Evaluation of simulation models for Batsim

Adrien Faure

2022-02-23
1 Introduction
- HPC: Scheduling and platform management
- Batsim

2 Profiles

3 Ptask validation

4 Ptask invalidation

5 Annex
HPC scheduling

Shared platform → multiple applications

RJMS:
- User requests = jobs
- Scheduling and placement
- Platform management
  - control jobs executions
  - node monitoring
- Complexe piece of code
  - lot of loc
  - system programming
  - distributed programming
How Study scheduling?

- Most cluster uses simple policy (FCFS for instance)
  - with a backfilling mechanism
- Scheduling is not anymore (only) about rectangles
- Applications performance depends:
  - the capacity of the platform
  - their placement on the platform
  - and on the other application running

Can we have simulation with job’s resources consumption?

To be able to study effects such as:
- Network interferences
- Energy optimization / management
- IO management
Batsim[3]: (Successful?) attempt to concentrate effort on RJMS simulation

- Good idea in software engineering
  - on top of Simgrid
    - platform
    - applications
  - separation of concerns
    - isolated scheduling

"Realistic" view

- Platform have defined network / nodes that (simulated) application can use with profiles
Batsim with SimGrid: Application profiles

**Nomenclature**

- **Job** is from the scheduler perspective
- **Application** defines what is running
- **Profile** is a Batsim’s abstraction to model application

**List of jobs**
- Release date: $r_j$
- Number of resources: $q_j$
- Walltime: $wall_j$
- 1 simulation profile: $C_j$

**Running time** → Depends on the profile

**Profils:**

1. `{ "name": "cpu1", "type": "compute", "flops": [1e3, 1e3], "nb_res": 2, }
2. `{ "name": "cpu2", "type": "compute", "flops": [1e3, ...], "nb_res": 5, }`
1 Introduction

2 Profiles
   - List of profiles
   - Choosing a model

3 Ptask validation

4 Ptask invalidation

5 Annex
Profile Delay

Delay

- Fixed execution time
- Static input
- The most used [4, 5]

Model inputs

- Number of seconds

Properties

- Fast to compute
- No interference effects
- Realistic when:
  - Homogeneous platform
  - There is no interference
Profile Time Independent Trace (TiT)

**Time Independent Trace (TiT)**

- Fine grained model
- Replay SMPI traces
- Running time depends on:
  - amount of work (cpu/network)
  - platform capacity
  - placement
  - other jobs

**Model inputs**

- Trace TiT: List of SMPI calls to simulate

**Properties**

- Only for MPI jobs
- + Inter / Intra interference effects
- + Has been validated [2]
- – Slow
- Realistic when:
  - Static jobs MPI
  - The performances doesn't depends on the data
Profile Ptask

Parallel task model
- Coarse grain model
- Homogeneous progression
- Running time depends on:
  - Amount of work
  - Platform capacity
  - Placement
  - Other jobs

Model inputs
- Uses two resources
  - Array of computation amounts
  - Communication matrix

Description on how it is computed in my defense slides (slide 21).

Properties
- + Intra / inter effects
- + Fast
- - Not evaluated before
Model performances

- Replay at: https://gitlab.inria.fr/adfaure/ptask_tit_eval
How to choose a model?

A model resources consumption of job?

<table>
<thead>
<tr>
<th>Profile Delay</th>
<th>Profile ptask</th>
<th>Profile TiT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very fast</td>
<td>Fast (enough)</td>
<td>Slow</td>
</tr>
<tr>
<td>No interference</td>
<td>Interference effects</td>
<td>Interference effects</td>
</tr>
<tr>
<td>Poor in information</td>
<td>Resource consumption</td>
<td>Dedicated for MPI</td>
</tr>
<tr>
<td></td>
<td>Computations and communications</td>
<td>application</td>
</tr>
</tbody>
</table>
How to choose a model?

A model resources consumption of job?

Profile Delay
- Very fast
- No interference
- Poor in information

Profile ptask
- Fast (enough)
- Interference effects
- **Resource consumption**
  - Computations and communications

Profile TiT
- Slow
- Interference effects
- Dedicated for MPI application

We need to validate the Ptask model
1 Introduction

2 Profiles

3 Ptask validation
   - Methodology and platform setup
   - Results and conclusion

4 Ptask invalidation

5 Annex
Experimental methodology

- Network interference
- Comparison between simulation and reality

<table>
<thead>
<tr>
<th>Reality</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel application (MPI)</td>
<td>Execute one ptask</td>
</tr>
<tr>
<td>Matrix product</td>
<td>Corresponding to our real application</td>
</tr>
<tr>
<td>Controled interference generation</td>
<td>With the simulated interference</td>
</tr>
</tbody>
</table>

. Source code and data: https://gitlab.inria.fr/batsim/ptask-eval/-/tree/tcp_connections
Platform setup

Platform network

Network without contention

Creating on contention point
- Two group of machines
- Split in two subnetwork
- Inter group communications uses the routing nodes

Plateformes Grid’5000 :
- Grisou et Paravance
- Homogeneous - *Dell poweredge R640*
- Not the same switch
Platform setup

