High Performance Computing

2015/02/02

Kazuki Tsuzuku

Endo lab.

Review Paper

Title:

Energy Consumption of Resilience Mechanisms in Large Scale Systems

Author:

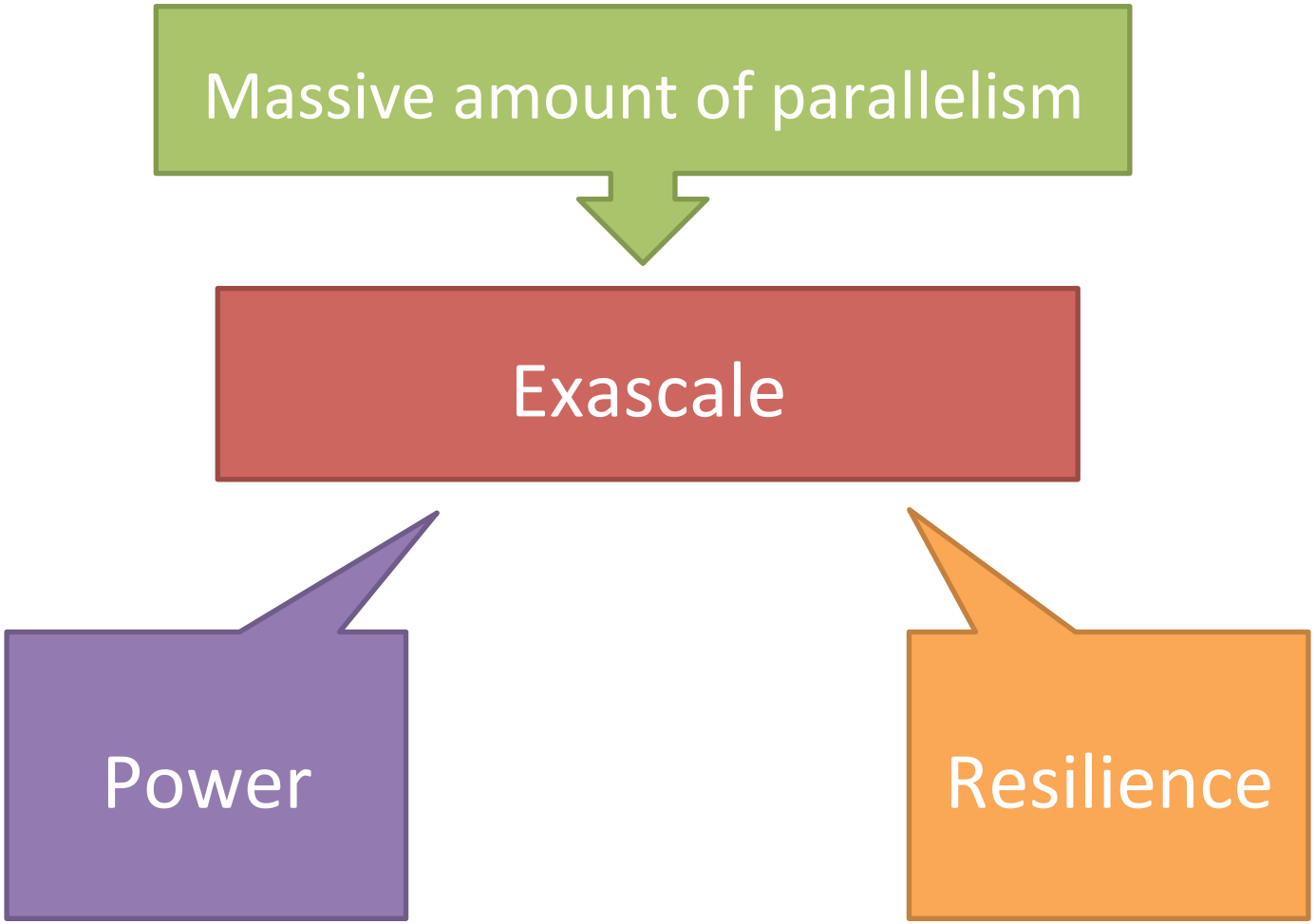
Bryan Mills,Taieb Znati,Rami Melhem

Kurt B. Ferreira and Ryan E. Grant

Published in:

