rmf_tool – A library to Compute (Refined) Mean Field Approximation(s)

Sebastian Allmeier

> *Univ. Grenoble Alpes, Inria, Grenoble, France - sebastian.allmeier@inria.fr (mailto:sebastian.allmeier@inria.fr)*

Nicolas Gast

> *Univ. Grenoble Alpes, Inria, Grenoble, France - nicolas.gast@inria.fr (mailto:nicolas.gast@inria.fr)*

How can we efficiently *analyze, understand* and *optimize* large scale stochastic systems?

Example: Load balancing systems

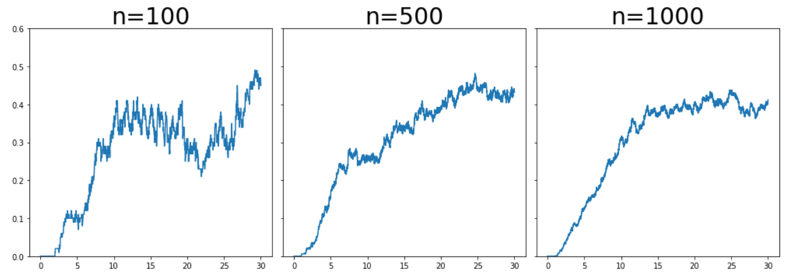
→ compare policies & evaluate performance

Mean Field Approximation technique can be help analyzing

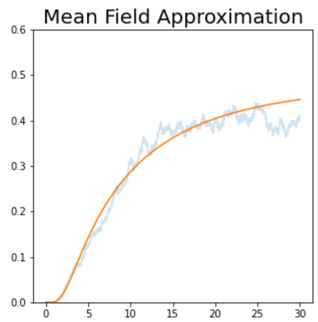
rmf_tool (refined mean field tool) aims to facilitate the useage

Some Intuition

System with:
→ n interacting objects
    i.e. servers
→ finite states for each object
    i.e. queue length
Problem: ⇒ exploding state space (n^S possible states)



$\downarrow \quad n \rightarrow \infty$

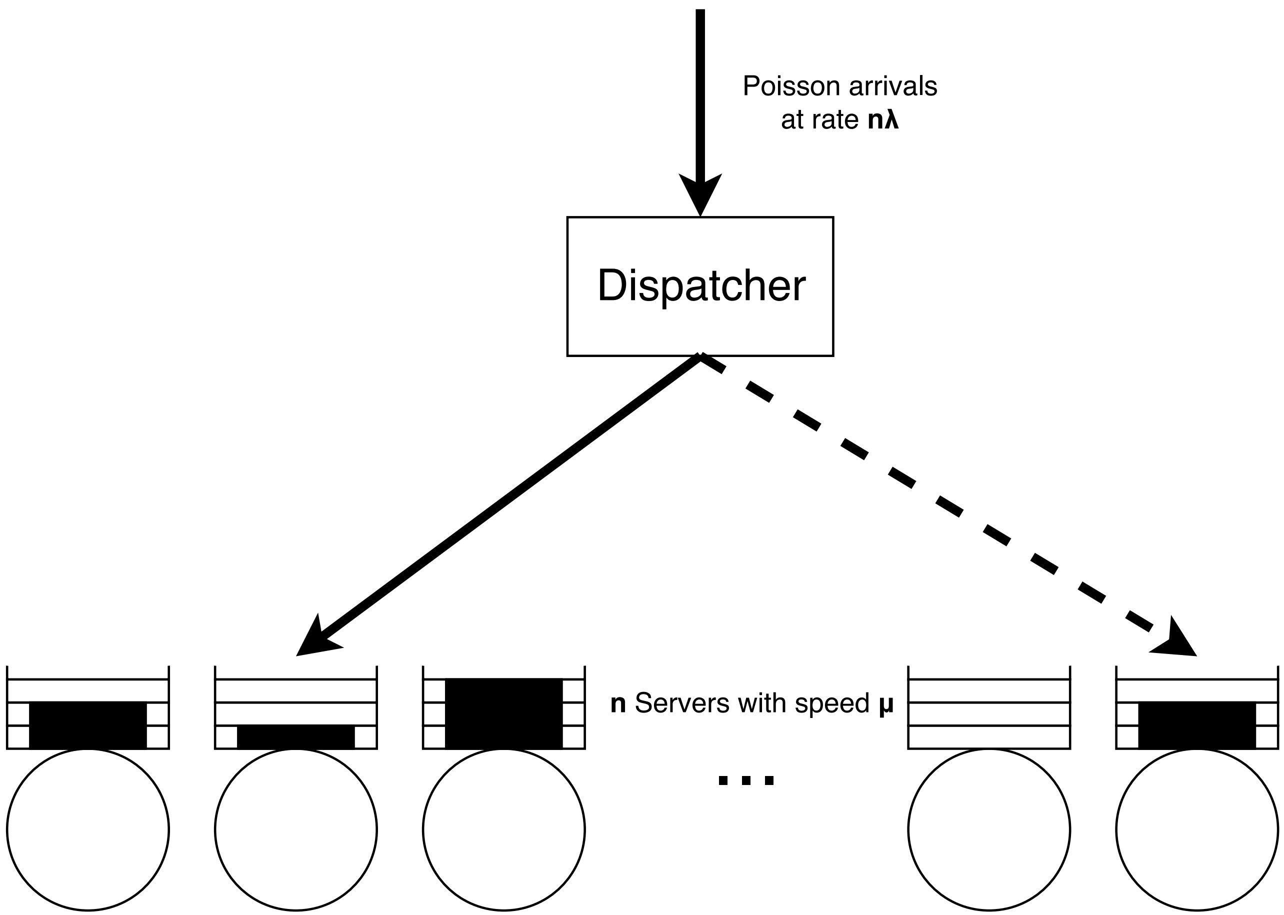


Mean field can simplify

ROADMAP

1. Load Balancing Example - JSQ(2)
2. Tool Features

Example: Power-of-two-choices load balancing



Model representation

$X^{(n)} = \left(X_0^{(n)}, \dots, X_K^{(n)}\right)$ with $X_i^{(n)}$ the fraction of servers having at least i jobs in the queue.

servers	server speed	arrival rate	buffer size
n	$\mu = 1$	$\lambda = 0.9$	K=9

State changes to servers with queue length i

Arrival transition	Arrival Rate
$X \mapsto X + \frac{1}{n} e_i$	$n\lambda(X_{i-1}^2 - X_i^2)$

Removal Transition	Removal Rate
$X \mapsto X - \frac{1}{n} e_i$	$n\mu(X_i - X_{i+1})$

Model Implementation

In [2]:
```
import rmf_tool.src.rmf_tool as rmf

# This code creates an object that represents a "density dependent population process"
ddpp = rmf.DDPP()

# Set parameters
mu, _lambda, K = 1.0, 0.9, 9
```

In [4]:
```
# Add transitions using mathematical formulation:
for i in range(K):
    if i >= 1:
        ddpp.add_transition(e(i), eval('lambda x: _lambda*(x[{}]*x[{}] - x[{}]*x[{}] )'.format(i-1,i-1,i,i) )) # arrivals
    if i < K-1 and i > 0:
        ddpp.add_transition(-e(i), eval('lambda x: mu*(x[{}] - x[{}])'.format(i,i+1))) # removals
```

Calculating Mean Field Approximation and Simulation

Mean Field Approximation is the average variation:

$$\dot{x}_i(t) = \underbrace{\lambda(x_{i-1}^2 - x_i^2)}_{\text{arrival}} - \underbrace{\mu(x_i - x_{i+1})}_{\text{removal}}, \qquad x(0) = X(0)$$

In [15]:
```
# Set nitial state
ddpp.set_initial_state(e(0))

# Calculate mean field
T, x_transient = ddpp.ode(time=30)
```

In [16]:
```
# Simulate a trajectory for N=50
T_n50, X_n50 = ddpp.simulate(N=50,  time=30)
# and for N=1000
T_n1k, X_n1k = ddpp.simulate(N=1000,time=30)
```

Tool features:

Mean Field Approximation (transient + steady state results) and Simulation for

Homogeneous Population Models

→ systems with similar object behavior

Heterogeneous Population Models [Allmeier and Gast, 2021] (https://arxiv.org/abs/2111.01594)

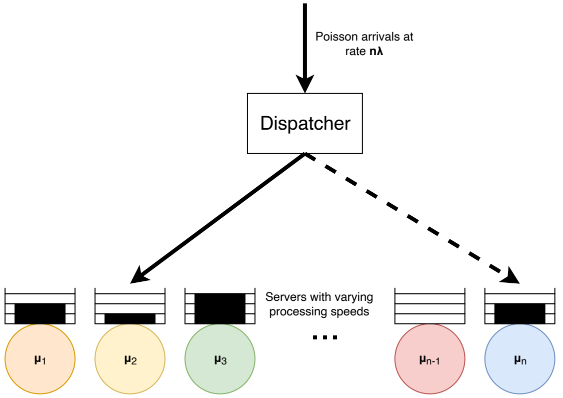
→ systems with varying object behavior

Refined Mean Field Approximation [Gast et al., 2019] (https://www.sciencedirect.com/science/article/abs/pii/S0166531618302633?via%3Dihub)

→ increased accuracy
→ especially important for $n \approx 10 - 100$

Examples:

Power-of-two-choice model with varying server speeds



More examples: (Heterogeneous) epidemic model (SIR/SIS) , caching policies, SSD garbage collection, load balancing models, etc.

Thank you

Sebastian Allmeier

sebastian.allmeier@inria.fr (mailto:sebastian.allmeier@inria.fr)

References

rmf_tool – A library to Compute (Refined) Mean Field Approximation(s) by Allmeier and Gast

→ https://github.com/ngast/rmf_tool (https://github.com/ngast/rmf_tool)

Mean Field and Refined Mean Field Approximations for Heterogeneous Systems: It Works! by Allmeier and Gast

Size Expansions of Mean Field Approximation: Transient and Steady-State Analysis by Gast, Bortolussi, Tribastone

Expected Values Estimated via Mean Field Approximation are O(1/N)-accurate by Gast.