Modelling and control of BigData services

Application to MapReduce performance and dependability

Mihály Berekméri

PhD Defense

Université Grenoble Alpes

Supervised by:
Sara Bouchenak
Bogdan Robu

INSA Lyon · PERSYVAL-Lab · gipsa-lab

Context

Cloud computing

- Cloud services are everywhere

Dropbox

YouTube

LinkedIn

twitter

facebook

Gmail

Context

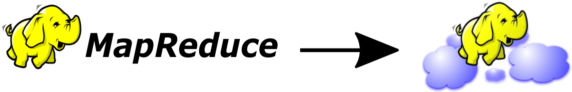
Cloud computing

- Cloud services are everywhere

Dropbox · YouTube · LinkedIn · twitter · facebook · Gmail

- More and more BigData cloud services

amazon web services · Amazon Elastic Map Reduce (EMR) · HDInsight Microsoft · Google Cloud Platform



Challenges in the cloud

Performance issues

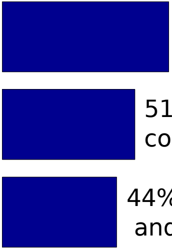
- Recent survey from Compuware, 468 CIOs and senior IT professionals

Challenges in the cloud

Performance issues

- Recent survey from Compuware, 468 CIOs and senior IT professionals

Biggest concerns about managing cloud services?



64% Poor end user experience due to performance bottlenecks

51% Impact of poor performance on consumer loyalty

44% Loss of revenue due to availability and performance, or troubleshooting

Challenges in the cloud

Dependability issues

- September 2015, overloaded with requests

Challenges in the cloud

Dependability issues


TechWeek europe

TECH TIMES

PERSONAL TECH | BIZ TECH | FUTURE TECH | SCIENCE | LIFE | T-LOUNGE

TAG Amazon Web Services , Tinder , Netflix , Reddit , AWS Services

Amazon Web Services Suffers Crash, Takes Down Netflix, Reddit, Tinder And Other Huge Parts Of The Internet

By Romellaine Arsenio, Tech Times | September 23, 11:14 AM

Like | Follow | Share(19) | Tweet(?) | Reddit | 5 Comments | SUBSCRIBE

The Amazon Web Services crashed Neflix, Reddit, Tinder

A monstrous outage from Amazon Web Services crashed down Netflix, Reddit, Tinder and other major websites, sending netizens in fury for missing movies, hook-ups and other fun online activities.

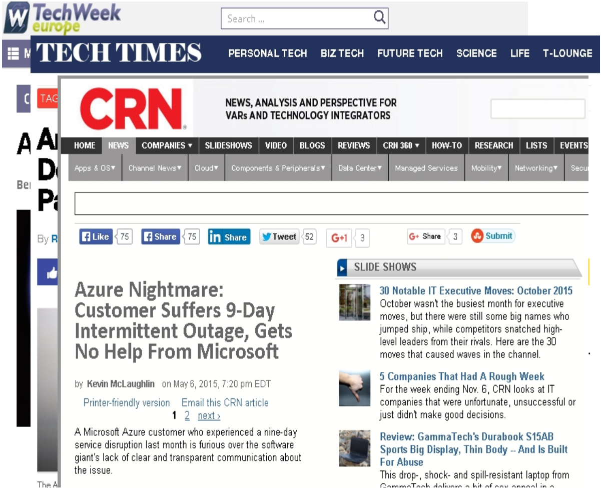
AWS powers web and mobile applications, and provides data processing and warehousing, storage and archiving to websites all over the world.

Instead of building their own data infrastructures, countless important online marketers place their trust on the cloud services offered by Amazon to lessen their long-term investments in the

- September 2015, overloaded with requests

Challenges in the cloud

Dependability issues



- September 2015, overloaded with requests

Problem statement

- Highly dynamic workloads
- Multiple criteria at the same time
  - performance, dependability, cost
- No fully automatic control solutions

Contents

Contents

QoS and SLA

- Quality of Service (QoS)
  - The ability to meet different requirements
  - e.g. performance, availability, cost
- Service Level Objectives(SLO)
  - Specific, measurable service characteristics
  - e.g. Maximum response time 5s
- Service Level Agreement (SLA)
  - Contract $\rightarrow$ formalises QoS
  - Parties involved (roles), penalties
  - Multiple SLOs
  - e.g. Maximum response time 5s; Minimum availability 98%;

MapReduce

- One of the most used BigData systems
- Simple to use, scalable and fault-tolerant
- Wide range of applications:
  - log analysis, data mining, web search engines, scientific computing, business intelligence

MapReduce usage

Yahoo	40.000 computers
LinkedIn	$120 \cdot 10^9$ relationships/day

MapReduce

- One of the most used BigData systems
- Simple to use, scalable and fault-tolerant
- Wide range of applications:

MapReduce

- One of the most used BigData systems
- Simple to use, scalable and fault-tolerant
- Wide range of applications:
  - log analysis, data mining, web search engines, scientific, computing, business intelligence

MapReduce usage

Google	100.000 jobs/day
Yahoo	40.000 computers
Facebook	100 petabytes (10^{15})
LinkedIn	$120 \cdot 10^{9}$ relationships/day

MapReduce

- One of the most used BigData systems
- Simple to use, scalable and fault-tolerant
- Wide range of applications:
  - log analysis, data mining, web search engines, scientific, computing, business intelligence

MapReduce usage

Google	100.000 jobs/day
Yahoo	40.000 computers
Facebook	100 petabytes (10^{15})
LinkedIn	$120 \cdot 10^9$ relationships/day