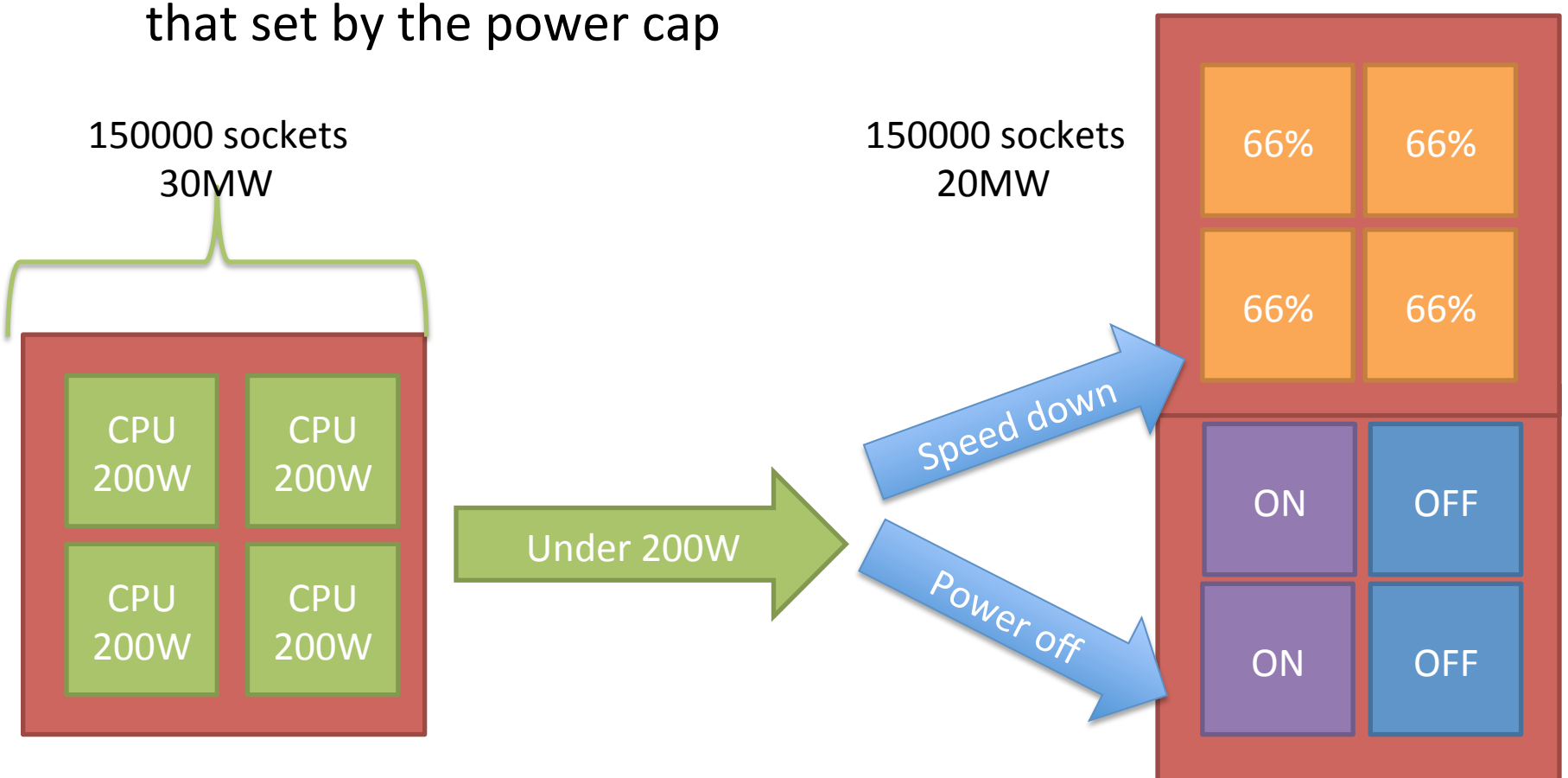
Parallel,Distributed and Network-Based Processing (PDP),2014 22nd Euromicro International Conference on

Challenges of Exascale



Power of Exascale

- Established by the DOE at no more than 20MW
  - Top1 Tianhe-2 33.9PFLOPS 17.8MW
- The machines will be capable of consuming more power than that set by the power cap



Resilience of Exascale

- The number of components will grow
  →system failures will become routine
- Resilience scheme must consider its effect on the application's energy and power consumption
- Reduce considerably the overall performance

