MPI calls per second than LULESH

MILC



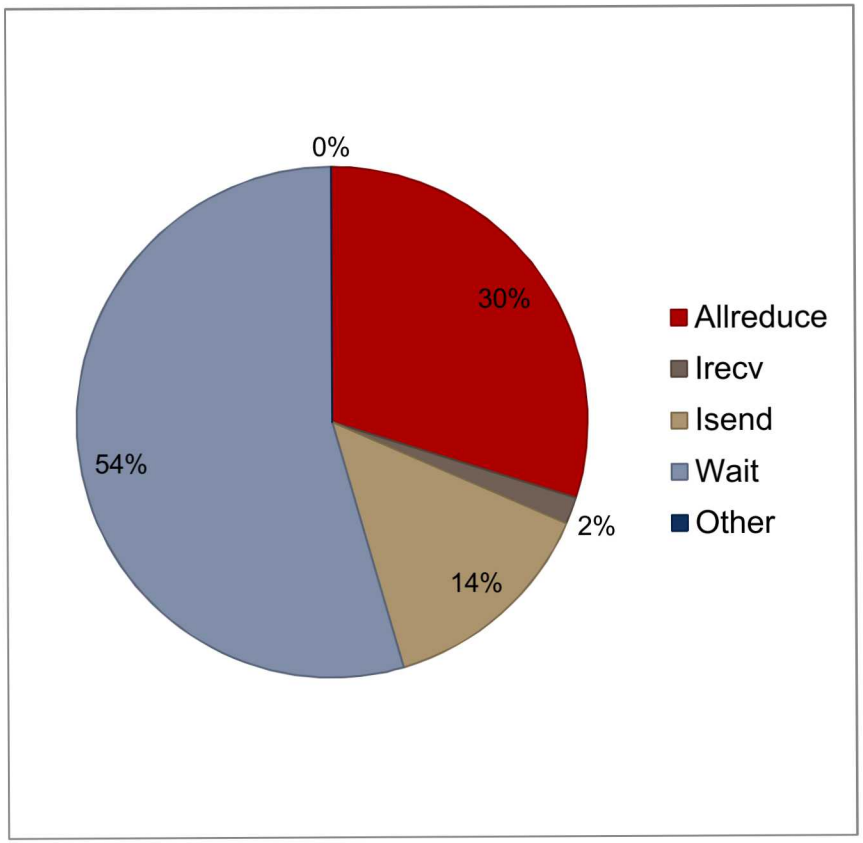
LULESH



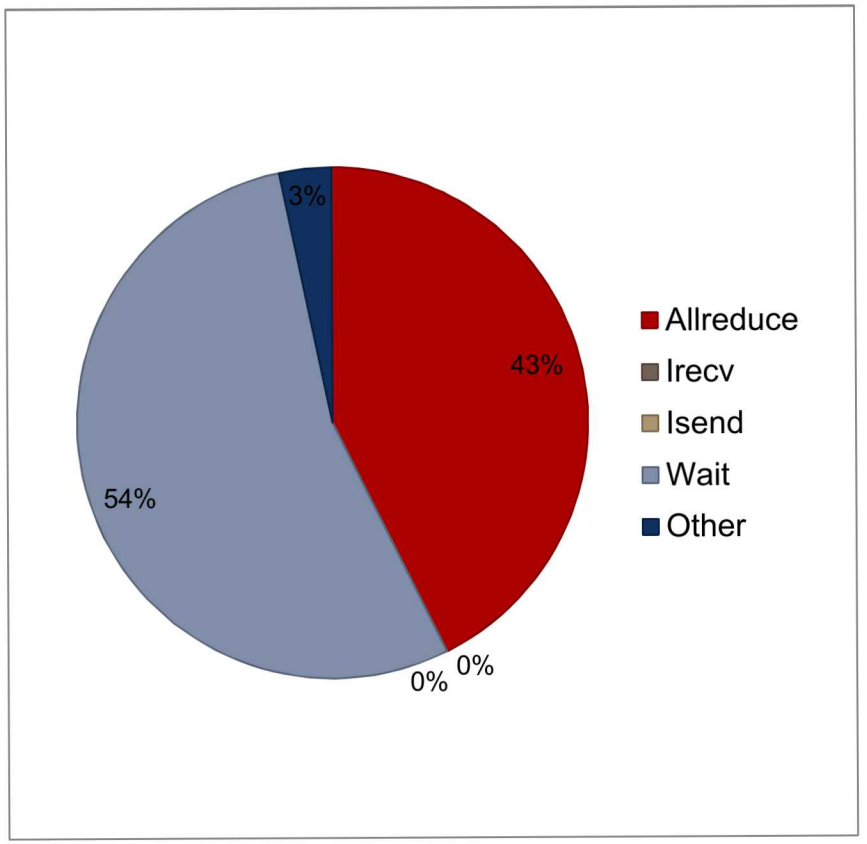
Distribution of Time In MPI for Onloaded RNICs

MILC Spends 1.2x more time in MPI per second than LULESH

MILC



LULESH



What Do These Results Show?