MapReduce

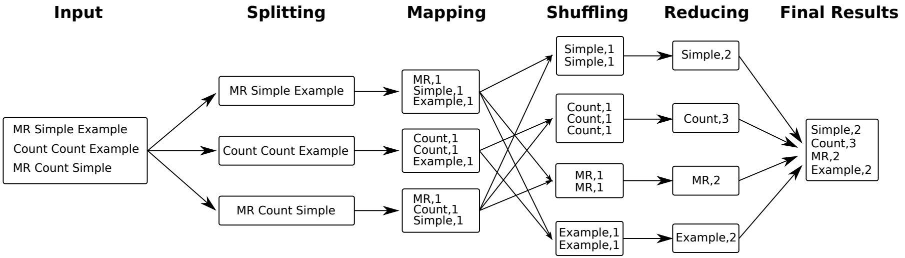
- One of the most used BigData systems
- Simple to use, scalable and fault-tolerant
- Wide range of applications:
  - log analysis, data mining, web search engines, scientific, computing, business intelligence

MapReduce usage

Google	100.000 jobs/day
Yahoo	40.000 computers
Facebook	100 petabytes (10^{15})
LinkedIn	$120 \cdot 10^9$ relationships/day

MapReduce Simple Example

- Task: count the occurrence of words in a file
- Only two functions to implement
  - Mapper
  - Reducer



MapReduce quality-of-service

- Service performance:
  - response time - average time (y_{rt}) to process a client request

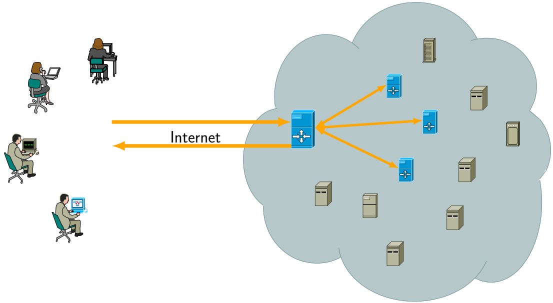
$$y_{rt}[T] = avg(y_{rt_1}, y_{rt_2}, ..., y_{rt_N})$$

- Service dependability:
  - availability - system accessibility to users per unit of time

$$y_{av}\left[\frac{\%}{T}\right] = \frac{N_{SuccessfulJobs}}{N_{SuccessfulJobs} + N_{RejectedJobs}} * 100$$

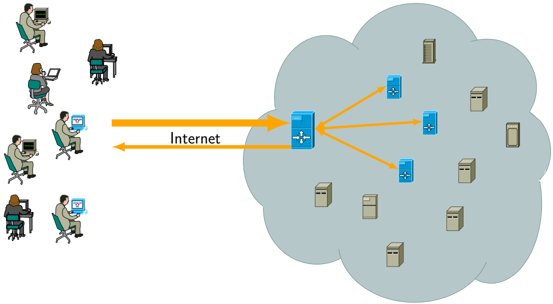
MapReduce parameters

- Many parameters (Hadoop $\rightarrow$ 170)
  - most not configurable on-line
- Examples of on-line controllable parameters:
  - N - number of processing nodes
  - MC- maximum accepted client requests



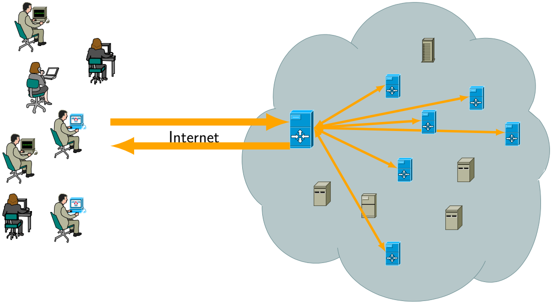
MapReduce parameters

- Many parameters (Hadoop $\rightarrow$ 170)
  - most not configurable on-line
- Examples of on-line controllable parameters:
  - N - number of processing nodes
  - MC- maximum accepted client requests



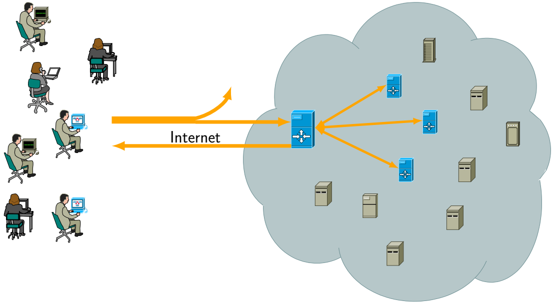
MapReduce parameters

- Many parameters (Hadoop $\rightarrow$ 170)
  - most not configurable on-line
- Examples of on-line controllable parameters:
  - N - number of processing nodes
  - MC- maximum accepted client requests



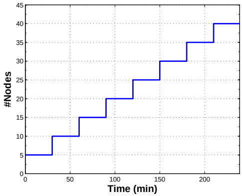
MapReduce parameters

- Many parameters (Hadoop $\rightarrow$ 170)
  - most not configurable on-line
- Examples of on-line controllable parameters:
  - N - number of processing nodes
  - MC- maximum accepted client requests

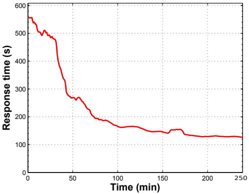


Motivating example

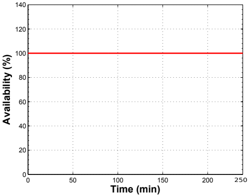
Impact of nodes variation on performance and dependability