Objectives and achievements

Related Work

- Energy considerations in checkpointing and fault tolerance protocols[M. el Mehdi Diouri, et al. 2012]
  - Measure energy consumption of the three main tasks associated with checkpoint-restart methods
- Assessing Energy Efficiency of Fault Tolerance Protocols for HPC Systems[Esteban Meneses, et al. 2012]
  - Evaluate the energy consumption for three different checkpointing technique
  - Find that uncoordinated checkpointing with parallel recovery was the best technique
  - Show that as the number of sockets grows beyond 25600 the trend in energy savings of parallel recovery is decreasing

↔This work shows that replication increase energy saving as the system size grows

Resilience Methods

- Coordinated Checkpoint/Restart
- Uncoordinated Checkpointing
- Traditional Replication
- Replication Optimizations

Coordinated Checkpoint/Restart

- All running processes periodically pause their execution and write their state to a stable storage device
- In the event of a failure,all processes restore from the checkpoint and resume execution
- This methods are the most widely used fault tolerance method in HPC

Uncoordinated Checkpointing

- Improve the performance of checkpointing systems
  - Assumes the availability of local storage
- Nodes checkpoint and restore from local storage without the synchronization
- Result in cascading rollbacks

Traditional Replication

- Each application process is replicated on independent computing node
- Replication in HPC has largely been dismissed
  - Additional resources required
  - In this paper, proposed replication techniques use fewer resources

Replication Optimizations

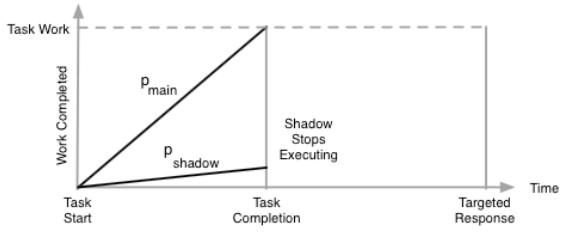
- Power-aware optimizations to traditional replication
  - Stretched Replication
  - Shadow Replication

Stretched Replication

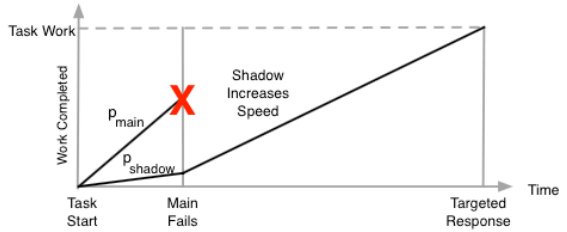
- Assume that performing work slowly can save energy
- Slow down the execution of all processes
  - To satisfy power limits
  - To increase reliability

Shadow Replication

- performing work slowly can save energy
  - This is not always the case in today's computers
    - "overhead" power 60~67%
  - Time vs power tradeoff
- The shadow executes concurrently with the main process
  - If no failure occurs
    - The main process executes at the optimal speed
    - The shadow process executes at the reduced speed
  - If main process fails
    - The shadow process increases its speed and executes the task



(a) Case of no failure



(b) Case of failure

Analytical Framework

- Computational Model
- Power Model
- Failure Model
- Checkpointing Energy Model
- Replication Energy Model

Computational Model

- Distributed computing environment of a large number of collaborative tasks
  - The successful execution of the application depends on the successful of all tasks
- Assume the application is perfectly parallizable
  - Total amount of work “W”
  - The work is evenly divided into “N”
  - Work for each socket “$W_{task}=W/N$”
  - Speed of each socket “σ”
  - The total solution time for application when all sockets are operating at maximum speed “$T_s=W_{task}/\sigma_{max}$”
  - Targeted response time “t_{resp}”,which is the maximum time that the process will complete it’s task
  - Targeted response time as a laxity factor “α”
    - In this framework $\alpha=2.0$

Power Model

- $P=A*C*V^2*f$
  - P:Dynamic CPU power,A:chip activity factor
    C:capacitance,V:voltage,f:frequency
  - Scaling both voltage and frequency
    →$P(\sigma)=\sigma^3$
- The energy consumed by a socket executing at speed "σ" during an interval $[t_1,t_2]$
  - fixed factor for overhead power "ρ"

$$E_{soc}(\sigma,[t_1,t_2]) = \int_{t=t_1}^{t_2} (\sigma^3 + \rho\sigma_{max}^3)dt$$

