Randomized Load Balancing

Asymptotic optimality of power-of-d-choices with memory

Jonatha ANSELMI

Joint work with Francois DUFOUR, INRIA

Data Centers and Cloud Computing

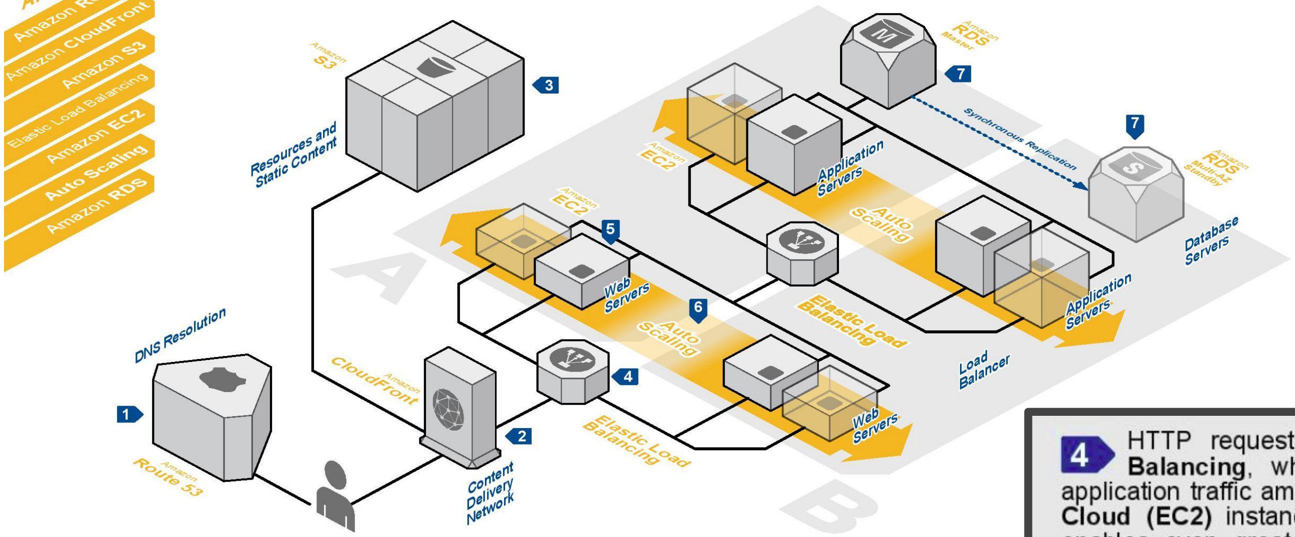
Resource allocation problems

- ➢ Align IT resources with service demand
- ➢ optimizing resource usage
- ➢ reducing costs

Load balancing in data-storage and computing systems

- Replication
- Minimizing delays

Computing systems



WEB APPLICATION HOSTING

Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale-out and scale-down infrastructure to match IT costs in real time as customer traffic fluctuates.

System Overview

1 The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.

2 Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

3 Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.

4 HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.

5 Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.

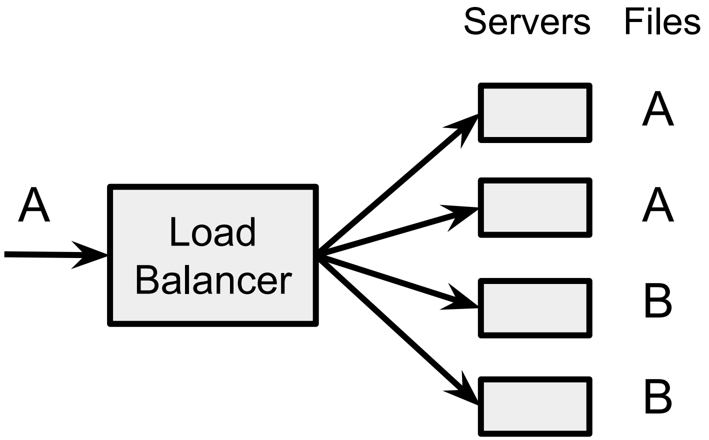
6 We... **Aut**... your capa... With Aut... **Amazon** ... during demand spikes to maintain performance and decreases automatically during demand to minimize costs.