Reconfigured network

Network without contention

Creating on contention point
- Two group of machines
- Split in two subnetwork
- Inter group communications uses the routing nodes

Plateformes Grid’5000 :
- Grisou et Paravance
- Homogeneous - Dell poweredge R640
- Not the same switch
Real application
- Homogenous application
  - Cycles: communications + compute
- Distributed matrix
- 8 nodes per group (16 core/node)
- Différent parameters
  - Matrix splitting
  - Sync/Async broadcasts

Interferences
- Tcpkali = open source software
- Generates interferences
- 60 s periods
- % interferences: (0%, 25%, .., 100%)
  - 25%: 15 s interferences / 45 s idle
  - 25%: 15 s interférences / 45 s idle
Application behavior and configuration

- homogeneous network
- 0% interferences = faster
- 100% interferences = slower

Application configuration:
- 50 sub-matrix
- Async Broadcasts
ptask generation: resources consumption

Simulation needs:
- computation and communication amounts

Computation array
- Computed from the algorithm
- Depends on the matrix size:
  \[
  \left( \frac{\text{matrix\_size}}{\sqrt{\text{nb\_processes}}} \right)^3
  \]

Communication matrix
- Depends on the broadcast algorithm
  - end-to-end \(\neq\) peer-to-peer
  - Simgrid and SMPI \(\rightarrow\) Tracing peer-to-peer
Results: Reality versus simulation

Running times don’t match → The behavior match
Ptask conclusion (of my thesis)

1st step towards validation
- Accurate without interference
- Interferences look good (spoiler they are not)

Évolution nécessaire
Network degradation needs calibration
Two platforms : 2 results

How to calibrate the model ?
- Application profiling ?
- Add parameters in the model ?

Next steps
- Why platforms have different behavior?
- How to calibrate the interferences?
1 Introduction

2 Profiles

3 Ptask validation

4 Ptask invalidation
   - New setup
   - Results and conclusion

5 Annex
**Improved version of the experiment**

**H1: The sharing algorithm in paravance is different**
- Grisou has a behavior that looks normal
- Maybe paravance has a different sharing algorithm

**First improvement**
Interferences with more complexe communication pattern

**H2: How to calibrate the interferences ?**
- The number of tcp connection is important
  - overlooked in my thesis...
- The interferences used 100 TCP connections

**Second improvement**
Number of tcp connection variations
Different patterns of interferences

Random generation of similar Graph

- Number of connections wanted
- Probability to connect two nodes
- Probability set to 0 → nodes are not connected

Interferences now use several nodes
Other minor modifications

**To spare time and energy**

- Only configuration with 100% of interferences
- Homogeneous progress → so we stop the application early
  - Based on the number of cycle it does
Results: Number of connections

<table>
<thead>
<tr>
<th>Number of TCP connections</th>
<th>Mpi runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grisou</td>
<td></td>
</tr>
<tr>
<td>Paravance</td>
<td></td>
</tr>
</tbody>
</table>

Number of host pairs:
- 0
- 1
- 2
- 3
- 4
- 5
Results: Number of pairs

- **Grisou**
  - Number of connections:
    - 0
    - 1
    - 2
    - 4
    - 8
    - 12
    - 16
    - 24
    - 32

- **Paravance**
  - Number of connections:
    - 0
    - 1
    - 2
    - 4
    - 8
    - 12
    - 16
    - 24
    - 32

- The graphs show the relationship between the number of host pairs and MPI runtime for different numbers of connections.

26 / 34
Conclusion on difference between Paravance and Grisou

- **Grisou**
  - Sensitive to the number of TCP connections
  - Not impacted by their location

- **Paravance**
  - Sensitive to the number of different pair communicating

**H1: Grisou vs Paravance**

- Different switch with different sharing
- Paravance has a different sharing behavior
  - **Not yet supported in SimGrid**
- Related to how TCP handles contentions
  - filtering on hosts and omits the port number?

We use only Grisou for the following
Comparison between:
- Reality
- Ptask
- SMPI

Variating the number of connections
**Conclusion on model calibration**

**Model too sensitive to the number of connections:**
- **Inner connections are not taken into account** → wrong sharing
- and it’s not possible to fix it..
  - it supposes that it has one bottleneck
    (which is not the case)

**H2: Invalidation of the model**
- **Maybe useful in some cases?**
- with Arnaud and Bruno, we are working on a new model
  - based on bottleneck max fairness sharing algorithm [1]
  - see Arnaud talk on the subject
Application progress

**Internal monitoring**
- 0% interferences
- 25% interferences
- 50% interferences
- 75% interferences
- 100% interferences

**Homogeneous progress**
2 cadences distinctes:
- Fast with interferences
- Slow with interferences
Results: Total bandwidth

![Graph showing total bandwidth vs. number of connections for Grisou and Paravance. The graph includes scatter plots for different numbers of connections per pair, indicated by different markers.]

- **Grisou**
- **Paravance**

**Number of connections per pair**
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 11
- 15
- 20
Results: Total bandwidth

<table>
<thead>
<tr>
<th>Number of connections</th>
<th>Total bandwidth (Mo/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of connections:
- 1
- 2
- 4
- 8
- 12
- 16
- 24
- 32