Sandia National Laboratories

- MILC sees significant speedup with offload
  - Speed-up of 7.7% to 10.6% at 4 nodes.
- LULESH doesn't see a significant change
- The power consumption doesn't vary much between the different cards
- MILC has more communication processing than LULESH
  - 17x more MPI calls per second
  - 1.2x more time in MPI per second
- MILC is more point-to-point bound than LULESH
  - Spends significantly more time in irecv, isend, and wait

Conclusions

Sandia National Laboratories

- Offload is better for many-core systems
  - Offload shows a significant performance improvement, when there are a significant amount of calls into MPI
    - We expect more fine grained communication
      - Multiple threads
      - Communication and computation overlap
      - Over-decomposition of problems
  - Offload cards use an insignificant amount of extra power in applications

Conclusions

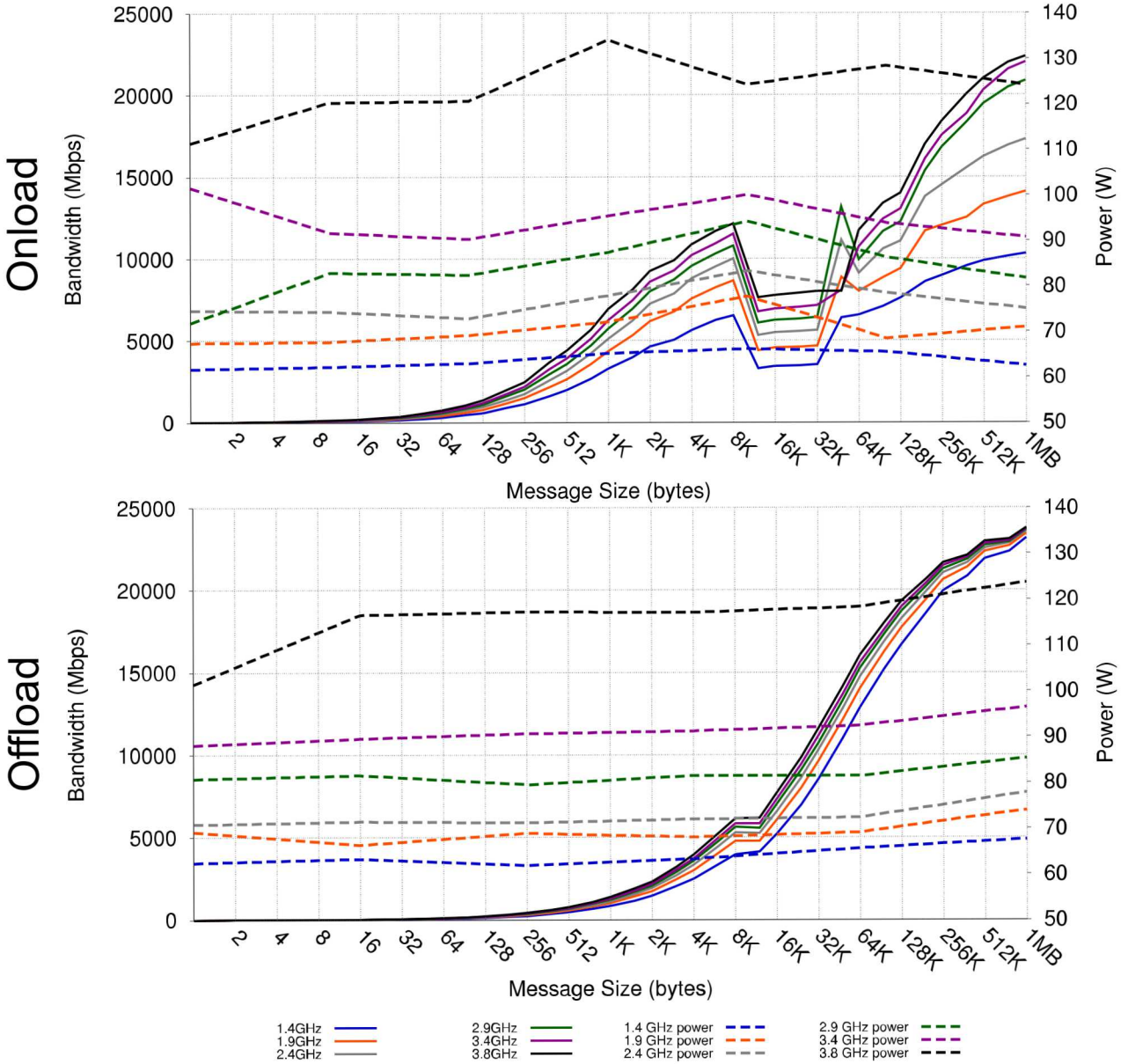
Sandia National Laboratories

- What about for other systems?
  - Offload beats Onload performance at
    - Lower CPU speeds
    - Higher rates of communication calls
  - Offload still uses an insignificant amount of extra power
    - The performance benefit means less energy use per application
    - For power caps offload is probably not issue
      - If a system runs very close to it's power cap, an extra <2% per node could be an issue

Sandia National Laboratories

QUESTIONS?

Stream (Without Cache Effects)



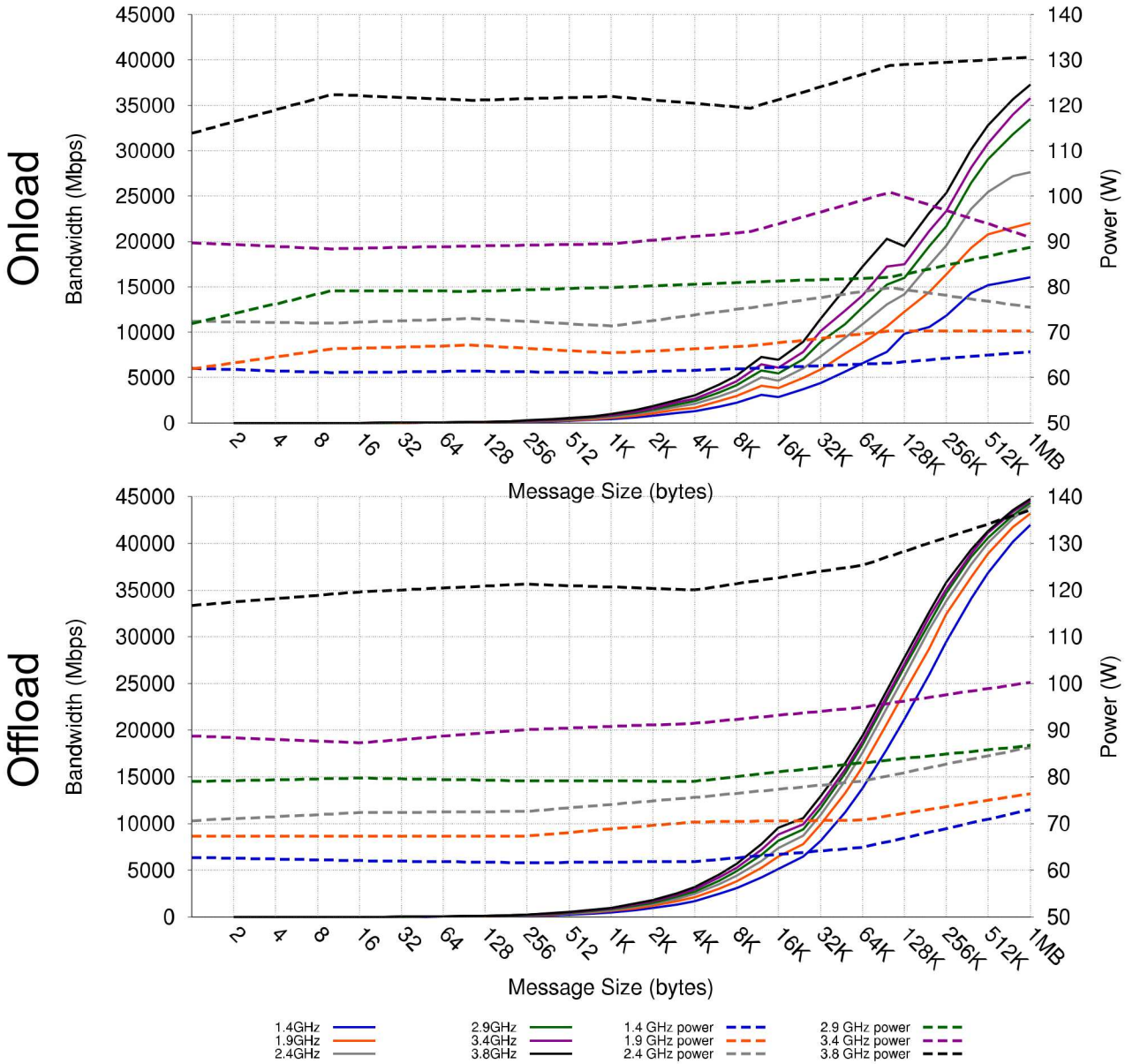
Onload shows sensitivity to not having cache effects in two ways;