7 To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service** (Amazon RDS).

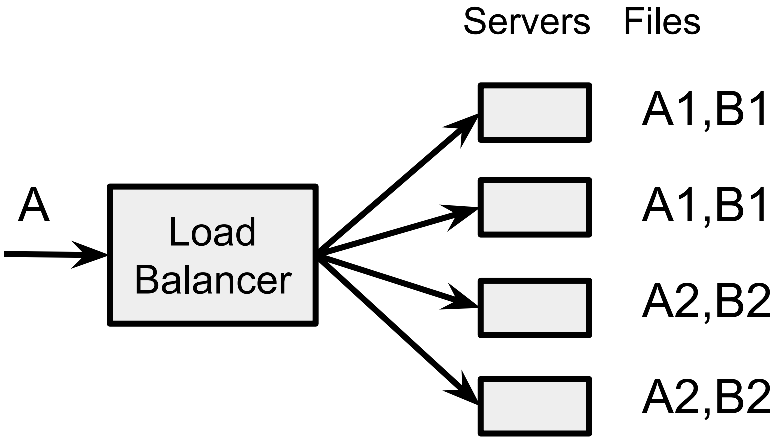
4 HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.

Data storage networks

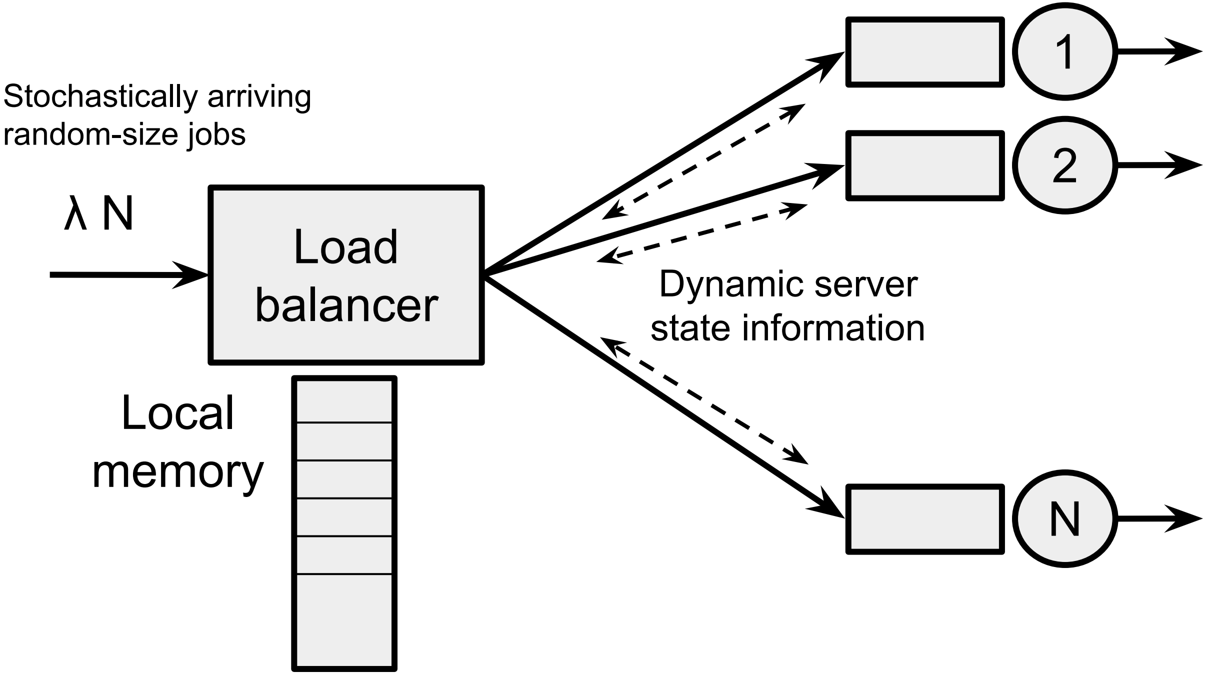
Replication



Coding



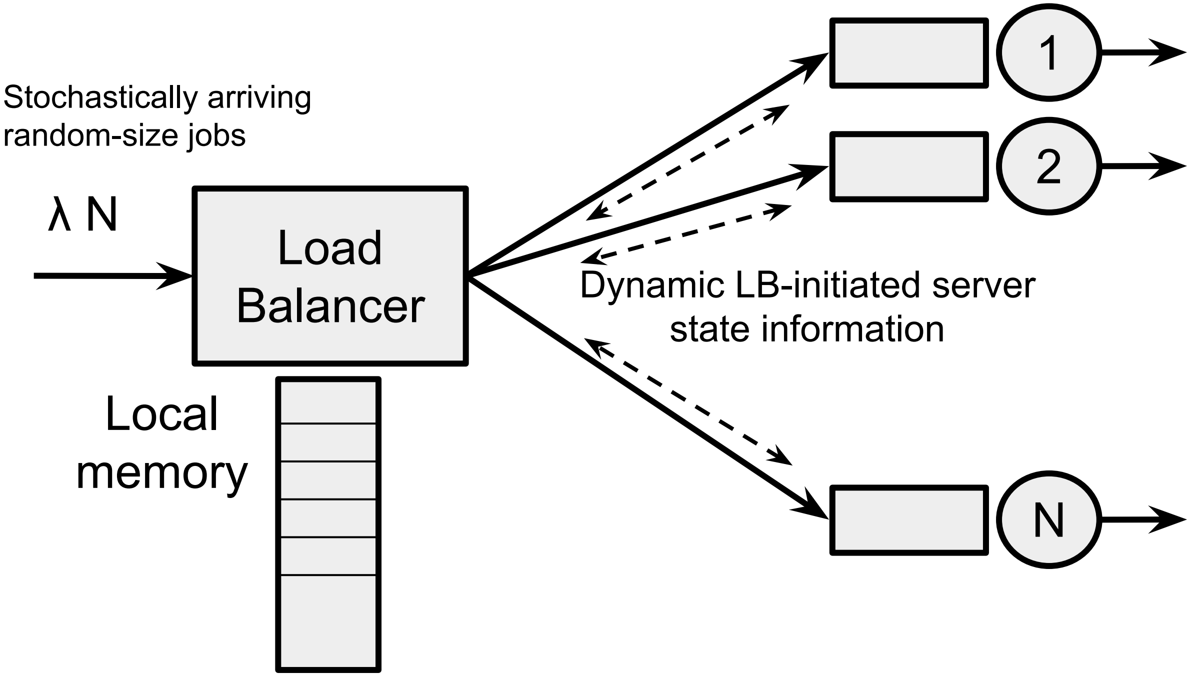
Architecture of the distributed system



Dispatching Algorithms

Random, RND
Round-Robin, RR
Join the shortest queue, SQ(N)
Power-of-d-choice, SQ(d)
Redundancy-d, Red-d
Join the idle queue, JIQ
Idle-one first, JIQ++
d-choices with memory, SQ(d,b)
. . .

➢ **Which one the best?**


Stochastically arriving
random-size jobs
λ N
Load
Balancer
Local
memory
Dynamic LB-initiated server
state information
1
2
N

What we know in a nutshell

Heavy traffic optimality

If the workload or queue-lengths process is minimized over all time in the diffusion limit as $\lambda \uparrow 1$ and N is fixed.

Fluid (or mean field) optimality

If the steady-state probability of an arriving job experiencing waiting converges to zero when $N \uparrow \infty$ and $\lambda < 1$ is fixed.