- The energy consumed while writing or recovering a checkpoint
  - Fixed factor for maximum I/O power "γ"

$$E_{io}([t_1,t_2]) = \int_{t=t_1}^{t_2} (\gamma\sigma_{max}^3)dt = (\gamma\sigma_{max}^3)(t_2-t_1)$$

Failure Model

- Assumptions
  - Failures are independent events
  - Only a single failure an occur during the execution of a task
- Probability of the main task failing at time t
  - The socket MTBF "M_{soc}"

$$f(t) = \frac{1}{M_{soc}} e^{-t/M_{soc}}$$

Checkpointing Energy Model

- Total wall clock time computed by Daly
  - T_s:original total solve time,M_{sys}:system MTBF
    γ:checkpoint interval,δ:checkpoint time,R:recovery time

$$T_w = M_{sys} e^{R/M_{sys}} (e^{(\tau+\delta)/M_{sys}} - 1)(\frac{T_s}{\tau} - \frac{\delta}{\tau+\delta})$$

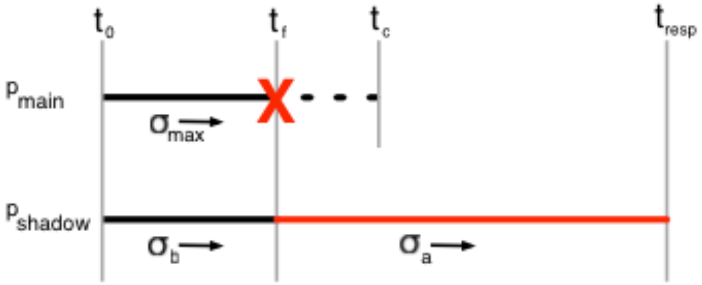
- The Energy for a single process using checkpoint and restart

$$\begin{aligned} E_{cpr} =& E_{soc}(\sigma_{max}, [0, T_w]) \\ &+ E_{io}([0, \delta)]) \times \frac{T_s}{\tau} + E_{io}([0, R)]) \times \frac{T_w}{M_{sys}} \end{aligned}$$

Replication Energy Model

- Energy consumption of the combination of replication and checkpointing

$$E_{rep} = \int_{t=0}^{t_c} (E_{soc}(\sigma_{max}, [0,t]) + E_{soc}(\sigma_b, [0,t]))f(t)dt$$
$$+ \int_{t=0}^{t_c} E_{soc}(\sigma_a, [t,t_r])f(t)dt$$
$$+(1 - \int_{t=0}^{t_c} f(t)dt)(E_{soc}(\sigma_{max}, [0,t_c]) + E_{soc}(\sigma_b, [0,t_c]))$$



  - $t_c=W_{task}/\sigma_{max}$, $t_r=t_f+(W_{task}-\sigma_b t_f)/\sigma_a$,
  - $\sigma_a=\sigma_{max}$,this is because we can trade the power consumed by the main process with the shadow process after failure of the main
- Traditional replication
  - $\sigma_m=\sigma_b=\sigma_a=\sigma_{max}$
- Stretched replication
  - $\sigma_m=\sigma_b=\sigma_a=W_{task}/T_{resp}$

Analysis

- Scaling and Failure Rates
- Scaling at Different Checkpoint I/O Rates
- Scaling at Different Overhead Power
- Energy Savings

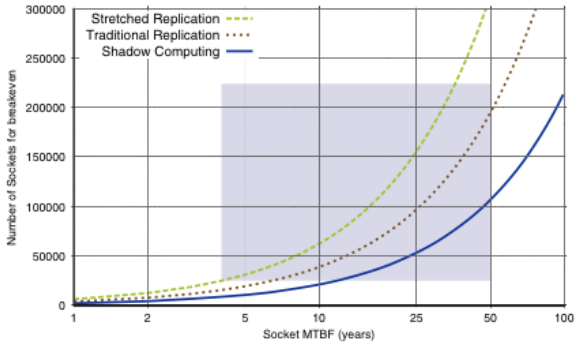
Scaling and Failure Rates

- Breakeven point at which the replication technique is equivalent to that provided by coordinated checkpointing
  - Metrics are energy and time
  - Checkpoint time is 15 minutes