First, the overall performance is lower

Second, because of this the protocol switch at 8k shows lower performance, until we use more memory than the L1 at 64k

Offload is less sensitive to cache effects

Ping Pong (Bandwidth)

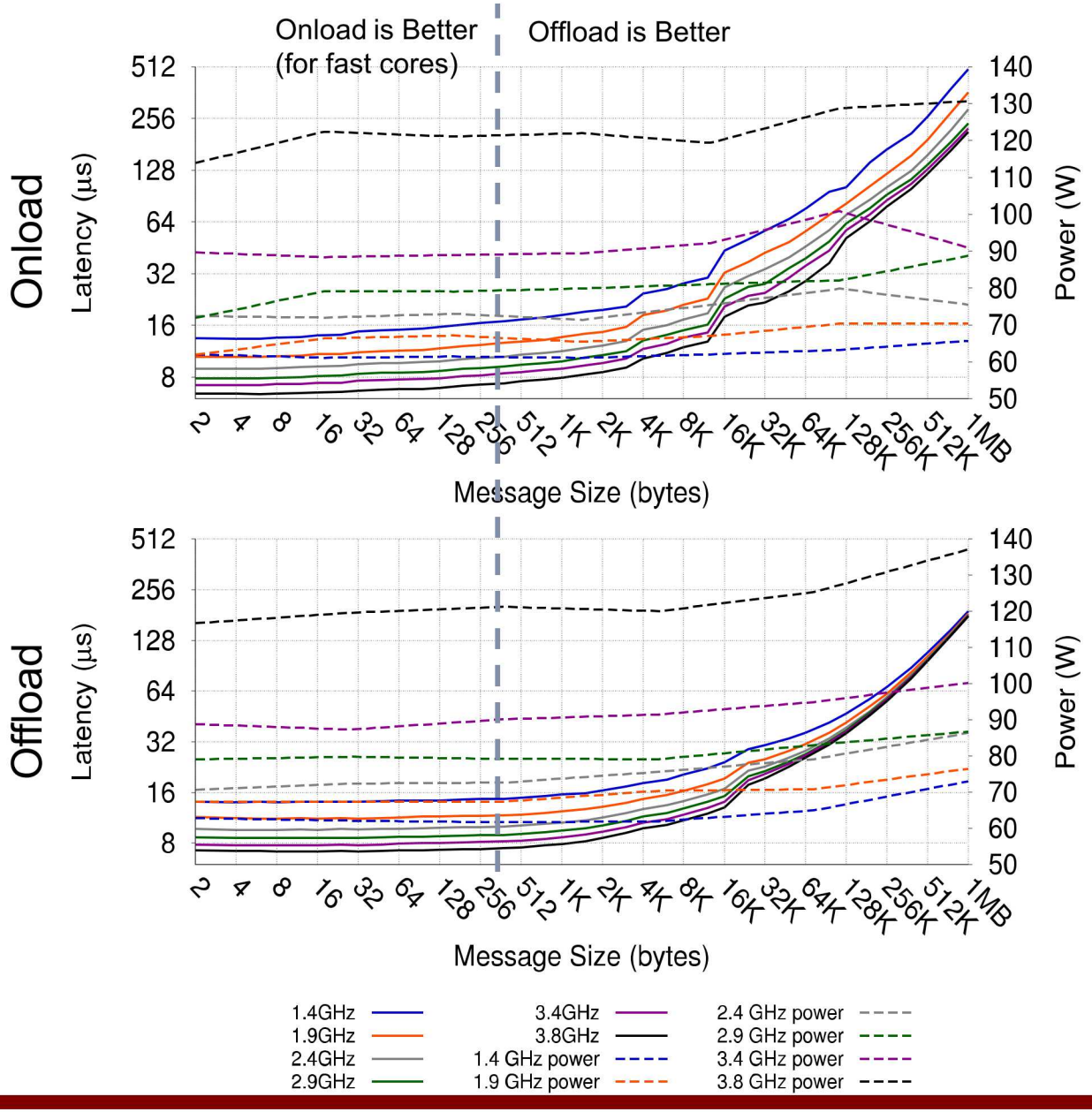


Looking at ping pong in terms of bandwidth illustrates how sensitive onload is to cpu speeds

In the onload results, the degradation appears to grow linearly with message size.

Looking at offload, we can see that there is a much less significant degradation in performance.

Ping Pong (Latency)



At small message sizes, Onload can be better.

For large messages offload is a clear winner

Power increases with offload for larger messages