➢ **Orthogonal standpoints**

QL	HT optimality	Fluid optimality	Overhead
RND	No, $\frac{1}{1-\lambda}$	No, $\frac{1}{1-\lambda}$	0
RR	No, $\frac{1}{2}\frac{1}{1-\lambda}$	No, ≥ $\frac{1}{2}\frac{1}{1-\lambda}$	0
SQ(N)	Yes	Yes	~N^2
SQ(d)	Yes	No, $\simeq \frac{\log \frac{1}{1-\lambda}}{\log d}$	~N
Red-d	-	No	~N
JIQ	No, $\simeq$ RND	Yes	~N
JIQ++	No, $\simeq$ RND	Yes	~N
SQ(d,b)	Yes	No	~N

Our approach: SQ*(d,N)*

Algorithm 1 Power-of-d-choices with memory and N servers.

```
procedure SQ(d, N)
    Memory[i] = 0, ∀i = 1, . . . , N;
    for each job arrival do
        for i = 1, . . . , d do
            rnd_server = random(1,. . . , N);
            Memory[rnd_server] = get_state(rnd_server);
        end for
        selected_server = random(arg min_{i∈{1,...,N}} Memory[i]);
        send_job_to(selected_server);
        Memory[selected_server]++;
    end for
end procedure
```

Our main results

SQ(*d*,*N*) is fluid optimal if and only if $\lambda < 1 - \frac{1}{d}$

If λ∈ [0,1), the longest queue is $\left\lceil \frac{-\ln(1-\lambda)}{\ln(\lambda d + 1)} \right\rceil$

Heavy traffic optimality

Consequence of the HT optimality of SQ(*d*).

- **SQ(*d*,*N*) unique fluid- and heavy-traffic optimal algorithm employing a linear overhead**

The remainder of the talk

➔ **A stochastic and a deterministic model for the dynamics of SQ(d,N)**
Continuous time Markov chain, differential equation.

➔ **Connection between both models**
Kurtz-like result

➔ **Fixed points, stability and fluid optimality**
Lyapunov-like result

➔ **Conclusions and future research**

Stochastic model

- ➔ **Arrivals:** Poisson process with rate *λ N*
- ➔ **Job sizes:** Poisson process with rate *1*
- ➔ **Server speeds:** constant *c=1*

$Q_k^N(t)$: number of jobs in queue *k* at time *t*

$M_k^N(t)$: the last observation collected from server *k* at time *t*

Define $X^N(t) = (X_{i,j}^N(t), 0 \leq i \leq j)$ where

$$X_{i,j}^N(t) = \frac{1}{N}\sum_{k=1}^{N} \mathbf{1}_{\{Q_k^N(t)=i, M_k^N(t)=j\}} \longrightarrow$$

Proportion of *(i,j)*-servers
ie servers with *i* jobs and observation *j*

Then, $X^N(t)$ is a continuous-time Markov chain (with involved non-Lipschitz transitions and rates!)

Sample path construction on (0,0)

- ➔ $(V_n^p)_{n=1}^{\infty}$ for *p*=1,...,*d*, to select the servers to sample at each arrival (Line 5 of Alg. 1)
- ➔ $(W_n)_{n=1}^{\infty}$ to randomize among the servers having the lowest observations (Line 8 of Alg. 1)

All random variables are independent and $U([0,1])$. Then,

$$X_{0,0}^N(t) = X_{0,0}^N(0) + \frac{1}{N}\sum_{n=1}^{\mathcal{N}_\lambda(Nt)} \overbrace{\sum_{p=1}^{d} \mathbb{I}_{(X_{0,0}^N(t_n^{N,\lambda-}),X_{0,\cdot}^N(t_n^{N,\lambda-})]}(V_n^p)}^{\text{at most } d \text{ new idle servers are sampled}}$$
$$+ \frac{1}{N}\sum_{n=1}^{\mathcal{N}_\lambda(Nt)} \underbrace{\mathbf{1}_{\{X_{0,0}^N(t_n^{N,\lambda-})=0\}} \prod_{p=1}^{d} \mathbb{I}_{(X_{0,\cdot}^N(t_n^{N,\lambda}-),1]}(V_n^p) - 1}_{\text{A job is always assigned to a (0,0)-server if it exists}}$$

Deterministic model

Let $\mathcal{S} = \left\{ (x_{i,j} \in \mathbb{R}_+ : 0 \le i \le j \le I) : \sum_{i=0}^{I} \sum_{j=i}^{I} x_{i,j} = 1 \right\}, \quad x_{i,\cdot} = \sum_{j=i}^{I} x_{i,j}, \quad x_{\cdot,j} = \sum_{i=0}^{j} x_{i,j}$

Definition.

A function $x(t) : \mathbb{R}_+ \to \mathcal{S}$ is said to be a *fluid model* (or fluid solution) if it is absolutely continuous and $\frac{dx_{i,j}(t)}{dt} = b_{i,j}(x(t))$ almost everywhere for all *i* and *j* where *b(x)* satisfies:

$$b_{0,0}(x) = \underbrace{\lambda d(x_{0,\cdot} - x_{0,0})}_{\text{New idle servers are discovered}} - \overbrace{\lambda}^{\text{All jobs are sent to (0,0)-servers...}} + \underbrace{\mathcal{R}_0(x)}_{\text{... unless no (0,0)-server exists}}$$

Note that $d(x_{0,\cdot} - x_{0,0}) = \sum_{p=1}^{d} p \binom{d}{p} (x_{0,\cdot} - x_{0,0})^p (1 - x_{0,\cdot} + x_{0,0})^{d-p}$ and $\mathcal{R}_0(x) = 0 \vee \lambda \left(1 - d x_{0,\cdot}\right) \mathbf{1}_{\{x_{0,0}=0\}}$

is interpreted as the rate in which $X_{0,0}^N(t)$ tends to remain on zero on [*t*,*t*+ε] when *N*→∞ and ε↓0.

Deterministic model (continued)

$$b_{1,1}(x) = -\underbrace{x_{1,1}}_{=A} + \underbrace{\lambda d(x_{1,\cdot} - x_{1,1})}_{=B} + \underbrace{\lambda - \mathcal{R}_0(x)}_{=C} - \underbrace{\mathcal{R}_0(x)\frac{x_{1,1}}{x_{\cdot,1}}\mathbf{1}_{\{x_{\cdot,1}>0\}}}_{=D} - \underbrace{\mathcal{G}_1(x)}_{=E}$$

Interpretation.

- A: Departures from (1,1)-servers. They occur with rate $NX^N_{1,1}(t)$ and decrease $X^N_{1,1}(t)$ by *1/N*
- B: Discovery of new (1,·)-servers, as for $b_{0,0}(x)$
- C: Job assignments to (0,0)-servers, see $b_{0,0}(x)$
- D: Job assignments to (1,1)-servers when a (strictly) positive mass of (·,1)-servers exists
- E: Job assignments to (1,1)-servers when a **null** (!) mass of (·,1)-servers exists

We let $\mathcal{G}_1(x) = \lambda d\,\mathbf{1}_{\{x_{0,0}+x_{0,1}+x_{1,1}=0,\, 2x_{0,\cdot}+x_{0,\cdot}\leq\frac{1}{d}\}}(x_{0,\cdot} + x_{1,\cdot})$

Deterministic model (continued)

The remaining coordinates of *b(x)* admit similar interpretations

$$b_{i,j}(x) = x_{i+1,j} - \mathbf{1}_{\{i>0\}} x_{i,j} - \lambda d x_{i,j} - \mathcal{R}_{j-1}(x)\frac{x_{i,j}}{x_{\cdot,j}}\,\mathbf{1}_{\{x_{\cdot,j}>0\}} + \mathbf{1}_{\{i>0\}}\mathcal{R}_{j-2}(x)\frac{x_{i-1,j-1}}{x_{\cdot,j-1}}\,\mathbf{1}_{\{x_{\cdot,j-1}>0\}}$$

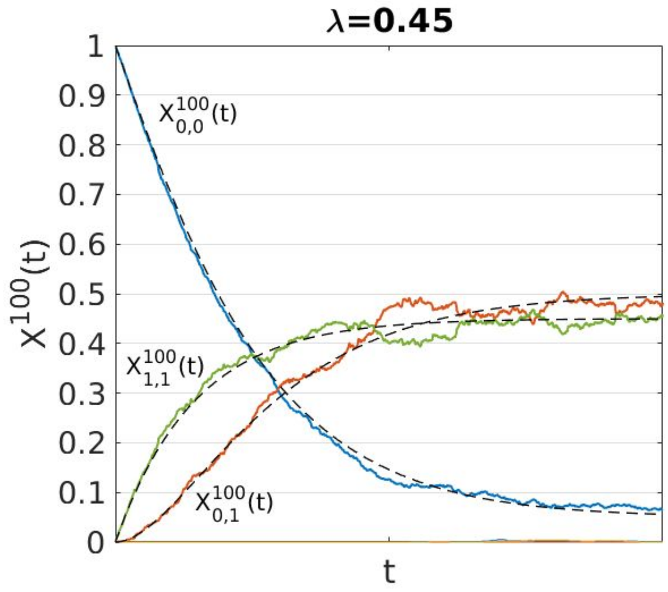
$$\begin{aligned} b_{i,i}(x) &= -x_{i,i} + \lambda d(x_{i,\cdot} - x_{i,i}) - \mathcal{R}_{i-1}(x)\frac{x_{i,i}}{x_{\cdot,i}}\,\mathbf{1}_{\{x_{\cdot,i}>0\}} + \mathcal{R}_{i-2}(x)\frac{x_{i-1,i-1}}{x_{\cdot,i-1}}\,\mathbf{1}_{\{x_{\cdot,i-1}>0\}} \\ &\quad + \mathcal{G}_{i-1}(x) - \mathcal{G}_i(x), \end{aligned}$$

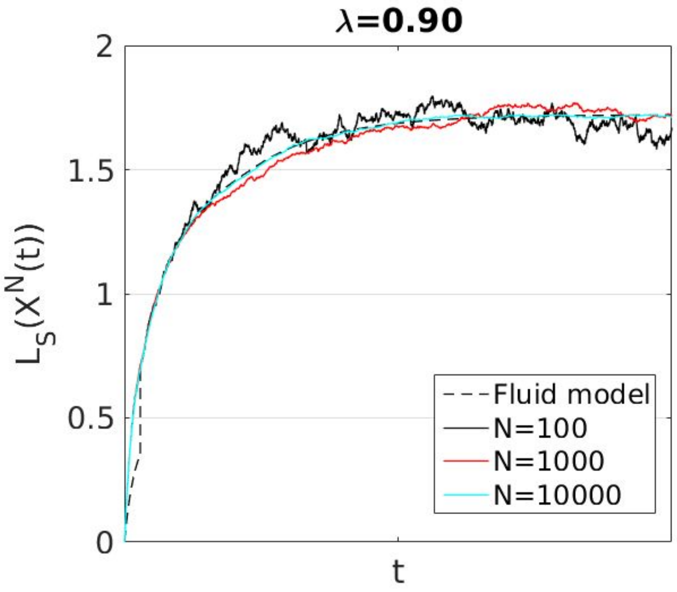
where

$$\mathcal{G}_j(x) = \lambda d\,\mathbf{1}_{\{\sum_{i=0}^{j} x_{\cdot,i}=0,\, d\sum_{i=0}^{j}(j+1-i)x_{i,\cdot}\le 1\}} \sum_{i=0}^{j} x_{i,\cdot}$$

$$\mathcal{R}_j(x) = 0 \vee \lambda\left(1 - d\sum_{i=0}^{j}(j+1-i)x_{i,\cdot}\right)\mathbf{1}_{\{\sum_{i=0}^{j} x_{\cdot,i}=0\}}$$

Connection between both models





Plot on the right:

$$\mathcal{L}_S(x) = \sum_i i x_{i,\cdot}$$

vs

$$\mathcal{L}_S(X^N(t)) = \sum_i i\, X^N_{i,\cdot}(t)$$

Theorem.
Assume that $X^N(0) \to x^0 \in \mathcal{S}$ almost surely. With probability one, any limit point of the stochastic process $(X^N(t))_{t\in[0,T]}$ satisfies the conditions that define a fluid solution.

Proof strategy

3 steps:

➔ Coupled construction of $(X^N(t))_{t\in[0,T]}$ for all *N≥d* on a single probability space in terms of the fundamental processes $(V_n^p)_{n=1}^\infty$, for *p=1,...,d*, $(W_n)_{n=1}^\infty$ and $(U_n)_{n=1}^\infty$

➔ Tightness of sample paths: limit trajectories exist and are Lipschitz continuous, with probability one [Tsitsiklis and Xu 2012, Bramson 1998]

➔ Any limit trajectory satisfies the differentiability condition of a fluid solution (main difficulty). Deep analysis of $(X^N(t))_{t\in[0,T]}$ on convergent subsequences.

Fixed points

Definition. A fluid solution is a *fixed point* if $b(x(t)) = 0$, for all *t*.

Let $j^\star = \left\lfloor \frac{-\ln(1-\lambda))}{\ln(\lambda d + 1))} \right\rfloor$

Theorem. There exists a unique fixed point, say $x^\star$.

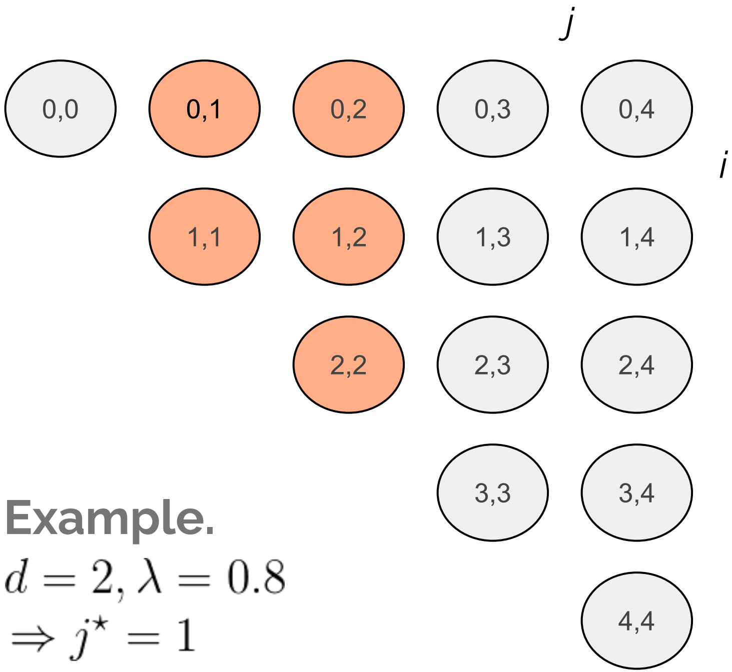
It is such that $x^\star_{\cdot,j^\star} + x^\star_{\cdot,j^\star+1} = 1$ and

$$\lambda d\, x^\star_{0,j^\star} = (1+\lambda d)(1-\lambda) - \frac{1}{(1+\lambda d)^{j^\star}}$$

$$x^\star_{0,j^\star} + x^\star_{0,j^\star+1} = 1 - \lambda$$

Queue lengths (and thus delays) are uniformly bounded!

In contrast with SQ(*d*)



Example.
$d = 2, \lambda = 0.8$
$\Rightarrow j^\star = 1$

Fixed points

Definition. A fluid solution is a *fixed point* if $b(x(t)) = 0$, for all *t*.

Let $j^\star = \left\lfloor \dfrac{-\ln(1-\lambda))}{\ln(\lambda d + 1))} \right\rfloor$

Theorem. There exists a unique fixed point, say $x^\star$.

