How Predictable are HPC Applications’ I/O (and why should we care)

A Case for Smarter I/O and Storage Systems

Matthieu Dorier
Argonne National Laboratory
Let’s take the machine’s point of view, shall we?

What can the machine know about the applications?
void output(const char* filename,...) {
    H5Fopen(filename,...);
    ...
    H5Dwrite(...);
    H5Dwrite(...);
    ...
    H5Fclose(...);
}

int main(int argc, char** argv) {
    ...
    while(not done) {
        solve(...);
        output("checkpoint.h5",...);
        output("analysis.h5",...);
    }
}
Modeling a Sequence of Events

Modeling Randomness

“what kind of distribution does it follow?”
“what kind of stochastic process drives it?”

ARIMA models
Markov models
Statistics

Modeling Determinism

“what rules does it follow?”
“what is the logic behind it?”

Compression
Formal Languages
void output(const char* filename,...) {
    H5Fopen(filename,...);
    ...
    H5Dwrite(...);
    H5Dwrite(...);
    ...
    H5Fclose(...);
}

int main(int argc, char** argv) {
    ...
    while(not done) {
        solve(...);
        output("checkpoint.h5",...);
        output("analysis.h5",...);
    }
}
void output(const char* filename,...)
{
    H5Fopen(filename,...);
    ...
    H5Dwrite(...);
    H5Dwrite(...);
    ...
    H5Fclose(...);
}

int main(int argc, char** argv) {
    ...
    while(not done) {
        solve(...);
        output(“checkpoint.h5”,...);
        output(“analysis.h5”,...);
    }
}
Build Rules, Build Models

size = always 512
Build Rules, Build Models
Example: Grammar Models with Omnisc’IO

Learning Phase

Fully Predictable Behavior

237 “colors”

Predicting the size of future accesses

32 different write sizes ranging from 3 B to 2.2 MB

Predicting the date of future accesses

Seasonal data, time series, let’s use an ARIMAL model!

Just grammar and simple averages!
What can the machine do with that?

• Solve I/O Interference:
  • Better scheduling of I/O requests
• Better resource management
  • Burst Buffers, I/O Servers
  • Caching, Read-Ahead
• Simulation of future I/O behaviors
• Better trace compression…
Take away

Find determinism first
Use randomness if necessary
Use the predictability of I/O

Thank you!