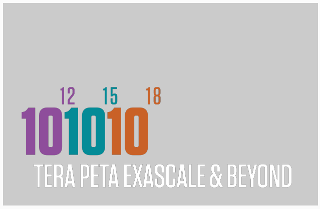
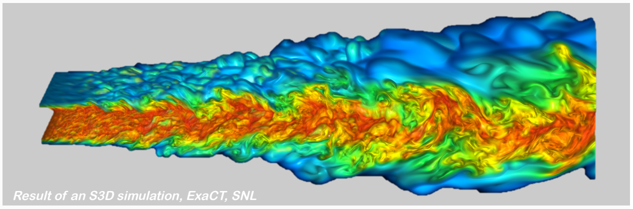
Titan Cray XK7 supercomputer, ORNL




Result of an S3D simulation, ExaCT, SNL

Towards Cross-layer Power and Resilience Management

Marc Gamell[*], Ivan Rodero[*], Keita Teranishi[+], Manish Parashar[*]

[*] *Rutgers Discovery Informatics Institute (RDI^2), Rutgers University*

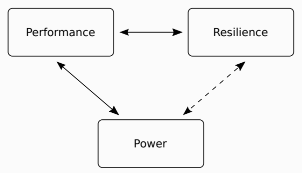
[+] *Scalable Modeling & Analysis, Sandia National Laboratories*

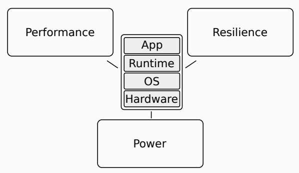
Abstract

As leadership class computers continue to increase in size, challenges like power and resilience become major concerns. Our goal is to model power consumption of several resilience techniques using scientific simulations in order to suggest a holistic cross-layer power and resilience management API that allows balancing tradeoffs and meet power budgets. This contrasts with the current way to handle these requirements, done mainly by the hardware and, in some cases, by the OS or runtime.

Since resilience algorithms are typically power-hungry, we begin by studying their power requirements. This allows us to understand the costs and tradeoffs between power and resilience guarantees of different fault tolerance mechanisms, which can include the evaluation of the used levels of memory hierarchy (SSD/NVRAM/DRAM) and which can be turned off. We then design a cross-layered power and resilience management API so that application programmers can choose the minimum level of resilience required in each code segment. By letting the application specify the requirements, lower levels (runtime/OS) will be able to choose the most convenient fault tolerance solution and configure the hardware appropriately. That will be done by developing policies to control the knobs to balance tradeoffs and meet power budgets.

Motivation





Exascale computing is the result of increasing demands from science and engineering.

Power and Resilience are two major issues towards an exascale machine.

Current approach:

- Focus on Resilience with low Performance impact
- Focus on Power with low Performance impact
- Some recent efforts tackle Resilience and Power

Proposed:

- Tackle Resilience, Power and Performance together

Current Resilience and Power Management approach:

- Each layer try to offer the view of a resilient substrate and automatic power management to the higher-level layer:
    - Processor and Memory: correct bit flips via ECC and control power consumption with proprietary policies
    - OS: automatic DVFS, Checksum-based Data Redundancy (RAID), in-node processing redundancy
    - Runtime: offer resilient abstraction of the machine by using replication, automatic checkpoint restart, or message logging.

These abstractions may not be the best option in all cases, and may result in degraded power and performance.

Proposed:

- Application-centric, cross-layer power and resilience management

System Architecture

Application uses our API to **specify goals or requirements for power, resilience, and performance**.

Based on,

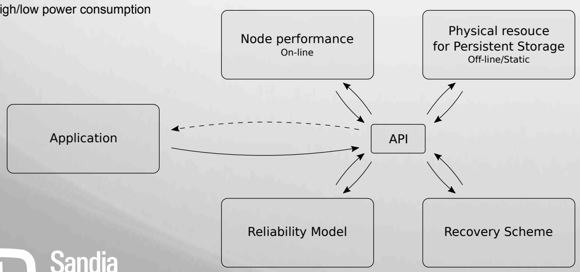
- on-line node measurements and application goals, a certain recovery scheme or reliability model will be chosen.
- off-line, pre-determined formulas, the hints from the application, and system status, the API will decide which storage to use in order to achieve a goal (i.e., optimize power consumption).