Nodes



Response time

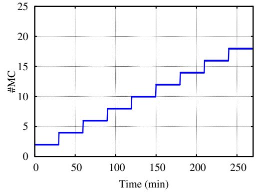


Availability

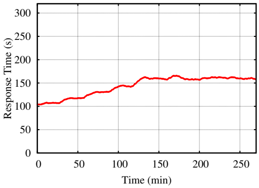
#MC=10,#Clients=10

Motivating example

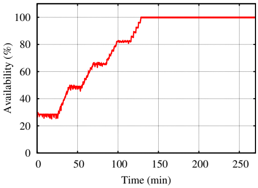
Impact of max clients variation on performance and dependability



MaxClients



Response time



Availablity

#Clients=10,#Nodes=20

Contents

Control theoretical approach

Develop and apply control strategies for MapReduce

- But why control theory?
  - Unified mathematical framework
  - Automated tools for modelling and control synthesis
  - Proven algorithms
- Challenges:
  - No physics behind algorithms, applications
  - Difficult to use classical techniques

Control theoretical approach

Develop and apply control strategies for MapReduce

- But why control theory?
  - Unified mathematical framework
  - Automated tools for modelling and control synthesis
  - Proven algorithms
- Challenges:
  - No physics behind algorithms, applications
  - Difficult to use classical techniques

Related work

Improving MapReduce

- Bottleneck management [Sharma, CLOUD'12]
- Data placement [ADAPT, ICDCS'12]
- Fault recovery [Ruiz, ICDE'11]
- Cost-based optimization [Herodotou, VLDB'11]

✗ No performance or dependability guarantees

✗ Heuristic, best-effort

Related work

- Modelling MapReduce performance
  - Fine grained, analytical models
    [Herodotos, VLDB'11]
    [Vianna, IJPP'13]
  - Course grained, regression models
    [Verma, Middleware'11]
    [Xu, IP&DPS'12]

✗ Single-output models

Related work

- Controlling MapReduce performance

[SteamEngine, HiPC'11]
[ARIA, ICAC'11]
[Jockey, ACM'12]

✗ Don't ensure multiple objectives

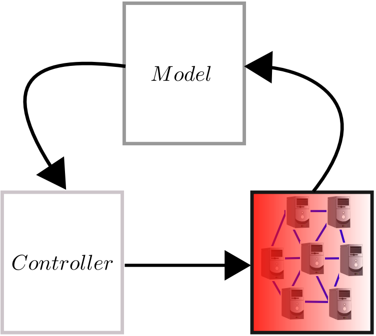
Related work

Open issues

- Multi-input, Multi-output (MIMO) models
- Controllers capable of ensuring multiple SLOs at the same time
- Handling the trade-off between contradictory SLOs

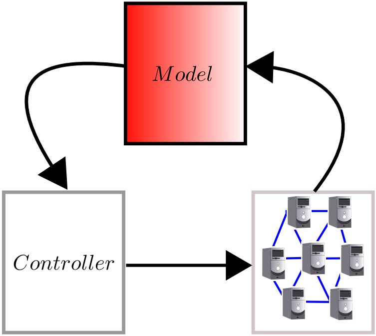
Objectives

- Experimental MapReduce environment
- MIMO MapReduce model
- Controllers → ensure a multi-objective SLA
  - Performance **and** dependability SLOs, meanwhile minimising cost
  - Faced with dynamic client variations



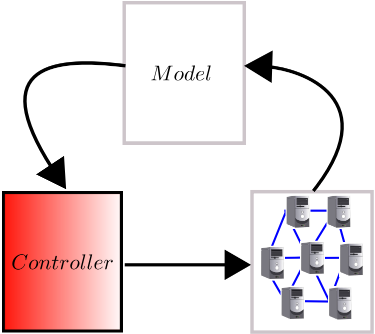
Objectives

- Experimental MapReduce environment
- MIMO MapReduce model
- Controllers → ensure a multi-objective SLA
  - Performance **and** dependability SLOs, meanwhile minimising cost
  - Faced with dynamic client variations



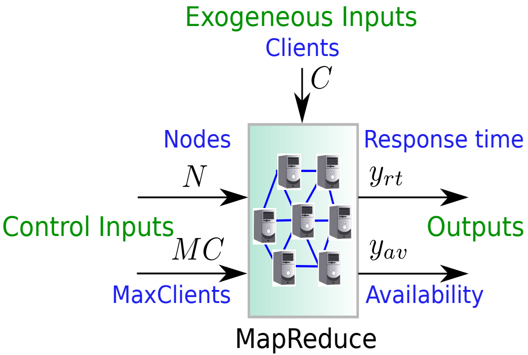
Objectives

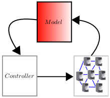
- Experimental MapReduce environment
- MIMO MapReduce model
- Controllers → ensure a multi-objective SLA
  - Performance **and** dependability SLOs, meanwhile minimising cost
  - Faced with dynamic client variations



Contents

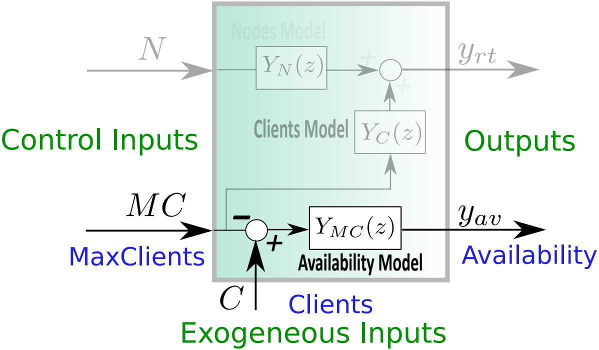
Model structure





- N number of nodes in the cluster
- MC maximum number of concurrent clients
- C number of concurrent clients

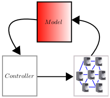
MapReduce dependability model



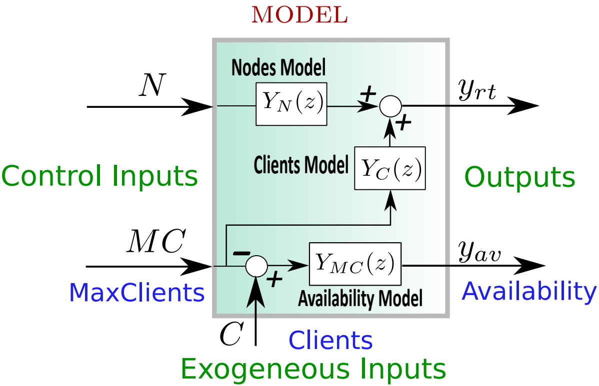
- Mathematical relation between number of clients, max clients and availability

$$y_{av} = Y_{MC}(z) \cdot (C - MC)$$

where: $MC \leq C$

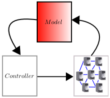


MapReduce performance and dependability



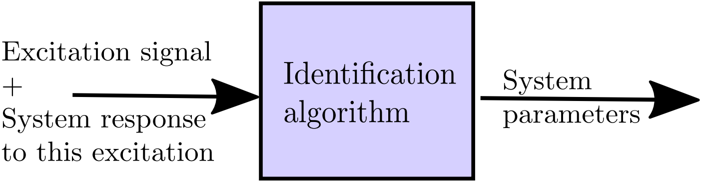
$$y_{rt} = Y_C(z) \cdot MC + Y_N(z) \cdot N$$
$$y_{av} = Y_{MC}(z) \cdot (C - MC)$$

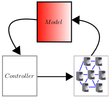
where: $MC \leq C$



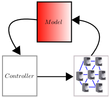
How to determine the model structure and parameters?

- Automated off-line identification and on-line adaptation tools





1. Prediction error method → off-line identification
2. Recursive least square → on-line adaptation

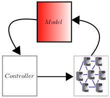


Towards identifying model parameters

- Transfer function formulation

$$y_{rt} = Y_C(z) \cdot MC + Y_N(z) \cdot N$$
$$y_{av} = Y_{MC}(z) \cdot (C - MC)$$

- State space formulation - matricial, compact



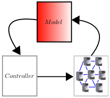
Towards identifying model parameters

- Transfer function formulation

$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + Y_N(z) \cdot N$$

$$y_{av} = Y_{MC}(z) \cdot (C - MC)$$

- State space formulation - matricial, compact



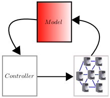
Towards identifying model parameters

- Transfer function formulation

$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z + a_n)} \cdot N$$

$$y_{av} = Y_{MC}(z) \cdot (C - MC)$$

- State space formulation - matricial, compact



Towards identifying model parameters

- Transfer function formulation

$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z + a_n)} \cdot N$$

$$y_{av} = \frac{b_{mc}}{(z + a_{mc})} \cdot (C - MC)$$

- State space formulation - matricial, compact

Towards identifying model parameters

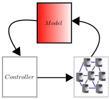
- Transfer function formulation

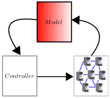
$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z + a_n)} \cdot N$$

$$y_{av} = \frac{b_{mc}}{(z + a_{mc})} \cdot (C - MC)$$

- State space formulation - matricial, compact

$$x_{k+1} = A_d \cdot x_k + B_d \cdot u_k$$
$$y_k = C_d \cdot x_k$$





Towards identifying model parameters

- Transfer function formulation

$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z + a_n)} \cdot N$$

$$y_{av} = \frac{b_{mc}}{(z + a_{mc})} \cdot (C - MC)$$

- State space formulation - matricial, compact

$$x_{k+1} = A_d \cdot x_k + B_d \cdot u_k$$
$$y_k = C_d \cdot x_k$$

Towards identifying model parameters

- Transfer function formulation

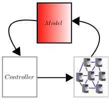
$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z + a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z + a_n)} \cdot N$$

$$y_{av} = \frac{b_{mc}}{(z + a_{mc})} \cdot (C - MC)$$

- State space formulation - matricial, compact

$$x_{k+1} = A_d \cdot x_k + B_d \cdot u_k$$

$$y_k = C_d \cdot x_k$$



Towards identifying model parameters

- Transfer function formulation

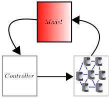
$$y_{rt} = z^{-\tau_{rtc}} \frac{b_c}{(z+a_c)} \cdot MC + z^{-\tau_{rtn}} \frac{b_n}{(z+a_n)} \cdot N$$

$$y_{av} = \frac{b_{mc}}{(z+a_{mc})} \cdot (C - MC)$$

- State space formulation - matricial, compact

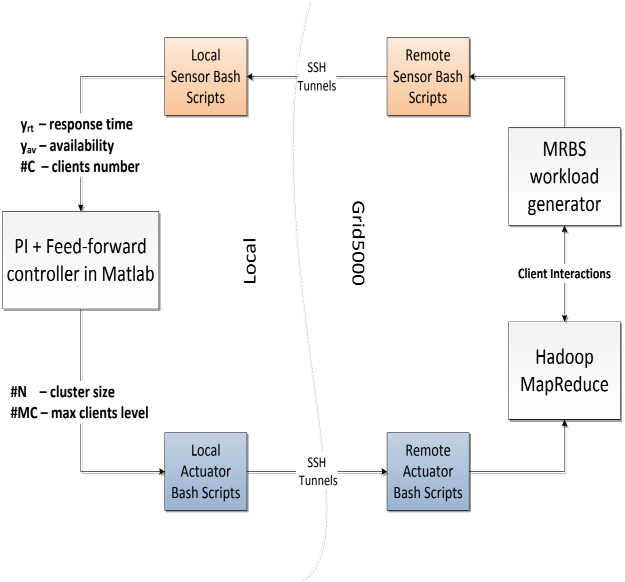
$$x_{k+1} = A_d \cdot x_k + B_d \cdot u_k$$

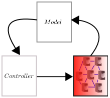
$$y_k = C_d \cdot x_k$$

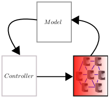


Contents

Experimental setup





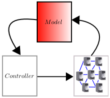


Experimental setup

- MRBS
  - Performance and dependability benchmark suite
  - Realistic multi-user workloads
  - Data intensive business intelligence benchmark
- Clients request: one or more jobs($\sim$10GB)
- French nationwide research cluster: Grid5000

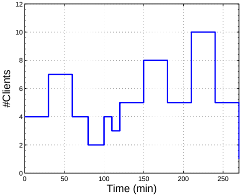
Node configuration

Quad-core Intel 2.53GHz CPU	15GB RAM
Infiniband 20G network	298GB disk

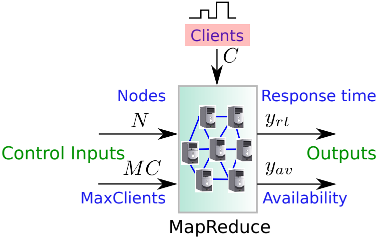


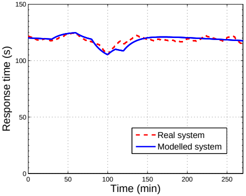
Model validation

Accuracy in capturing clients variation

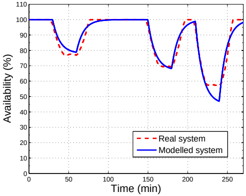


Clients



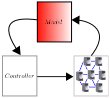


Response time



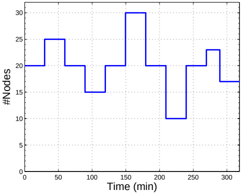
Availability

#MC=5, #Nodes=20

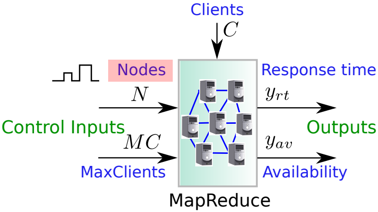


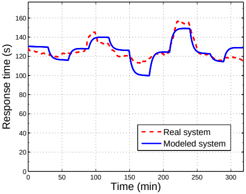
Model validation

Accuracy in capturing nodes variation

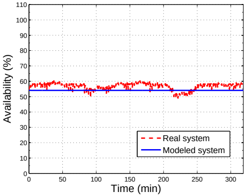


Nodes





Response time

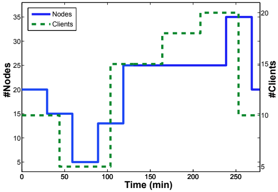


Availability

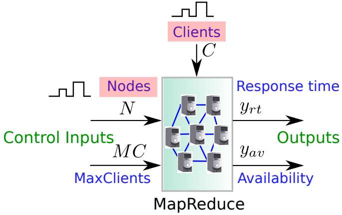
#MC=5, #Clients=10

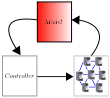
On-line parameter adaptation validation

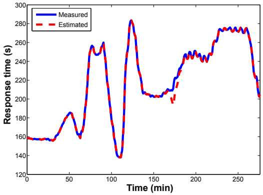
Acurracy in capturing nodes and clients variation



Nodes, Clients







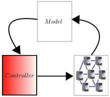
Response time

#MC=30

Contents

Controllers

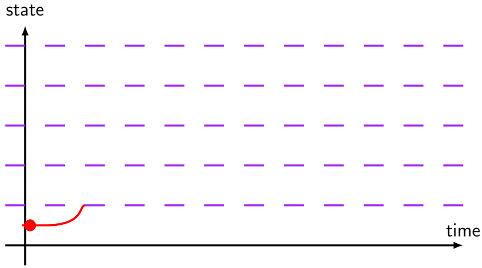
- 3 control types
  - *MR-Perf*
    - Time-based feedback-feedforward controller
    - Ensures performance
  - *MR-Perf-Cost*
    - Event-based feedback-feedforward controller
    - Ensures performance, minimal costs
  - *MR-Ctrl*
    - Constrained optimal control
    - Ensures performance and dependability, while minimising costs

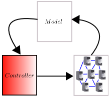


MR-Perf-Cost

Event-based control

- SLA:
  - Keep response time under a given threshold
  - Minimal costs (minimal node count and variation)

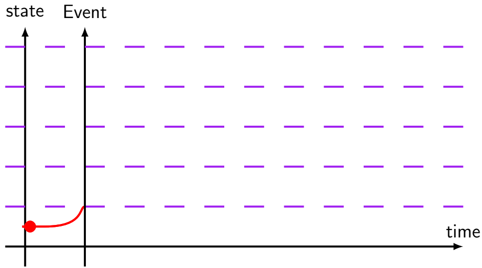


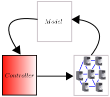


MR-Perf-Cost

Event-based control

- SLA:
  - Keep response time under a given threshold
  - Minimal costs (minimal node count and variation)

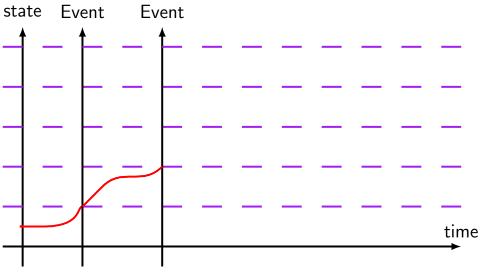


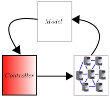


MR-Perf-Cost

Event-based control

- SLA:
  - Keep response time under a given threshold
  - Minimal costs (minimal node count and variation)

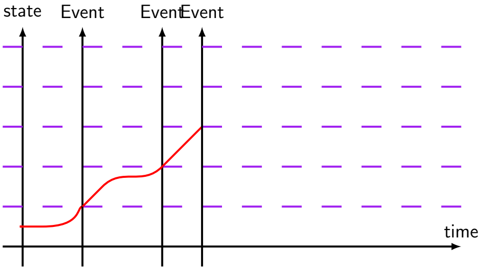


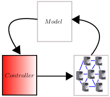


MR-Perf-Cost

Event-based control

- SLA:
  - Keep response time under a given threshold
  - Minimal costs (minimal node count and variation)

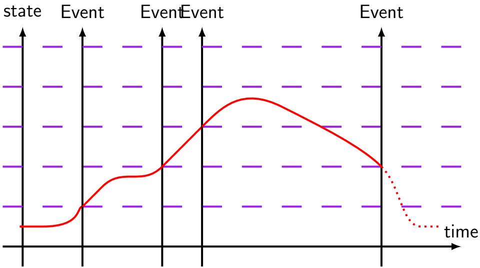




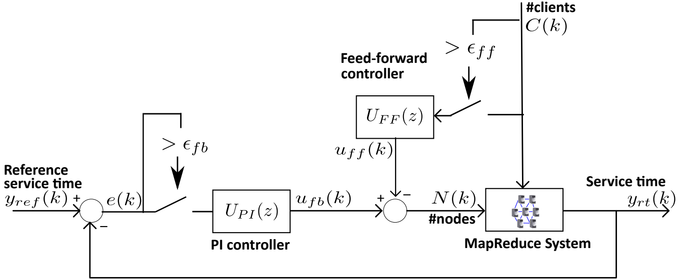
MR-Perf-Cost

Event-based control

- SLA:
  - Keep response time under a given threshold
  - Minimal costs (minimal node count and variation)



MR-Perf-Cost

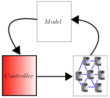


- Event-based PI control

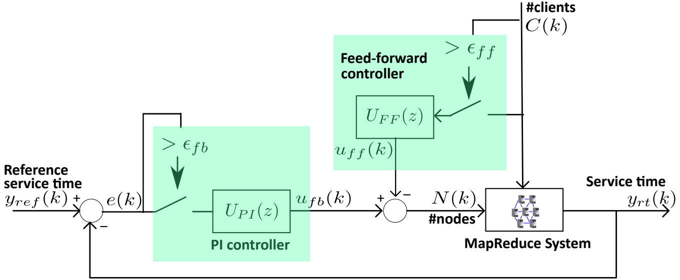
$$u_{fb}(k) = u_{fb}(k-1) + K_p \cdot (e(k) - e(k-1)) + K_i \cdot h \cdot e(k)$$

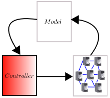
- Event-based feedforward control

$$u_{ff}(k) = -\frac{b_C}{b_N} \cdot C$$



MR-Perf-Cost



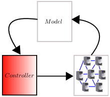


- Event-based PI control

$$u_{fb}(k) = u_{fb}(k-1) + K_p \cdot (e(k) - e(k-1)) + K_i \cdot h \cdot e(k)$$

- Event-based feedforward control

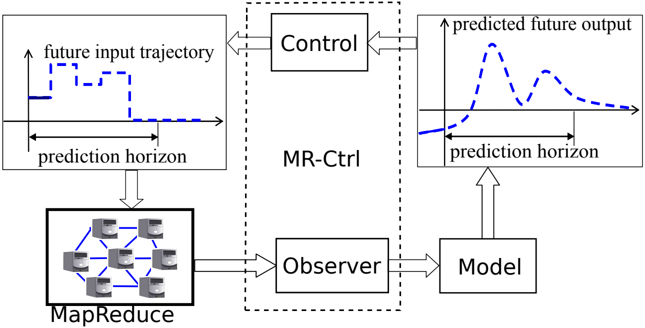
$$u_{ff}(k) = -\frac{b_C}{b_N} \cdot C$$

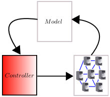


MR-Ctrl

Constrained optimal control

- SLA:
  - Keep response time under a given threshold
  - Keep availability above a given threshold
  - Minimal costs (node count and variation)





MR-Ctrl

Optimisation strategy

- Objective:
  - Optimal control trajectory to ensure SLA
- Optimisation based on objective function:

$$J = \min_{U_k}\{(Y_k - Y_{ref})^T \cdot Q \cdot (Y_k - Y_{ref}) + U_k^T \cdot R \cdot U_k\}$$

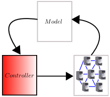
- Subject to constraints: $MC \leq C$ and $N \leq N_{max}$
- Quadratic programming $\rightarrow$ Optimal Control Synthesis

$$J = \min_{u}\{u^T \cdot H \cdot u + 2 \cdot f^T \cdot u\}$$

$$H = \Delta^T \cdot Q \cdot \Delta$$

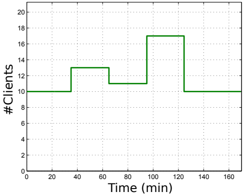
$$f^T = 2 \cdot (X_k^T \cdot \Gamma^T \cdot Q \cdot \Delta - Y_{ref}^T \cdot Q \cdot \Delta)$$

Contents

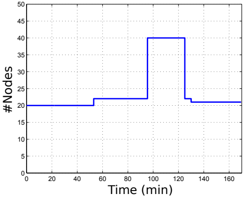


MR-Perf-Cost

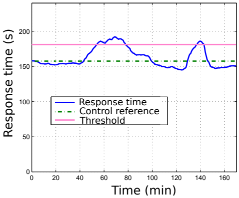
Experimental validation



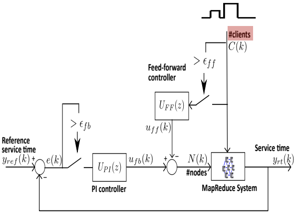
Clients



Nodes

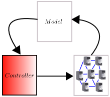


Response time



MR-Perf-Cost

- ✓ Can handle highly dynamic workloads
- ✓ Automatic control solution
- ✓ Minimal costs
- ✓ Easy to implement
- ✗ Single output objectives only



MR-Perf-Cost

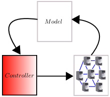
✓ Can handle highly dynamic workloads

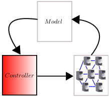
✓ Automatic control solution

✓ Minimal costs

✓ Easy to implement

✗ Single output objectives only



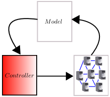


MR-Perf-Cost

- ✓ Can handle highly dynamic workloads
- ✓ Automatic control solution
- ✓ Minimal costs
- ✓ Easy to implement
- ✗ Single output objectives only

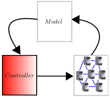
MR-Perf-Cost

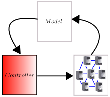
- ✓ Can handle highly dynamic workloads
- ✓ Automatic control solution
- ✓ Minimal costs
- ✓ Easy to implement
- ✗ Single output objectives only



MR-Perf-Cost

- ✓ Can handle highly dynamic workloads
- ✓ Automatic control solution
- ✓ Minimal costs
- ✓ Easy to implement
- ✗ Single output objectives only

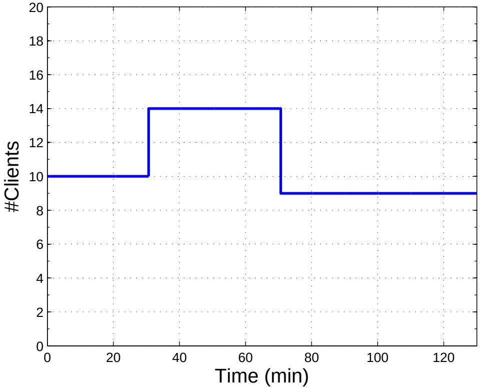




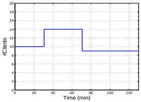
MR-Ctrl

Validation scenario

- Force trade-off between two objectives → #N=40

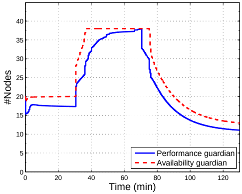


Clients

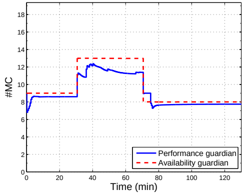


MR-Ctrl

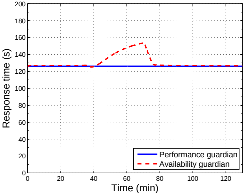
Validation in simulation



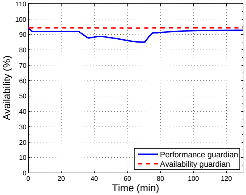
Nodes



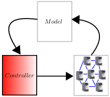
MaxClients

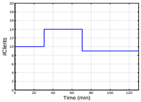


Response time



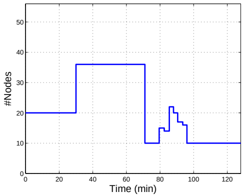
Availability



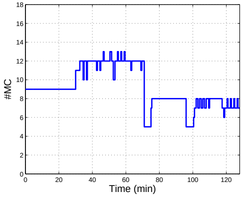


MR-Ctrl

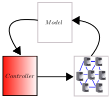
Experimental validation

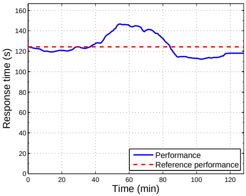


Nodes

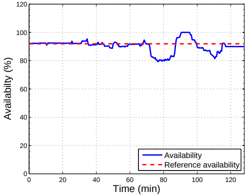


MaxClients

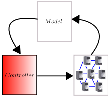




Response time

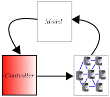


Availability



MR-Ctrl

- ✓ Handle highly dynamic workloads
- ✓ Ensure multiple objectives at the same time
- ✓ Trade-off between contradictory objectives easily quantified
- ✓ Automatic control solution
- ✓ Explicit cost minimisation criteria
- ✓ Handle varying control or output constraints



MR-Ctrl

- ✓ Handle highly dynamic workloads
- ✓ Ensure multiple objectives at the same time
- ✓ Trade-off between contradictory objectives easily quantified
- ✓ Automatic control solution
- ✓ Explicit cost minimisation criteria
- ✓ Handle varying control or output constraints

MR-Ctrl

✓ Handle highly dynamic workloads

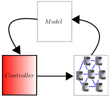
✓ Ensure multiple objectives at the same time

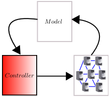
✓ Trade-off between contradictory objectives easily quantified

✓ Automatic control solution

✓ Explicit cost minimisation criteria

✓ Handle varying control or output constraints



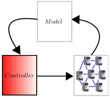


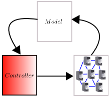
MR-Ctrl

- ✓ Handle highly dynamic workloads
- ✓ Ensure multiple objectives at the same time
- ✓ Trade-off between contradictory objectives easily quantified
- ✓ Automatic control solution
- ✓ Explicit cost minimisation criteria
- ✓ Handle varying control or output constraints

MR-Ctrl

- ✓ Handle highly dynamic workloads
- ✓ Ensure multiple objectives at the same time
- ✓ Trade-off between contradictory objectives easily quantified
- ✓ Automatic control solution
- ✓ Explicit cost minimisation criteria
- ✓ Handle varying control or output constraints





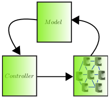
MR-Ctrl

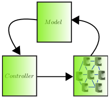
- ✓ Handle highly dynamic workloads
- ✓ Ensure multiple objectives at the same time
- ✓ Trade-off between contradictory objectives easily quantified
- ✓ Automatic control solution
- ✓ Explicit cost minimisation criteria
- ✓ Handle varying control or output constraints

Contents

Conclusions

- Our model:
  - Captures the system dynamics with high accuracy
  - On-line adaptation of the parameters is recommended
- Our control:
  - Handles dynamic workloads
  - Successfully ensures a multi-objective SLA
- General approach





Main contributions

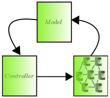
1. An autonomous on-line control framework, that allows to measure and control the performance and availability of a MapReduce cluster
2. Novel approach for the multi-input multi-output modelling of BigData services
3. Automatic control algorithms $\rightarrow$ ensures both performance and dependability objectives, while minimising costs

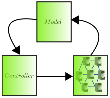
Main contributions

1. An autonomous on-line control framework, that allows to measure and control the performance and availability of a MapReduce cluster

2. Novel approach for the multi-input multi-output modelling of BigData services

3. Automatic control algorithms → ensures both performance and dependability objectives, while minimising costs



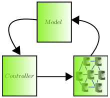


Main contributions

1. An autonomous on-line control framework, that allows to measure and control the performance and availability of a MapReduce cluster

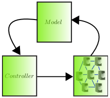
2. Novel approach for the multi-input multi-output modelling of BigData services

3. Automatic control algorithms $\rightarrow$ ensures both performance and dependability objectives, while minimising costs



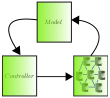
Perspectives

- Other QoS metrics: e.g. throughput, reliability
- Other service aspects: e.g. security
- Other BigData services: e.g. YARN, SPARK
- Other infrastructures: e.g. systems of systems (EU Project AMADEOS)
- Package of off the shelf control solutions



Perspectives

- Other QoS metrics: e.g. throughput, reliability
- Other service aspects: e.g. security
- Other BigData services: e.g. YARN, SPARK
- Other infrastructures: e.g. systems of systems (EU Project AMADEOS)
- Package of off the shelf control solutions

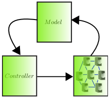


Perspectives

- Other QoS metrics: e.g. throughput, reliability
- Other service aspects: e.g. security
- Other BigData services: e.g. YARN, SPARK
- Other infrastructures: e.g. systems of systems (EU Project AMADEOS)
- Package of off the shelf control solutions

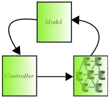
Perspectives

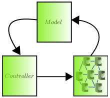
- Other QoS metrics: e.g. throughput, reliability
- Other service aspects: e.g. security
- Other BigData services: e.g. YARN, SPARK
- Other infrastructures: e.g. systems of systems (EU Project AMADEOS)
- Package of off the shelf control solutions



Perspectives

- Other QoS metrics: e.g. throughput, reliability
- Other service aspects: e.g. security
- Other BigData services: e.g. YARN, SPARK
- Other infrastructures: e.g. systems of systems (EU Project AMADEOS)
- Package of off the shelf control solutions





Main results

- International conference papers:
  - "Application du contrôle pour garantir la performance des systèmes Big Data", ComPAS 2014, Neuchâtel, Switzerland, April 22-25, 2014
  - "A Control Approach for Performance of Big Data Systems", IFAC 2014, Cape-Town, South Africa, August 24-29, 2014
- Journal papers:
  - "Feedback Autonomic Provisioning for Guaranteeing Performance in MapReduce Systems" IEEE Transactions on Cloud Computing (accepted)
  - "Event-Based Control in BigData Cloud Systems: Application to MapReduce" IEEE Transactions on Industrial Informatics (submitted)