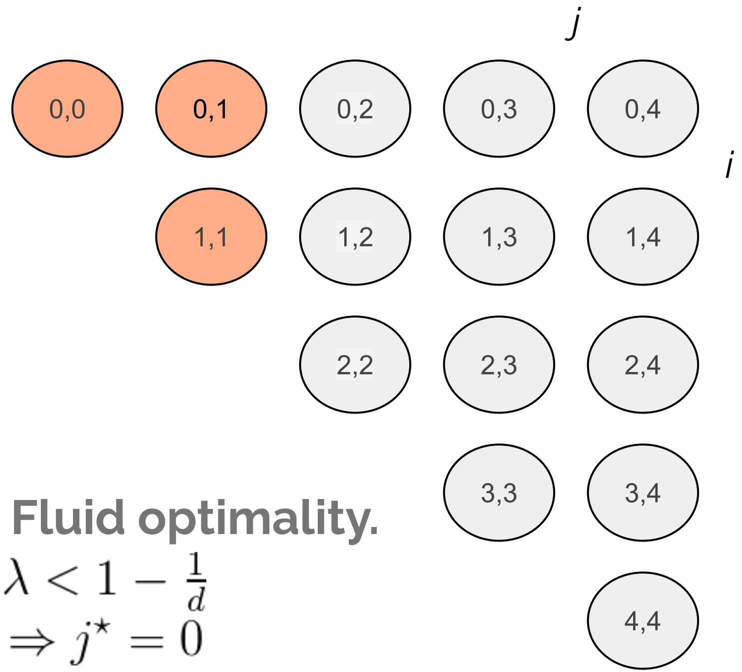
It is such that $x^\star_{\cdot,j^\star} + x^\star_{\cdot,j^\star+1} = 1$ and

$$\lambda d \, x^\star_{0,j^\star} = (1+\lambda d)(1-\lambda) - \frac{1}{(1+\lambda d)^{j^\star}}$$

$$x^\star_{0,j^\star} + x^\star_{0,j^\star+1} = 1 - \lambda$$

If $\lambda < 1 - \frac{1}{d}$, jobs always assigned to (0,0)-servers!

Fluid optimality of SQ(*d*,*N*)



Fluid optimality.

$\lambda < 1 - \frac{1}{d}$

$\Rightarrow j^\star = 0$

Mean queue lengths at the fixed point

Let $\mathcal{L}_S(x) = \sum_i i x_{i,\cdot}$ $\qquad \mathcal{L}_M(x) = \sum_j j x_{\cdot,j}$

$\mathcal{L}_S(X^N(t))$: the number of jobs scaled by *N* in the system at time *t*.
$\mathcal{L}_M(X^N(t))$: the number of jobs scaled by *N* the load balancer *believes* are in the system at time *t*.

Corollary.

$$\left\lfloor \frac{-\ln(1-\lambda))}{\ln(\lambda d + 1)} \right\rfloor - \frac{1}{d} \leq \mathcal{L}_S(x^\star) \leq \left\lceil \frac{-\ln(1-\lambda))}{\ln(\lambda d + 1)} \right\rceil - \frac{1}{d}$$

$$\mathcal{L}_M(x^\star) = \mathcal{L}_S(x^\star) + \frac{1}{d}$$

Stability

Theorem. Let $x(t)$ be a fluid solution. Then, $\lim_{t\to\infty} \|x(t) - x^\star\| = 0$. Furthermore, if $\lambda < 1 - \frac{1}{d}$ convergence occurs exponentially fast.

Proof (schema).

$V(t) = \sum_i |z_i(x(t)) - z_i(x^\star)|$ is a Lyapunov function, where $z_i = \sum_{i' \geq i} x_{i'},$.

Whenever *t* is a point of differentiability of a fluid solution $x(t)$ and $x_{0,0}(t) = 0$,

$$\dot{\mathcal{L}}_S(x(t)) = x_{0,\cdot}(t) - 1 + \lambda \leq \frac{1}{d} - 1 + \lambda$$

Thus, there must be a point where $x_{0,0}(t)$ increases. When it does, $x(t)$ "couples" with a linear ODE system.

Some practical improvements

- ➔ Server selections without replacements
- ➔ Do not allow to sample (0,0)-servers
- ➔ Upon a job arrival, if i is both the least loaded of the d sampled servers and the least observation contained in the memory immediately before the sampling, then send the job to one of the $(\cdot, i)$-server known to the load balancer immediately before the sampling.

. . . though the fluid limit does not change!

Merci