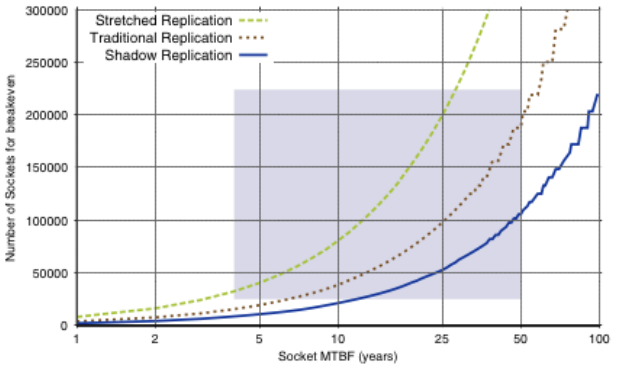
Scaling and Failure Rates

- Shadow replication can provide a significant energy saving over traditional replication
  - e.g. when MTBF is 25 years 46% energy efficiency
- Stretched replication turns out to be less energy efficient
  - Because of the increased time

- Shadow replication time is shorter than that provided by traditional replication
  - e.g. when MTBF is 25 years 46% performance improvement
  - Shadow replication can utilize additional sockets
  - Stretched replication is similar to shadow replication but both the replica and main use less power



(a) Breakeven Energy



(b) Breakeven Time

Overhead %	Method	# Sockets	# Main Sockets
60%	Checkpointing	100,000	100,000
60%	Traditional Replication	100,000	50,000
60%	Stretched $\alpha = 2.0$	153,846	76,923
60%	Shadow $\alpha = 2.0$	124,998	62,499
80%	Checkpointing	100,000	100,000
80%	Traditional Replication	100,000	50,000
80%	Stretched $\alpha = 2.0$	120,230	60,115
80%	Shadow $\alpha = 2.0$	110,636	55,318

TABLE I
AVAILABLE SOCKETS ASSUMING A 20 MEGA-WATT POWER LIMIT AND 200W PER SOCKET.

Scaling at Different Checkpoint I/O Rates

- The checkpoint write times have a significant effect on the efficiency of coordinated checkpointing
- Shadow replication is viable for all
  - But very extreme levels of I/O bandwidth

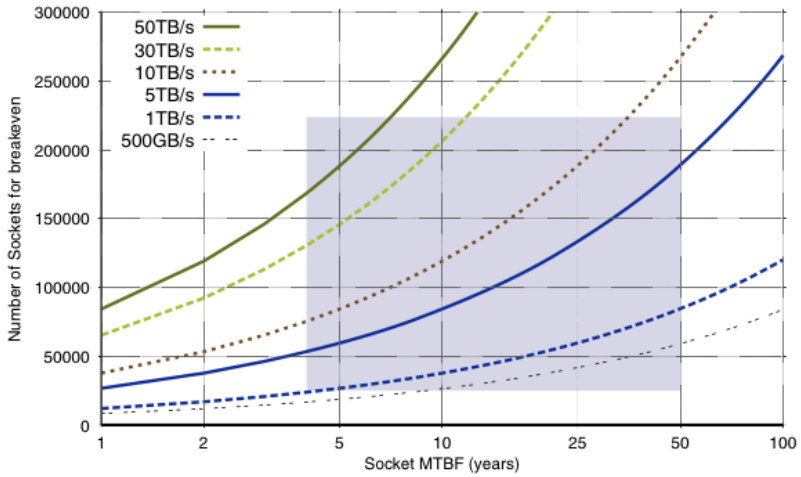


Fig. 4. Shadow replication energy breakeven for different I/O bandwidths. Assumes 16Gb per socket.

Scaling at Different Overhead Power

- The number of available sockets decreases as the percentage of overhead power increases
- As the power overhead increases the potential energy savings also decreases
- Even if the overhead is 100% it will be no worse than traditional replication

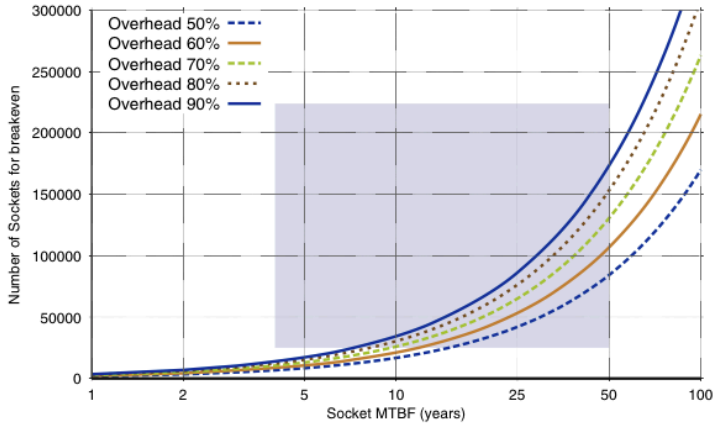
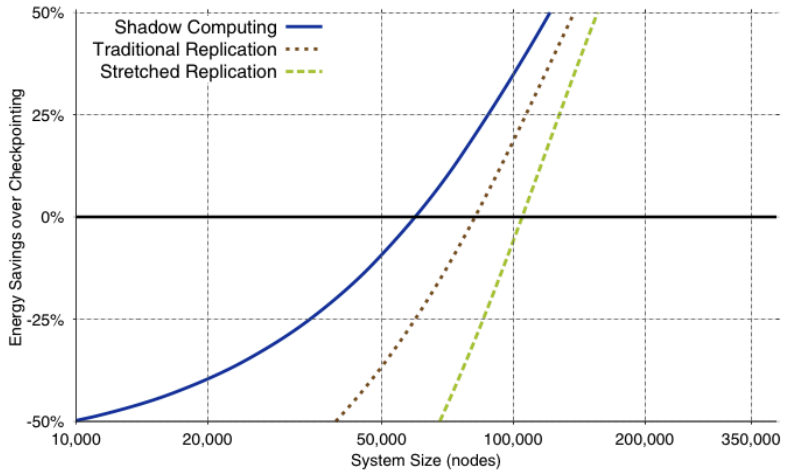


Fig. 5. Shadow replication energy breakeven for various overhead power percentages. Checkpoint time of 15 minutes.

Energy Savings

- Relative energy savings
  - 1TB/s I/O bandwidth
  - 25 Year MTBF
- Shadow replication consistently consumes about 20% less energy than that consumed by pure replication



Implementation and evaluation

- Implemented the replication techniques in MPI and measured the energy
- Experiment environment
  - 104nodes, each with a AMD Llano Fusion, which is a 4-core AMD K10 x86 paired with a 400-core Radeon HD 6550D
- Evaluated applications
  - LAMMPS
    - Molecular dynamics code
  - HPCCG
    - Conjugate gradient solver
  - miniFE
    - Implicit finite element method

Experimental Results

- HPCCG and miniFe show the maximum energy saving
  - Simple applications that are processor bound
- To confirm assumptions about overhead power
  - Lowest possible execution speed
    - CPU consumes 40% of the overall
  - Full power
    - CPU consumes 71% of the overall
  - Estimated amount of overhead power is 67%

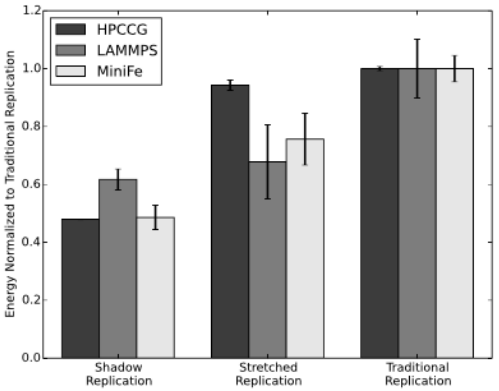


Fig. 7. Experimental results of the energy savings achieved by different replication schemas.

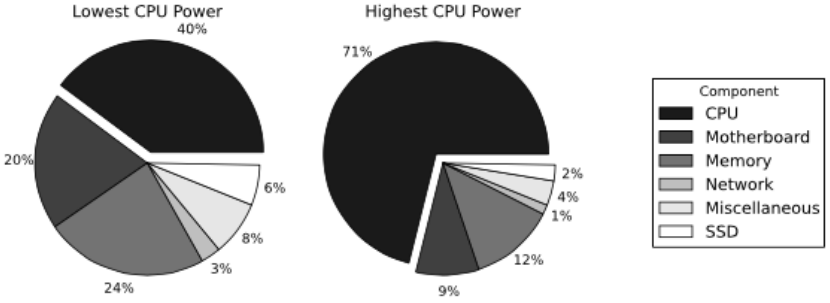


Fig. 8. Component level energy usage for LAMMPS

Conclusion

- Show the benefit of power-aware modifications to replication
  - 40% more time and energy efficient
- This savings makes replication a viable fault tolerance solution through the majority of the exascale-class design space
- Demonstrate at small scale

Discussion

- Viable of shadow replication for extreme levels of I/O bandwidth