Embed analytical models to describe behavior of network and storage and combine it with dynamic measurements.

We will use Sandia's PowerAPI to control the power usage of each node, of the network and storage
We will leverage Fenix and FenixLR to control recovery procedures.

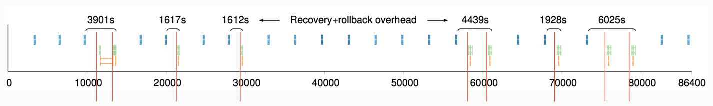
The **application** might **get feedback** from our API so that it can decide how to react to certain events:

- Hard/soft failures
- High/low power consumption



Illustrative Usecases

S3D: Recovering Locally and Reducing Peak Power



S3D production runs on Titan Cray XK7 (125k cores)
9 process/node failures over 24 hours
Failures are promoted to **job failures**
Checkpoint (5.2 MB/core) stored in the PFS

	Average	Total
Checkpoint data	**55 s**	1.72 %
Restarting processes	**470 s**	5.67 %
Loading checkpoint	**44 s**	1.38 %
Rollback overhead	**1654 s**	22.63 %
Total overhead due to fault tolerance		**31.40 %**

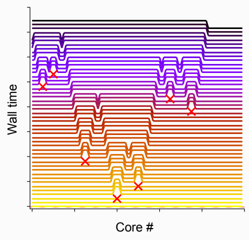
Towards exascale, **O(1)** process/node **failure per minute**

- **Checkpoint frequency** has to be dramatically **increased**
- Current **checkpoint cost, O(1) minute, is unfeasible**

in-memory, application-specific, local, fine-grained, high-frequency checkpointing

- Recovery cost must be reduced
- Node failures cannot be propagated

local online recovery



Local recovery has better power and energy behavior as compared to global recovery as the entire system does not have to roll back and redo computations.

Local recovery also enables **masking multiple failures** (left):

- time to solution appear as if only a single failure occurred

Trading off Failure Masking for Reduced Peak Power

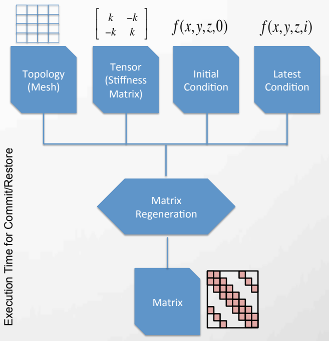
- Depending on the system MTBF or on the usage of collective operations, it might be more beneficial to **throttle the power of survived processes** that have to wait after the failure so that the effect of the failure is immediately spread to the whole system.

Reducing hardware reliability to reduce power consumption

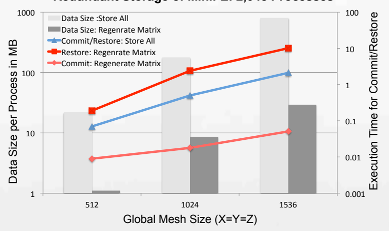
- Power-hungry hardware resilience mechanisms (ECC) can be turned off.
- If the application can easily identify these failures, failure masking can be used to tolerate them with low to none performance overhead.

MiniFE: Reducing Impact of Checkpointing by Recomputing

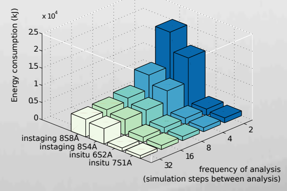
- For some systems and applications, storage can be power hungry. Application has the knowledge to decide.
- Exploit Data Dependencies of Application data
- Recovery through inexpensive local computation
    - Reuse the existing Matrix Assembly code
    - Localized matrix regeneration
- Substantial storage reduction



Redundant Storage of MiniFE: 2,048 Processes



Coupled Codes: In-transit Analysis for Increased Resilience



- Online analysis offers scientists the possibility of observing results from a long-running simulation before it finishes
- Several approaches have been suggested:
    - In-situ offers faster turn-around times but interrupts the execution
    - In-transit does not interrupt the simulation but requires data movement

- If the reliability model requires checkpointing to an staging area, the analysis can occur on the checkpoints themselves.
- Even though this can be more costly in terms of power, the cost of data movement is amortized by checkpointing and analysis.

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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.

RDI^2
Rutgers Discovery Informatics Institute

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY