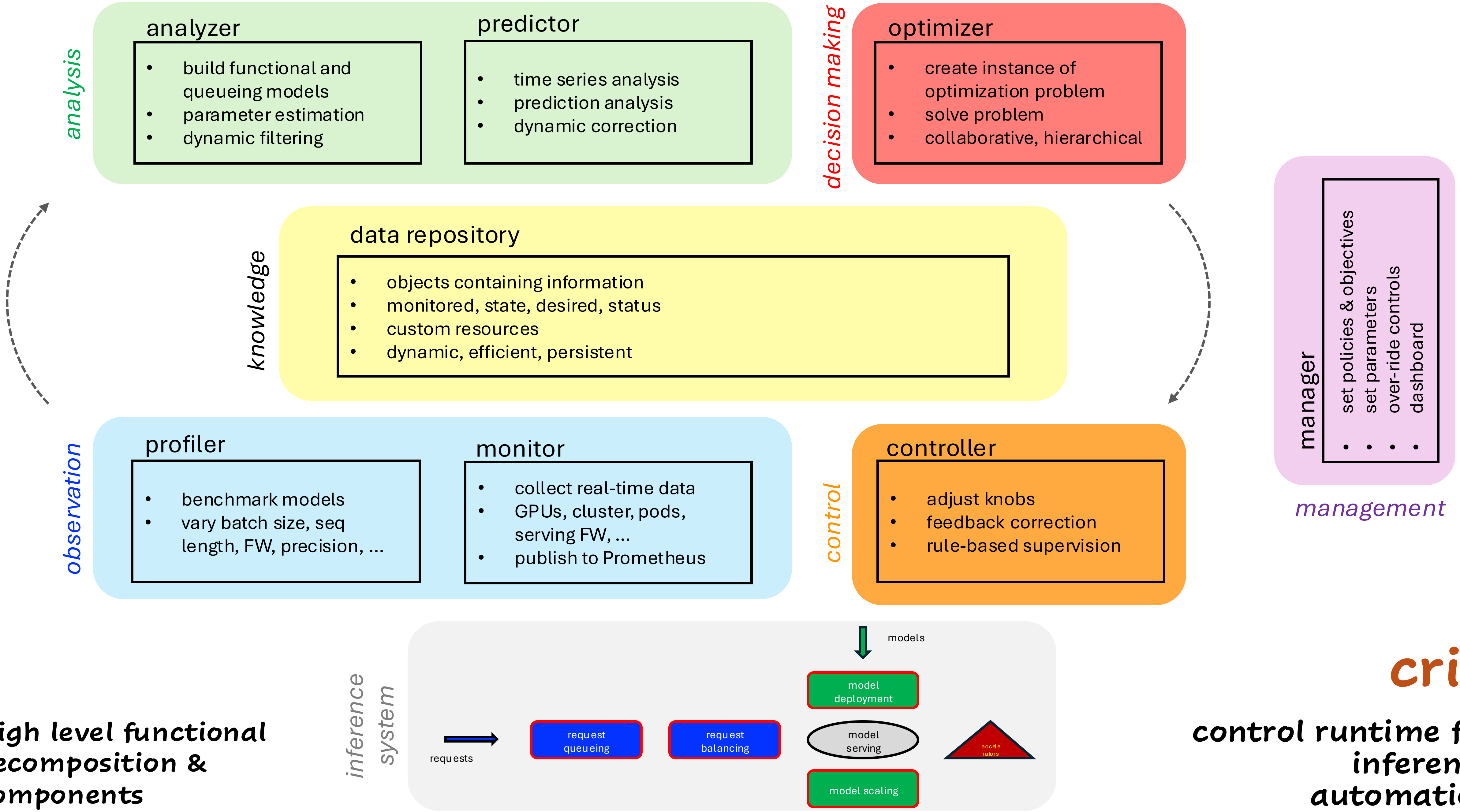
inferno
inference platform,
naturally optimized

Inferencing operational optimization

Model-based, dynamic global optimization

Asser Tantawi
IBM Research

16 August 2024

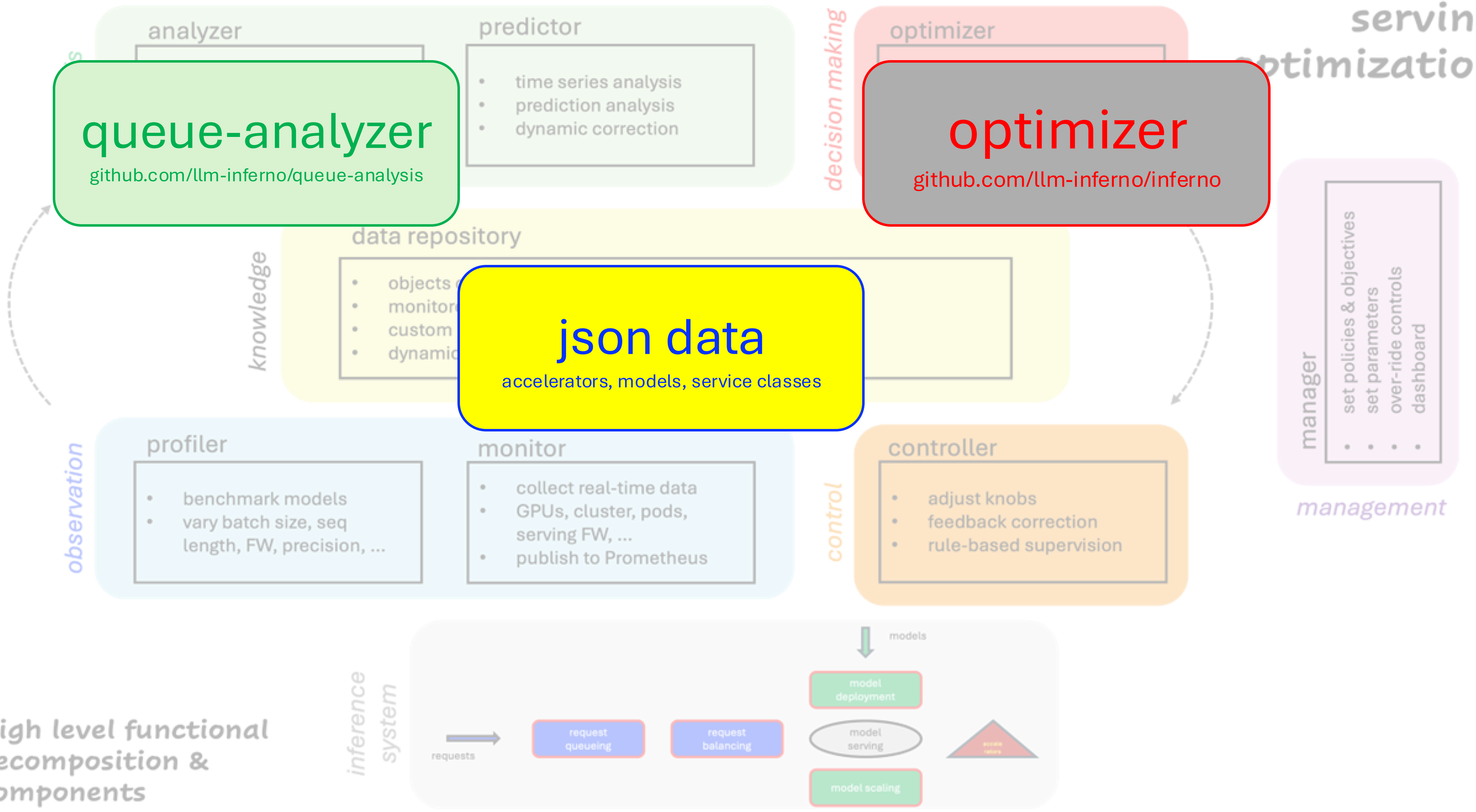


High level functional decomposition & components

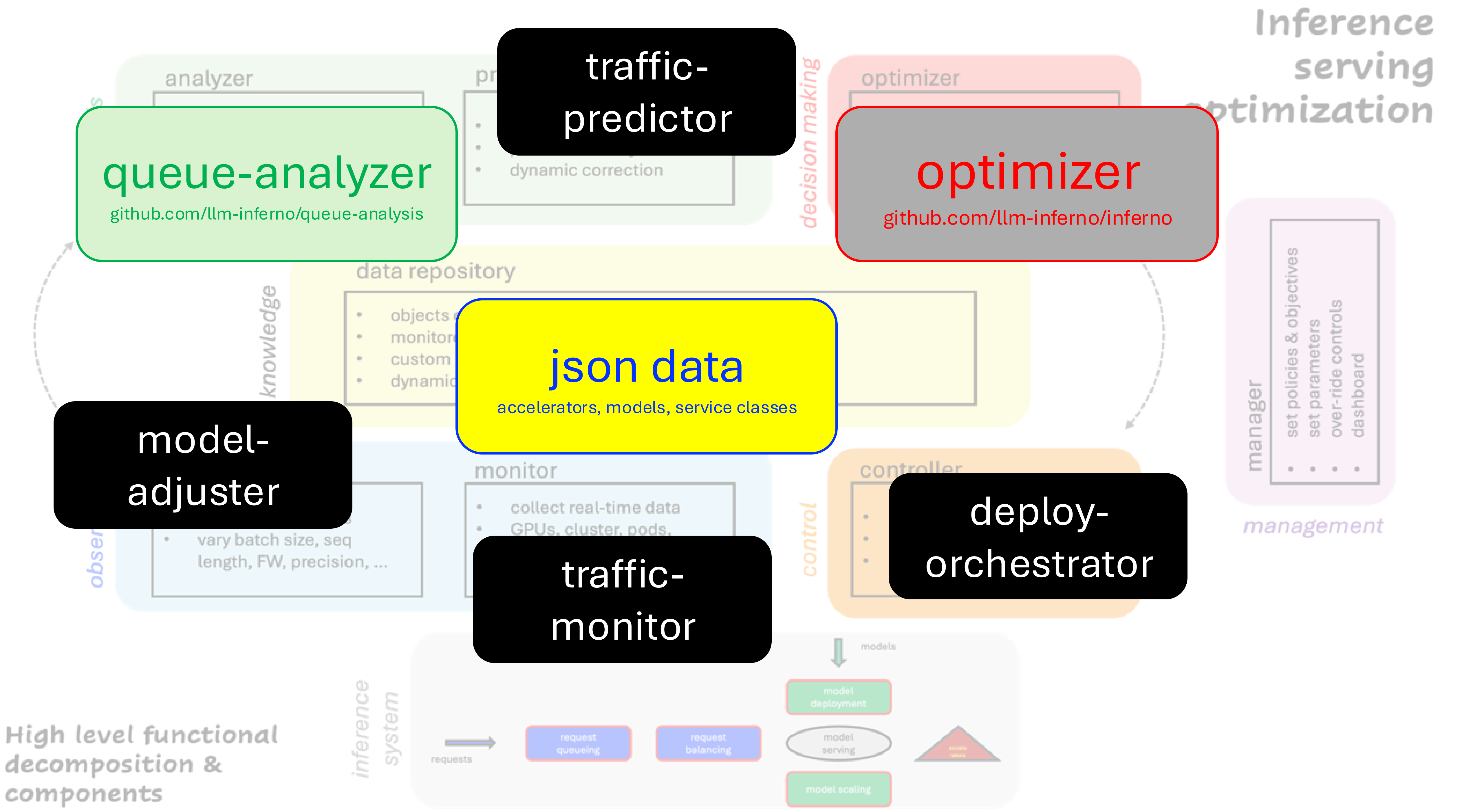
cria

control runtime for inference automation

Inference serving optimization



High level functional decomposition & components


Inference
serving
queue-analyzer
github.com/llm-inferno/queue-analysis
traffic-
predictor
optimizer
github.com/llm-inferno/inferno
analyzer
optimizer
dynamic correction
decision making
data repository
knowledge
objects
monitor
custom
dynamic
json data
accelerators, models, service classes
model-
adjuster
vary batch size, seq
length, FW, precision, ...
monitor
collect real-time data
GPUs, cluster, pods,
traffic-
monitor
controller
control
deploy-
orchestrator
manager
set policies & objectives
set parameters
over-ride controls
dashboard
management
inference
system
models
model
deployment
requests
request
queueing
request
balancing
model
serving
model scaling
High level functional
decomposition &
components

Problem scope

models

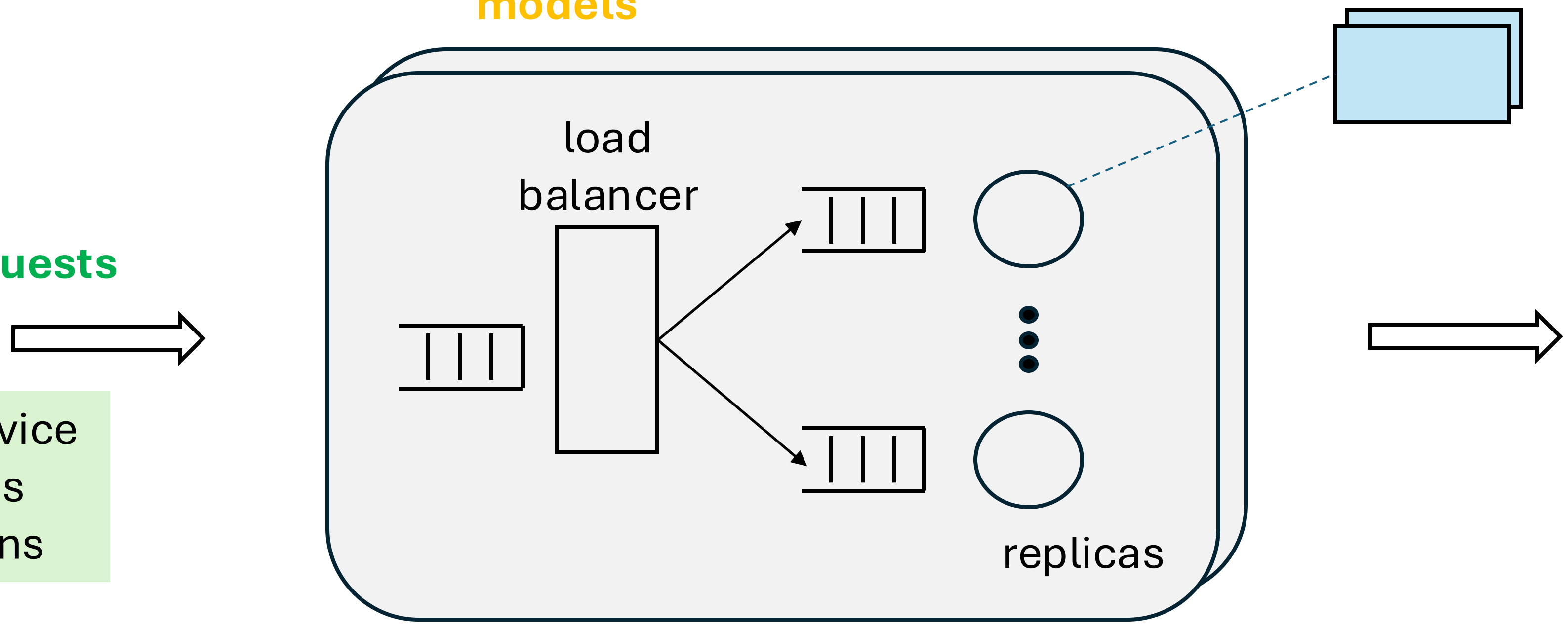
- memory demand
- performance characteristics (delay, throughput)

GPU types

- cost
- memory size
- memory BW

requests

- classes of service
- arrival patterns
- service patterns



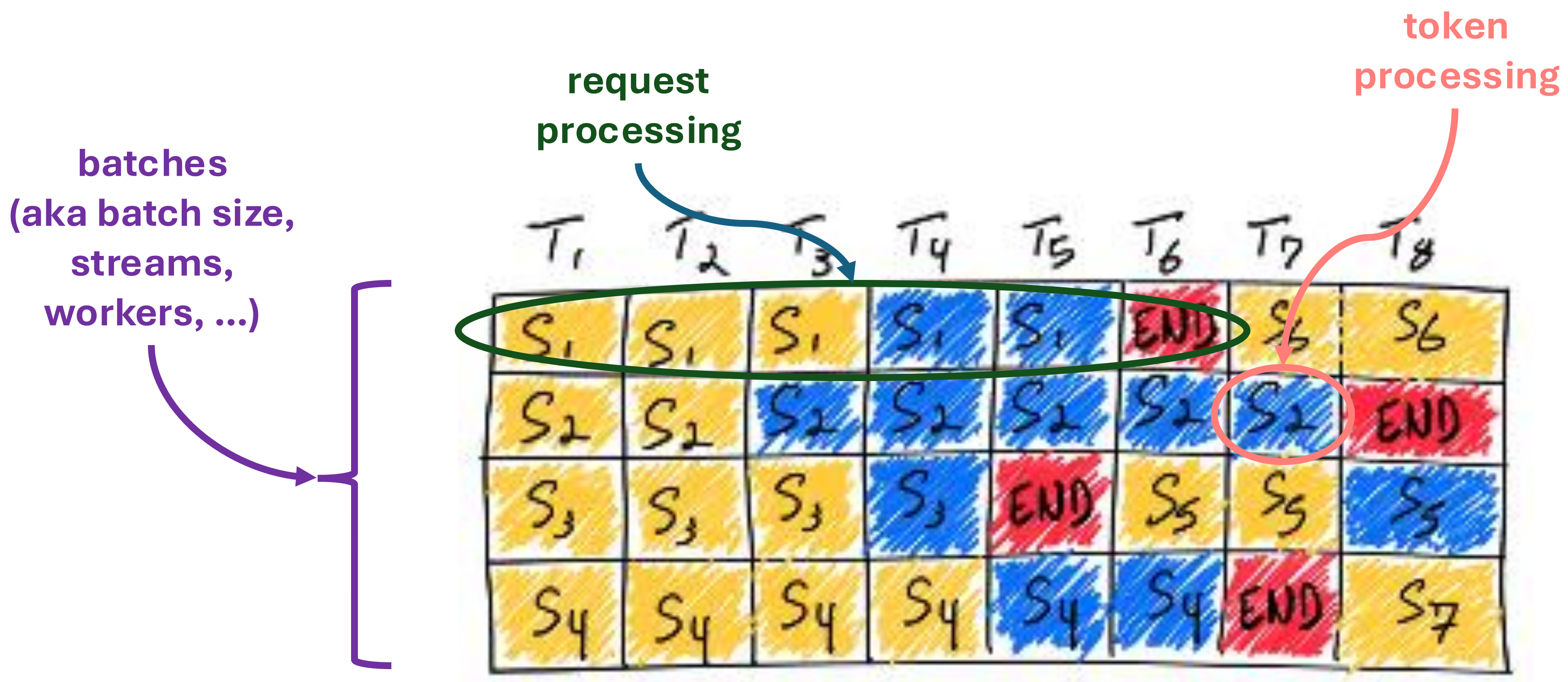
objectives

- min cost
- satisfy SLOs
- fairness, slack, power

decision variables

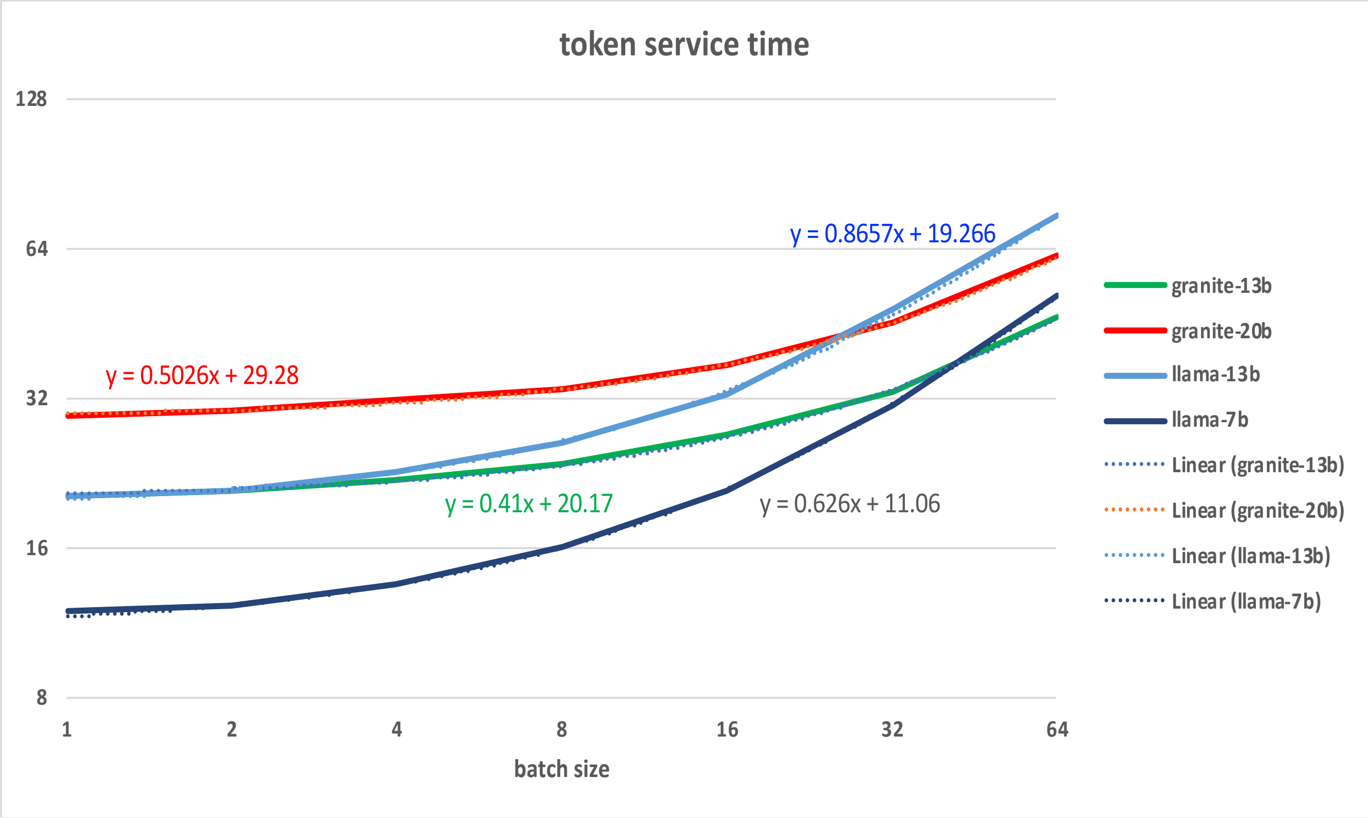
- GPU type
- number of replicas
- batch size

Request batching



N	batch size
$\tau(N)$	token service (processing) time (ITL)
K	number of tokens per request

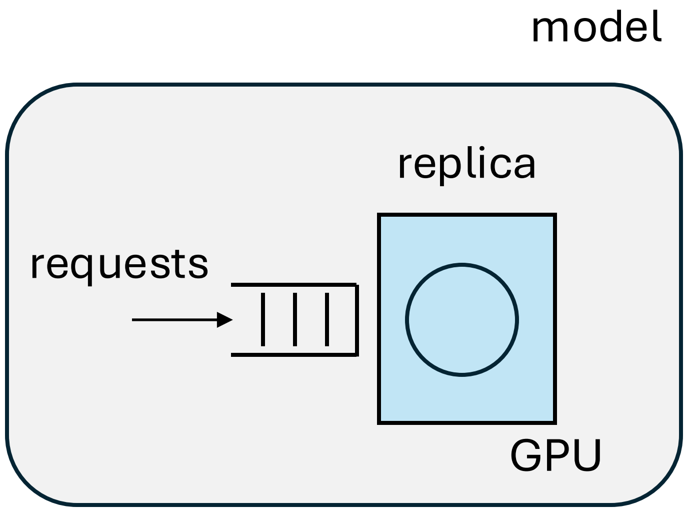
Fitting token service time

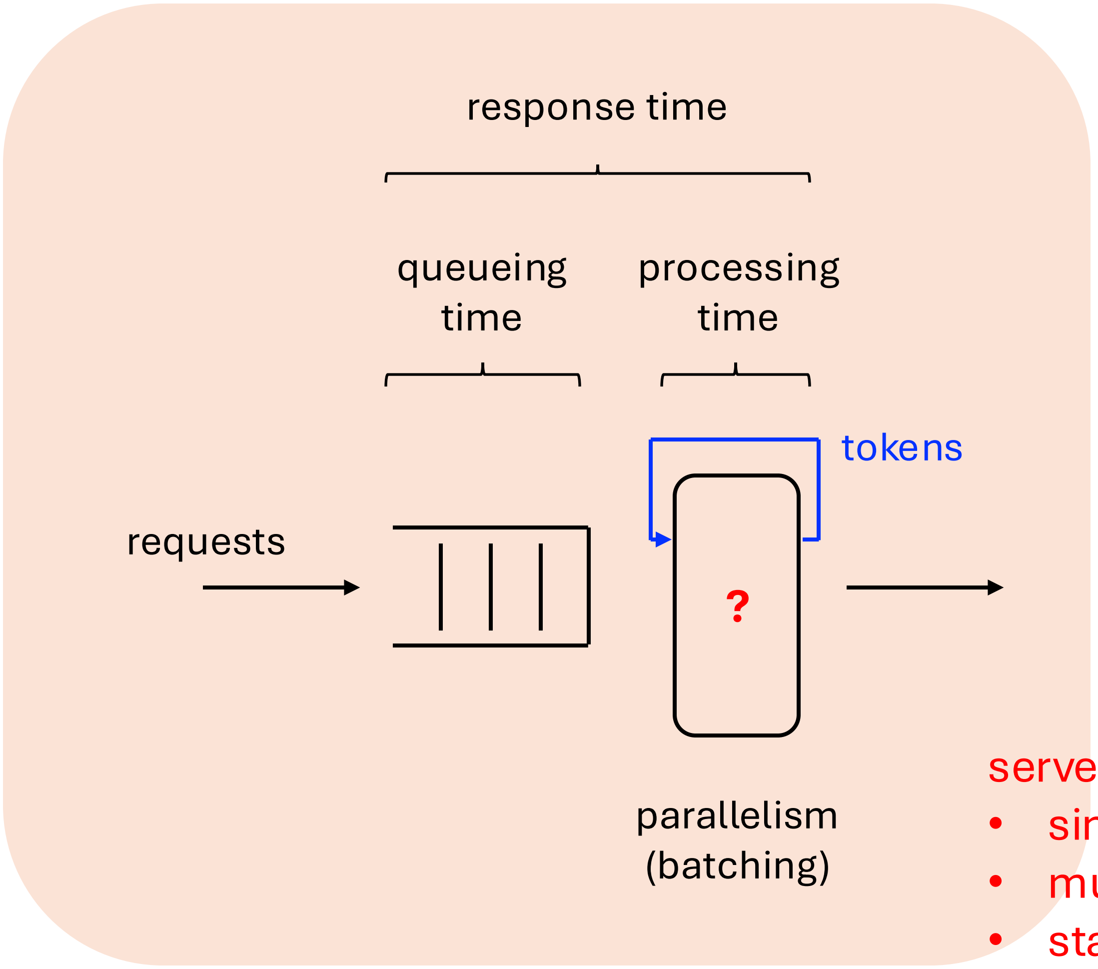


precision	fp16
accelerator	A100
tokens	1024
back	vLLM
accCount	1

$$\tau(N) \approx \alpha + \beta\, N$$

Queueing model of a model serving instance (replica)





avg request processing time =
avg num tokens per request *
avg token service time

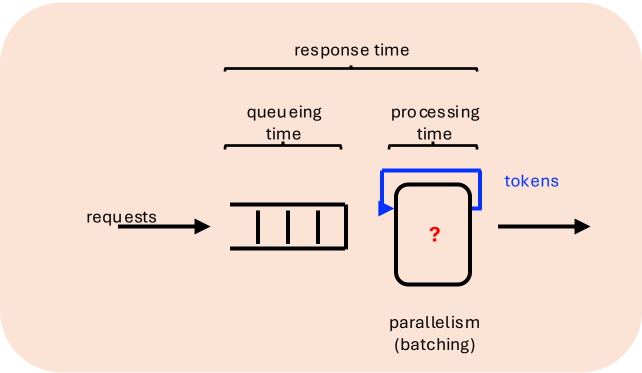
load = request rate * avg request processing time

avg token delay = avg request response time /
avg num tokens

server structure?

- single server
- multiple server
- state-dependent server
- queueing network

modeling batching

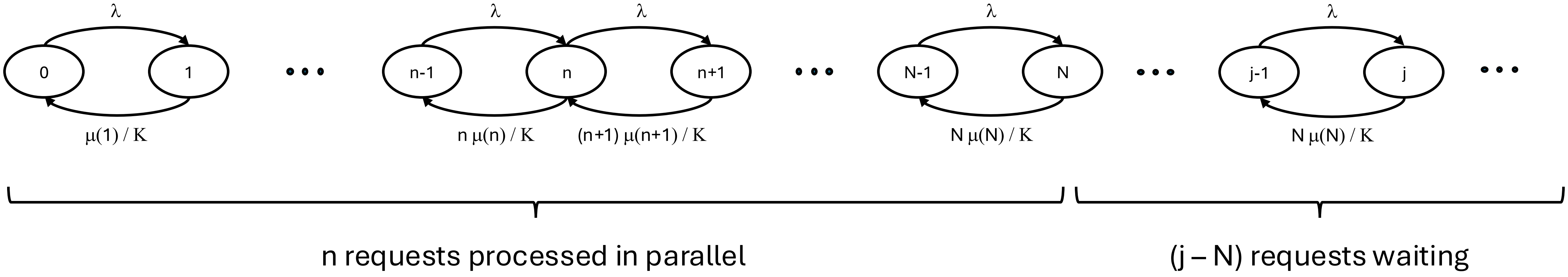


- Markovian assumptions

- state: number of requests in process
- λ request arrival rate
- K average number of tokens per request
- $\tau(n)$ average token service time given n batches
- N batch size (maximum)

$$\mu(n) = \frac{1}{\tau(n)}$$

$$\tau(n) \approx \alpha + \beta n$$



rate limiting

requestRate	36.613	/min	p(0)	3.0013E-09		avgNumSystem	33.77
lambda	0.00061	/msec	maxN	200		avgNumServer	33.16
K	1024		pFull	1.0758E-11		avgNumQueue	0.61
N	48		effecTput	36.61299	req/m	utilization	0.691
alpha	19			0.00061	/msec	avgRespTime	55,349
beta	1		P(N)	0.92202		avgSrvTime	54,349
lambda*K	0.62486163					avgWaitTime	1,000
						avgTokenServTime	53.1

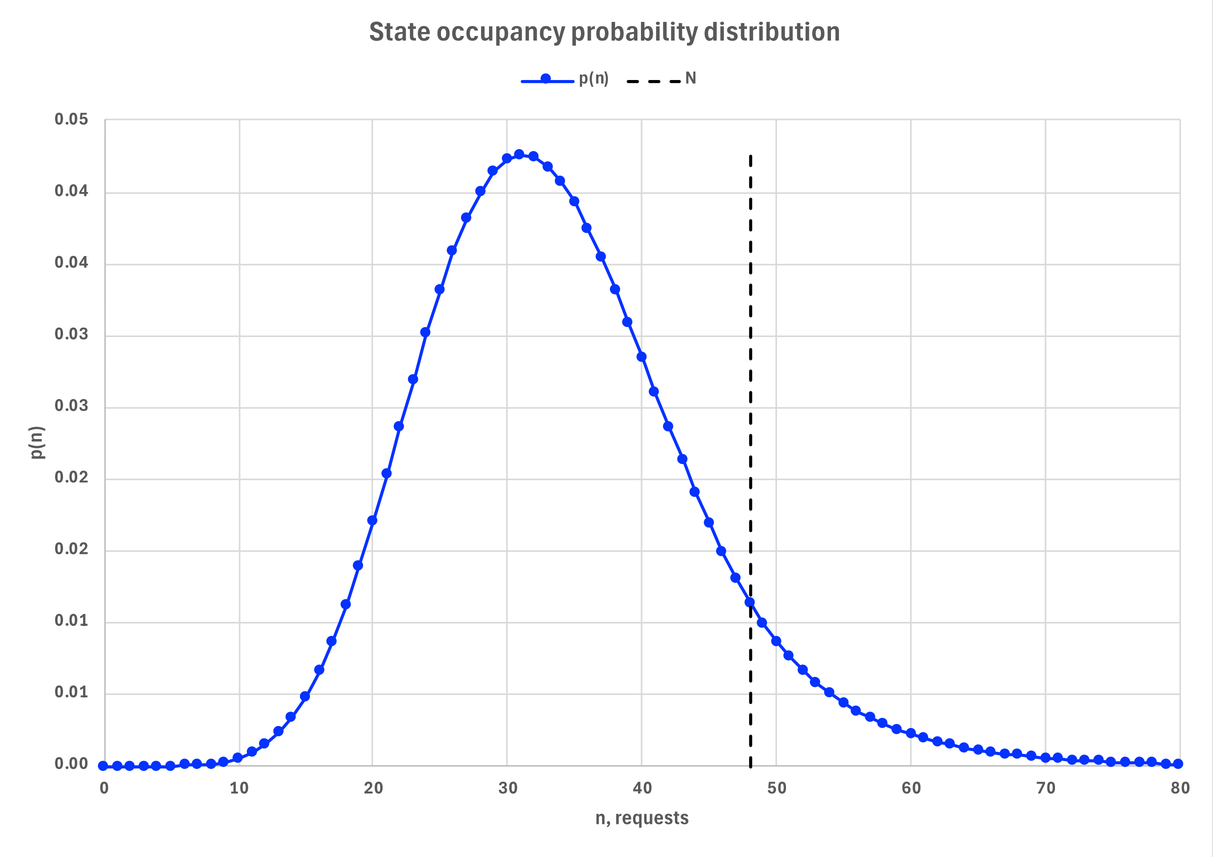
Find

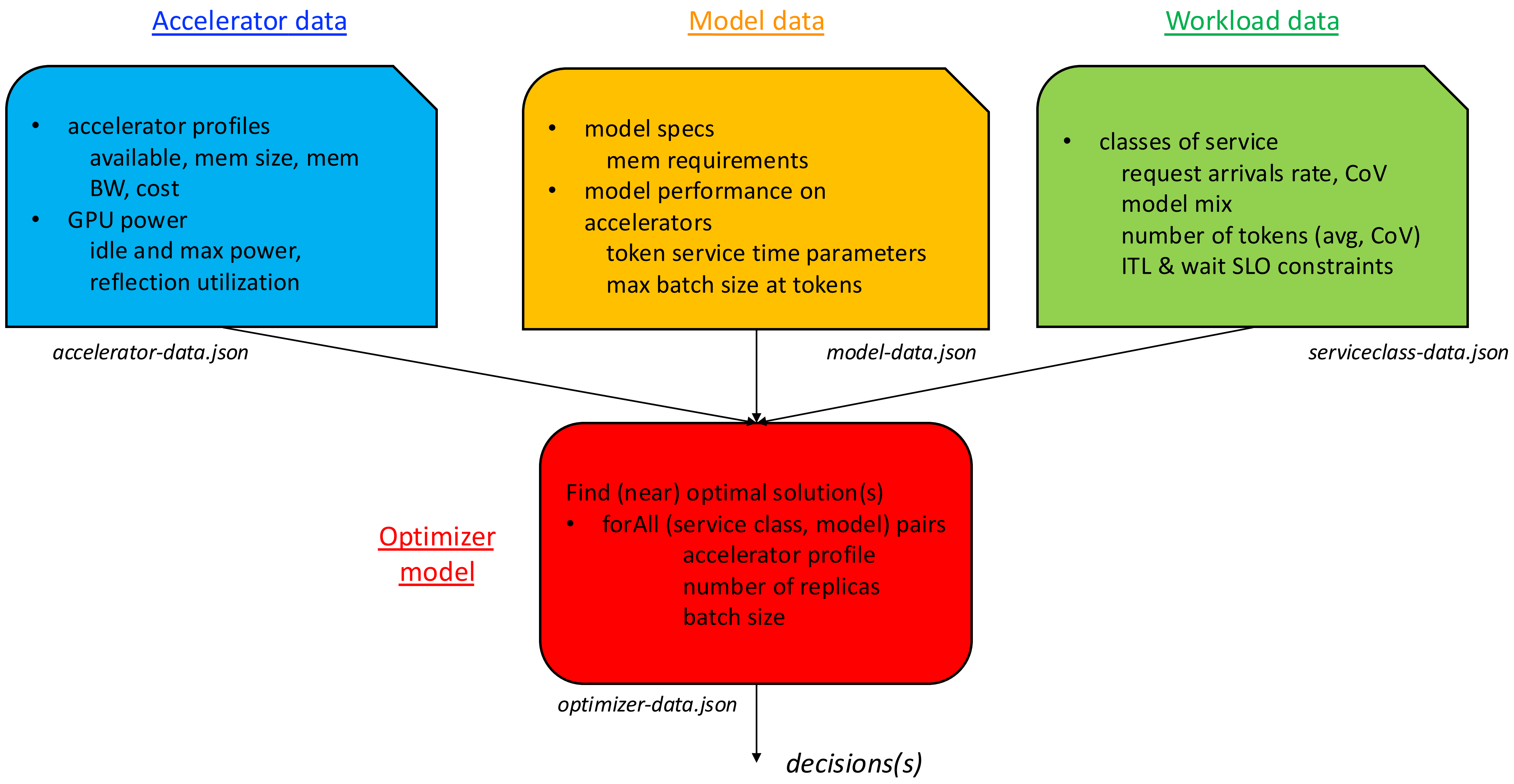
maximum request rate

s.t.

avgWaitTime <= target

target = 1.0 sec
requestRate = 36.613 req/min




Accelerator data
Model data
Workload data
accelerator profiles
available, mem size, mem BW, cost
GPU power
idle and max power, reflection utilization
model specs
mem requirements
model performance on accelerators
token service time parameters
max batch size at tokens
classes of service
request arrivals rate, CoV
model mix
number of tokens (avg, CoV)
ITL & wait SLO constraints
accelerator-data.json
model-data.json
serviceclass-data.json
Optimizer model
Find (near) optimal solution(s)
forAll (service class, model) pairs
accelerator profile
number of replicas
batch size
optimizer-data.json
decisions(s)

model-data.json

serviceclass-data.json

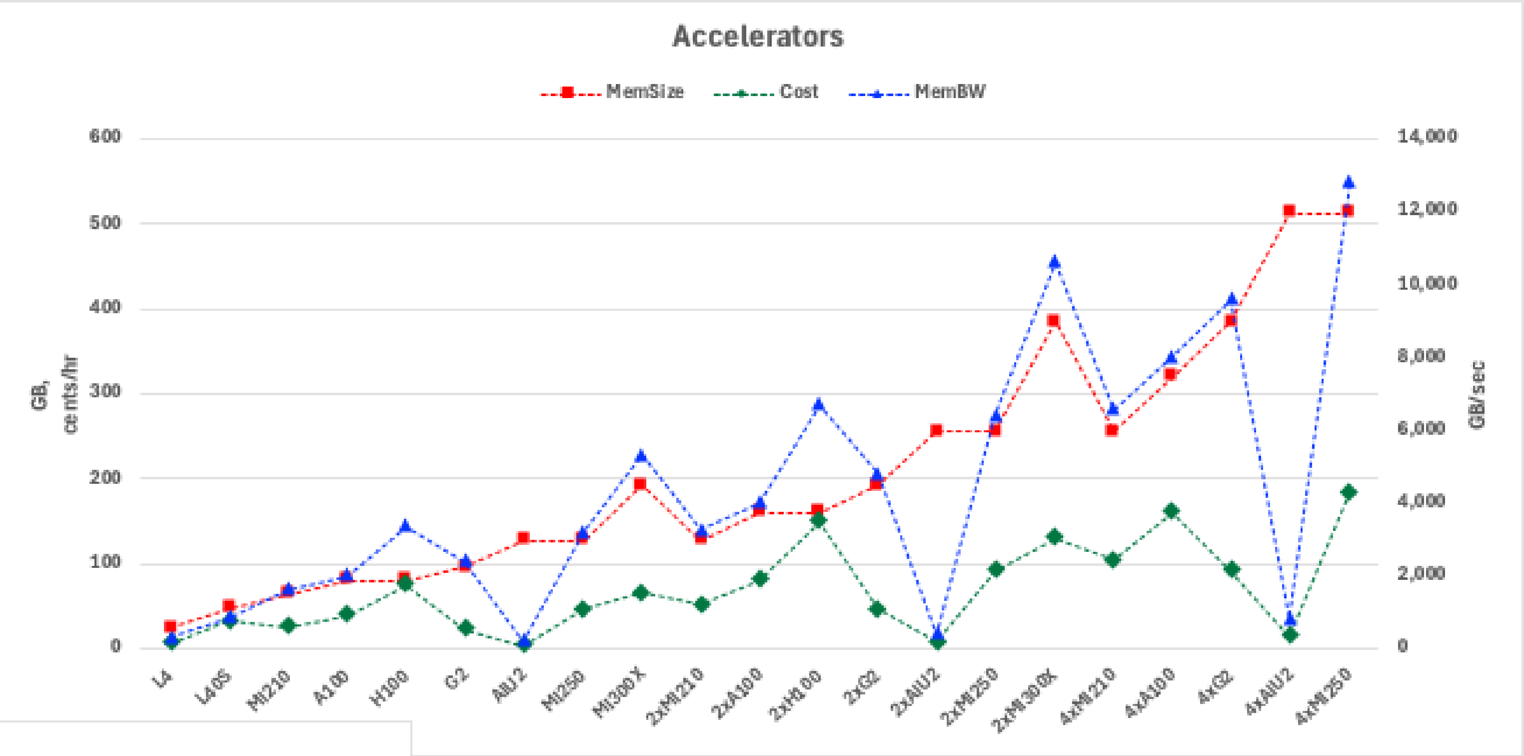
```json
"MI300X": {
    "type": "MI300X",
    "multiplicity": 1,
    "memSize": 192,
    "memBW": 5300,
    "power" : {
        "idle": 220,
        "full": 750,
        "midPower": 650,
        "midUtil": 0.6
    },
    "cost": 65.00
},
"2xA100": {
    "type": "A100",
    "multiplicity": 2,
    "memSize": 160,
    "memBW": 4000,
    "power" : {
        "idle": 300,
        "full": 800,
        "midPower": 640,
        "midUtil": 0.6
    },
    "cost": 80.00
},
```

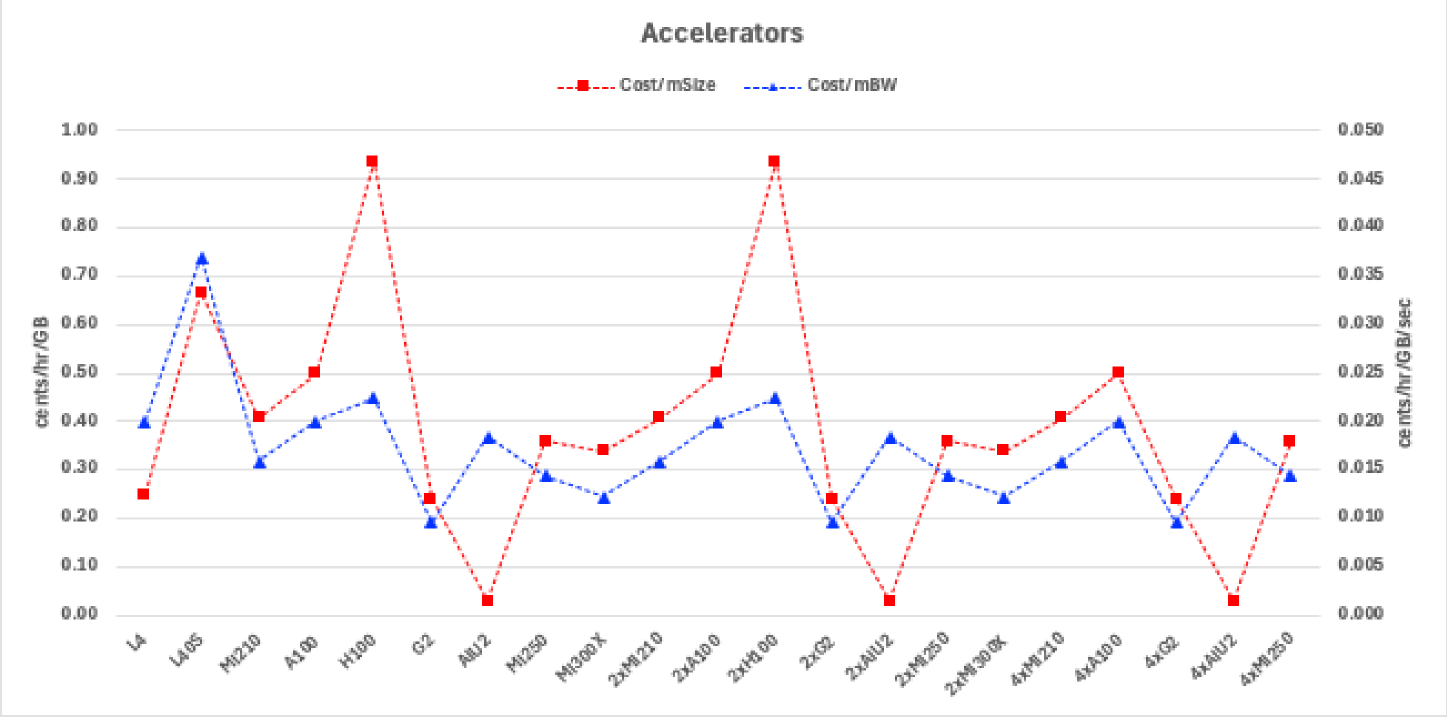
accelerator-data.json

```json
{
  "name": "granite_20b",
  "memSize": 60,
  "perfData": [
    {
      "name": "AIU2",
      "alpha": 297.8,
      "beta": 5,
      "maxBatchSize": 38,
      "atTokens": 512
    },
    {
      "name": "L4",
      "alpha": 198.53,
      "beta": 3.33,
      "maxBatchSize": 7,
      "atTokens": 512
    },
```

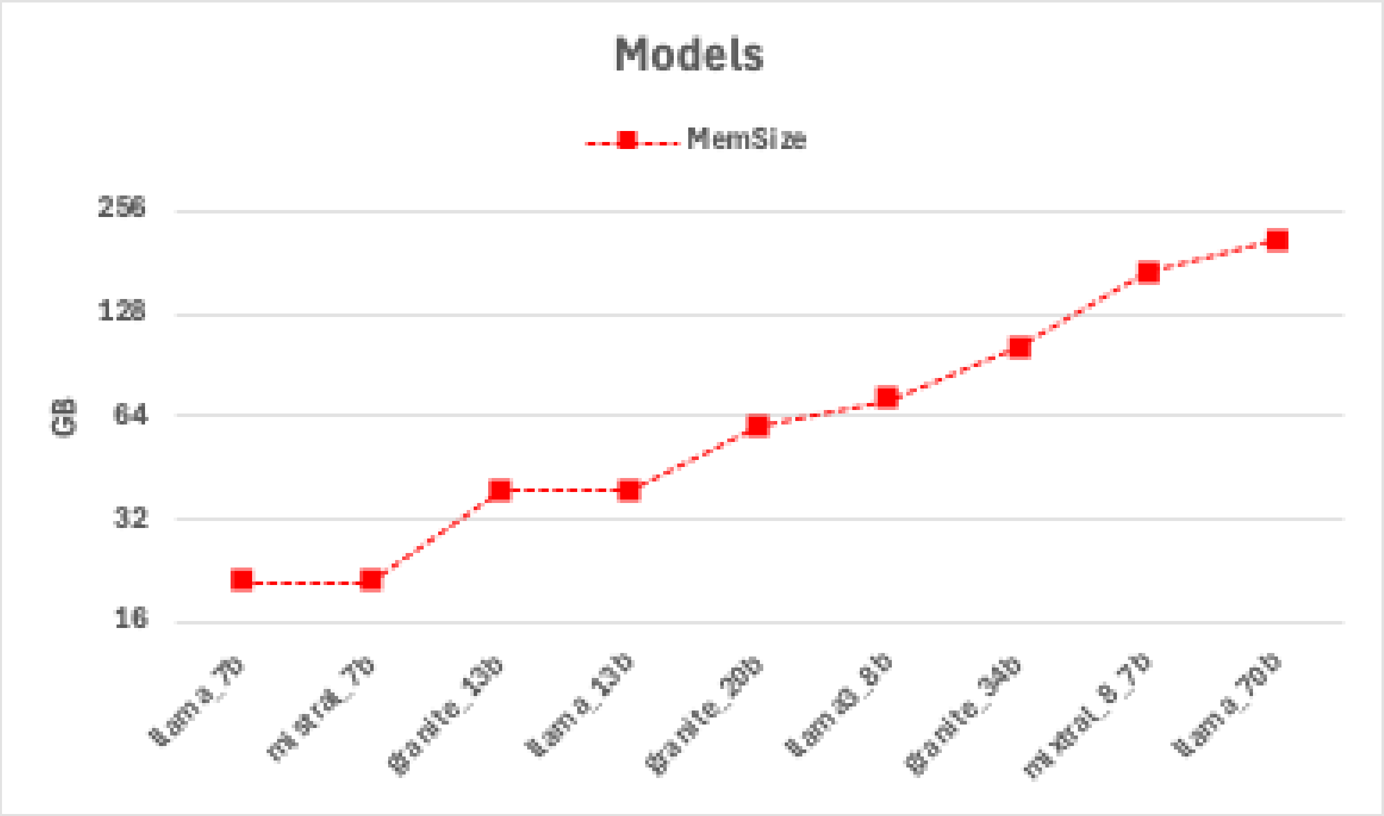
```json
"name": "Premium",
"load": [
    {
        "name": "llama_7b",
        "slo-itl": 40,
        "slo-ttw": 500,
        "arrivalRate": 120,
        "avgLength": 1024,
        "arrivalCOV": 1.5,
        "serviceCOV": 1.5
    },
    {
        "name": "llama_13b",
        "slo-itl": 80,
        "slo-ttw": 500,
        "arrivalRate": 120,
        "avgLength": 1024,
        "arrivalCOV": 1.5,
        "serviceCOV": 1.5
    },
```

Accelerators Specs



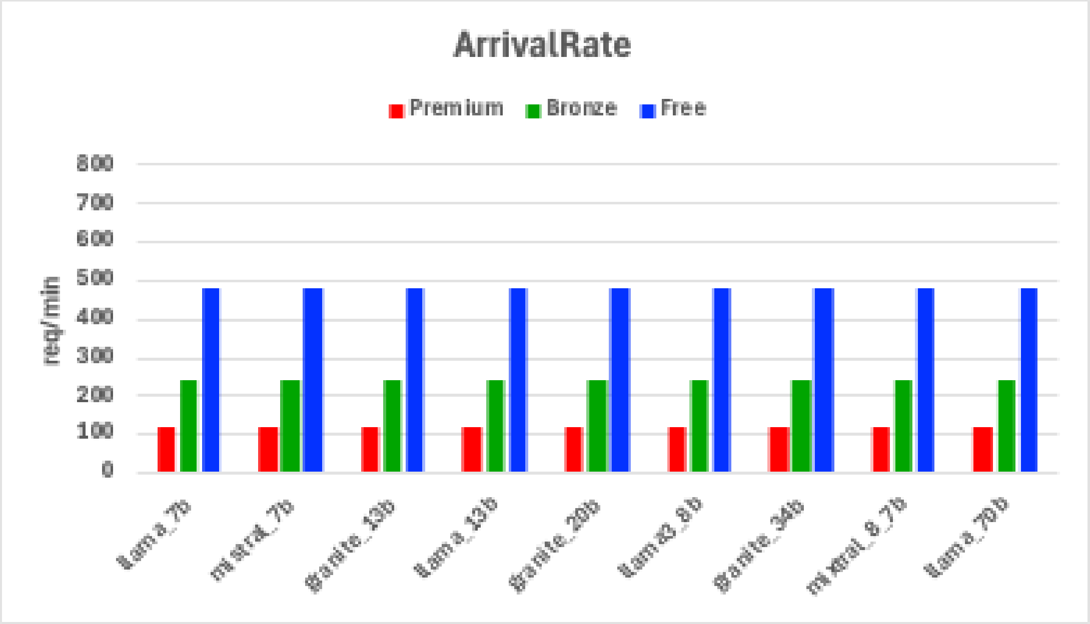


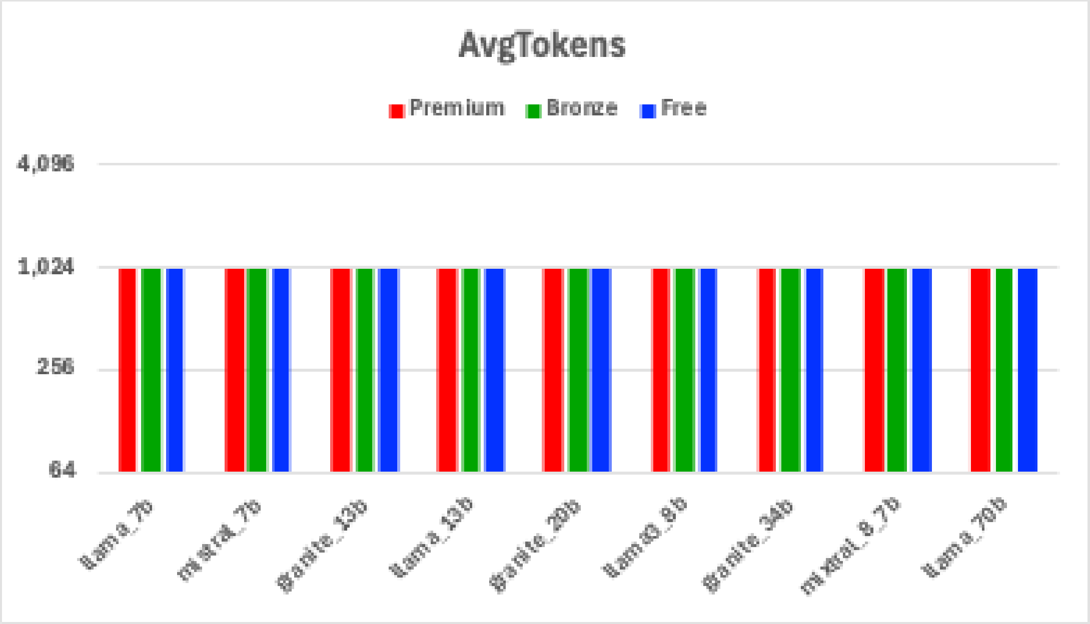
Models Specs



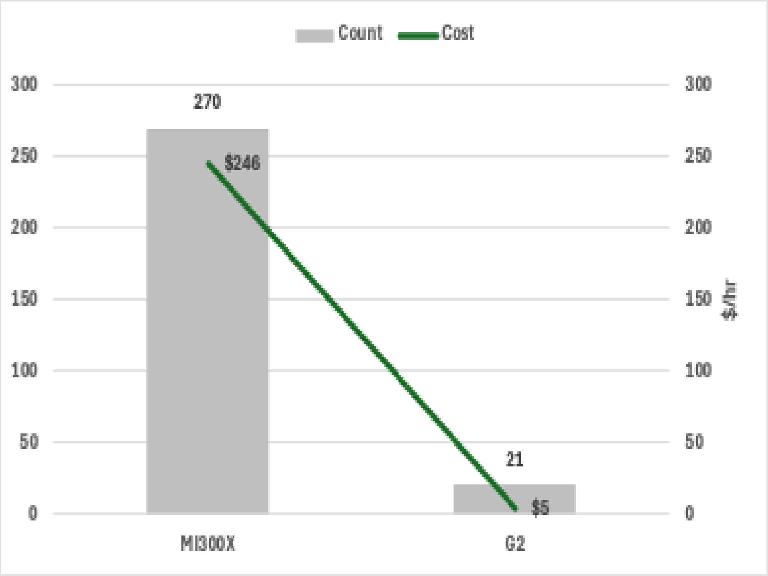
Unlimited accelerators

- capacity planning
- cloud deployment
- separable optimization

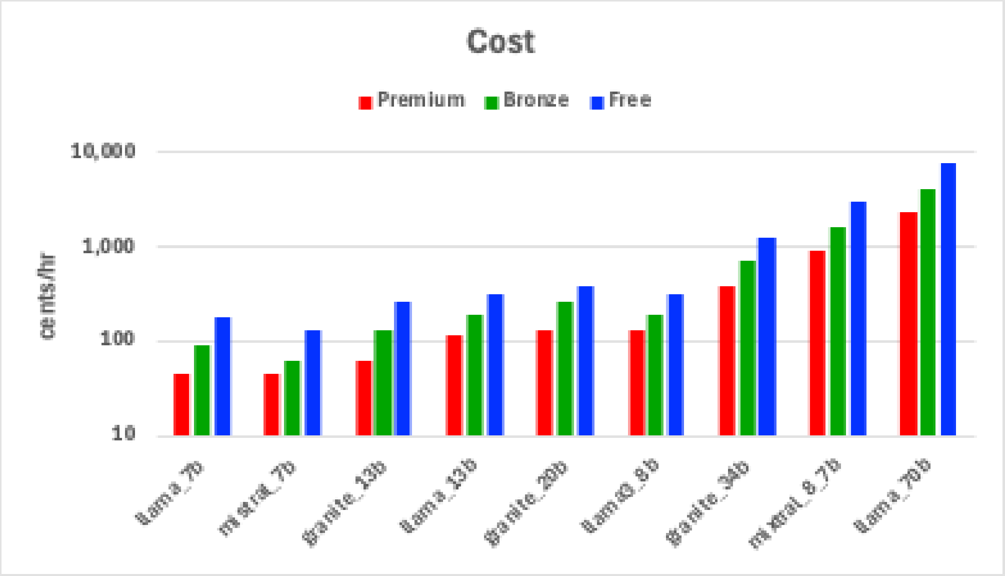


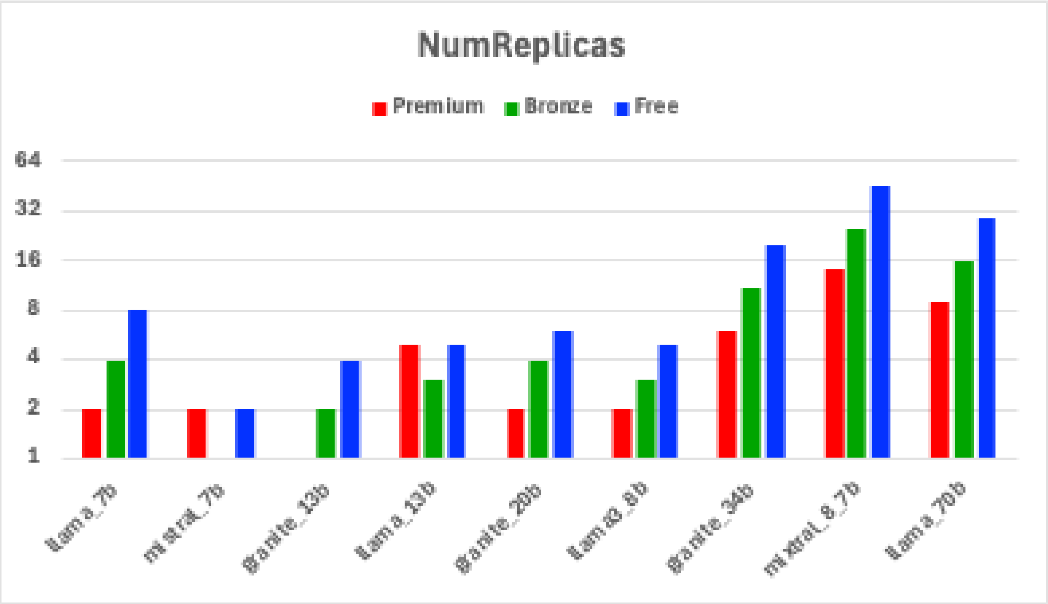


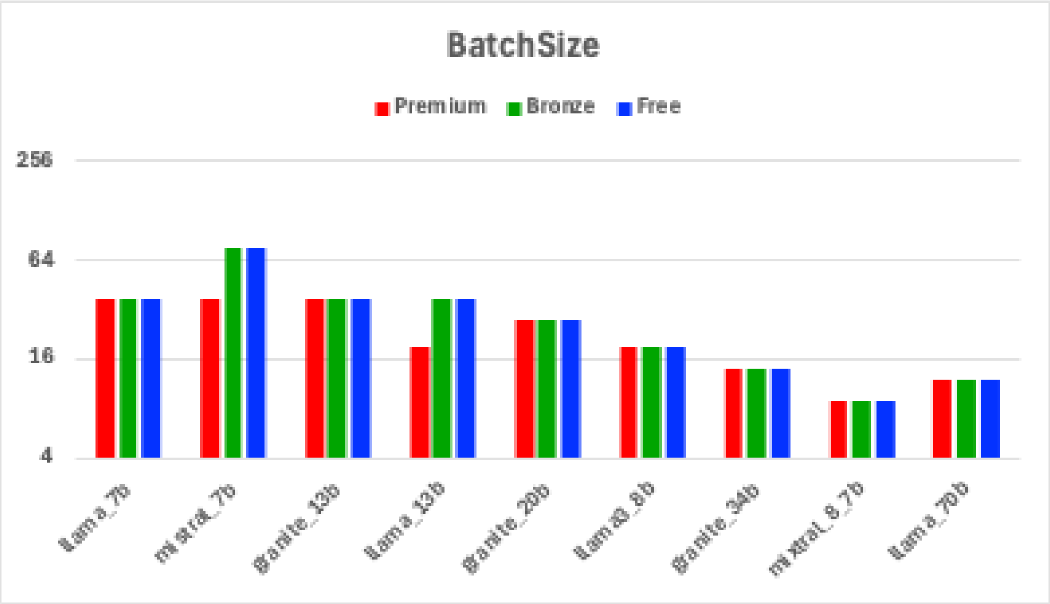
Accelerator			
	Premium	Bronze	Free
llama_7b	G2	G2	G2
mistral_7b	G2	MI300X	MI300X
granite_13b	MI300X	MI300X	MI300X
llama_13b	G2	MI300X	MI300X
granite_20b	MI300X	MI300X	MI300X
llama3_8b	MI300X	MI300X	MI300X
granite_34b	MI300X	MI300X	MI300X
mixtral_8_7b	MI300X	MI300X	MI300X
llama_70b	2xMI300X	2xMI300X	2xMI300X

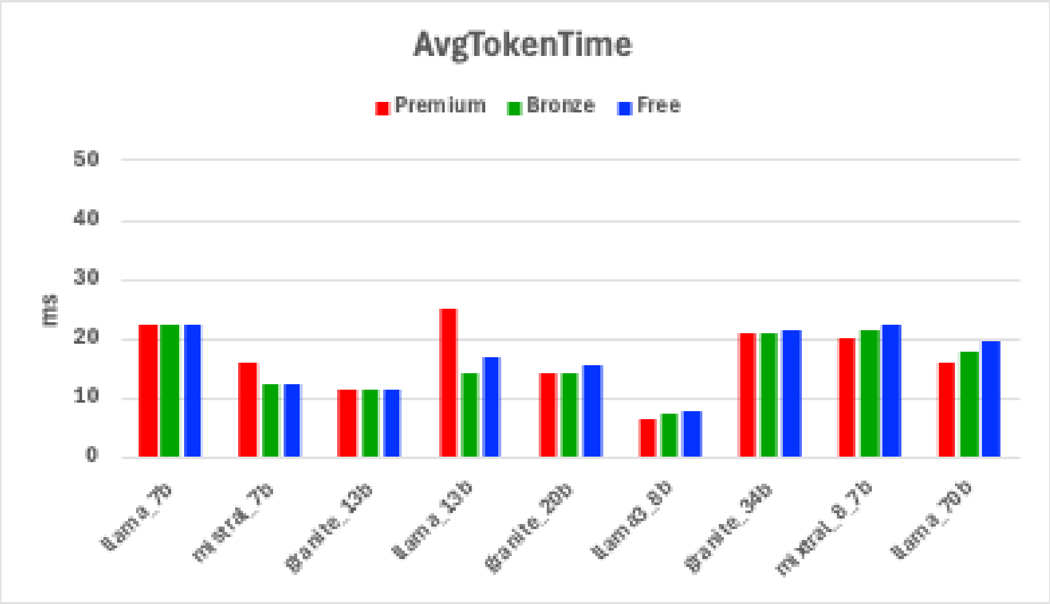


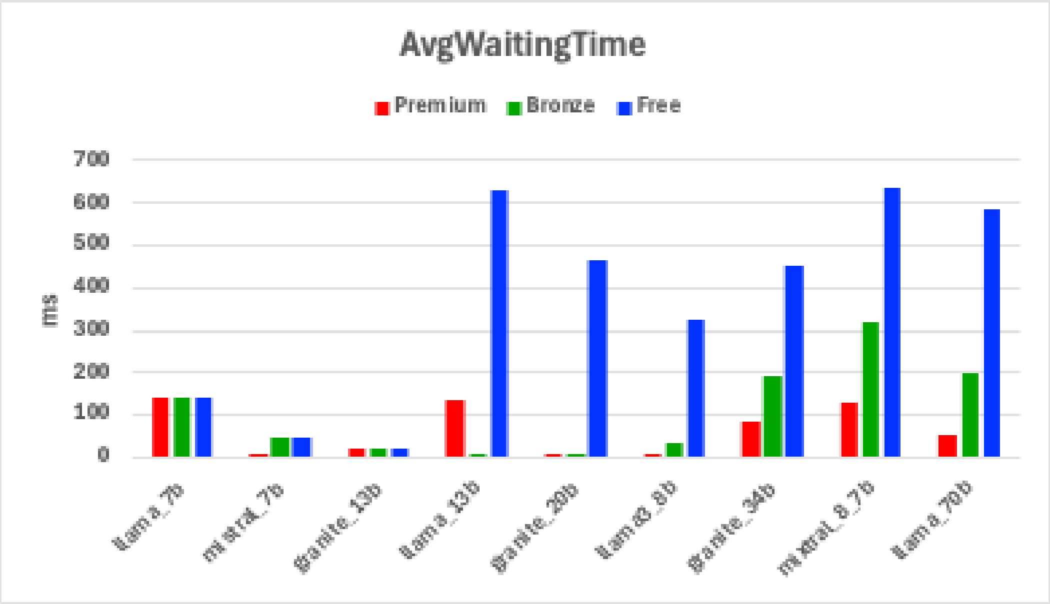
TotalCost
25,053.00




NumReplicas
Premium
Bronze
Free
64
32
16
8
4
2
1
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

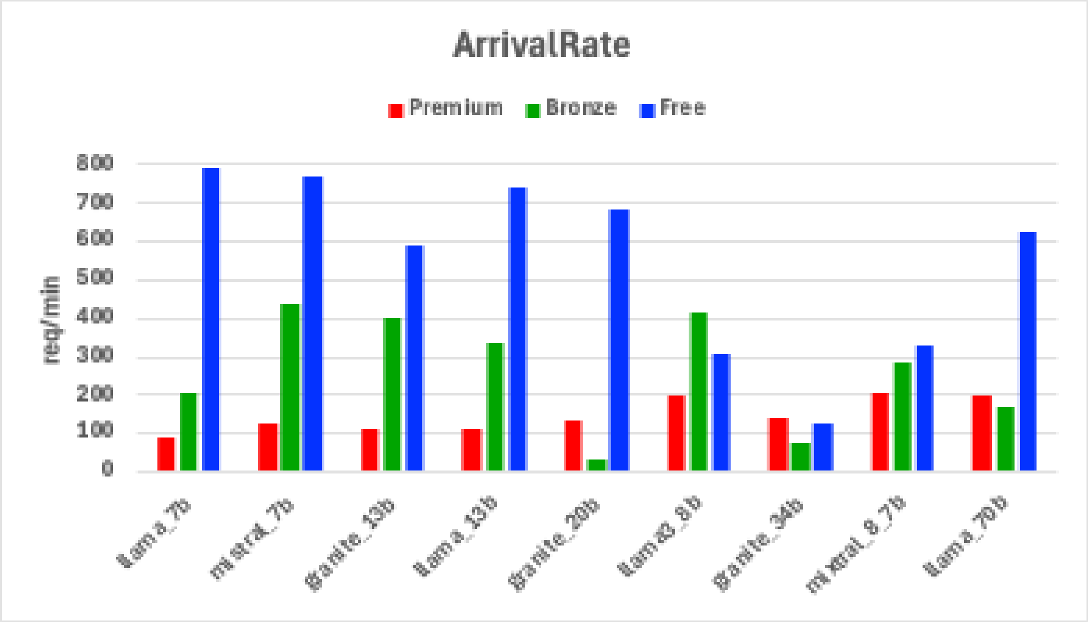

BatchSize
Premium
Bronze
Free
256
64
16
4
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

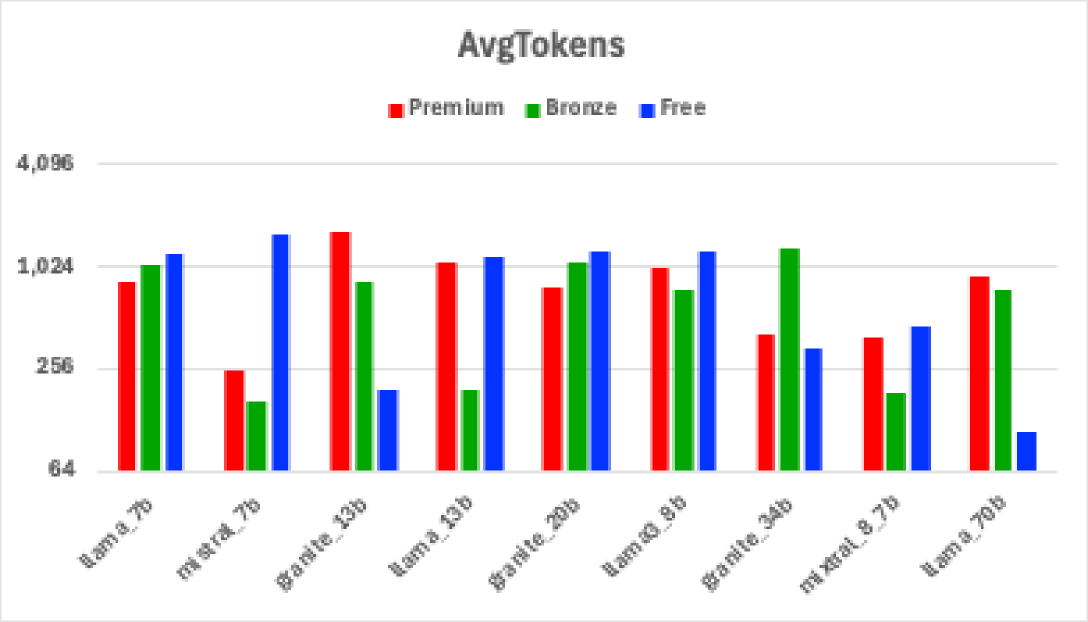

AvgTokenTime
Premium
Bronze
Free
50
40
30
20
10
0
ms
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b


AvgWaitingTime
Premium
Bronze
Free
700
600
500
400
300
200
100
0
ms
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

Unlimited accelerators - Dynamic

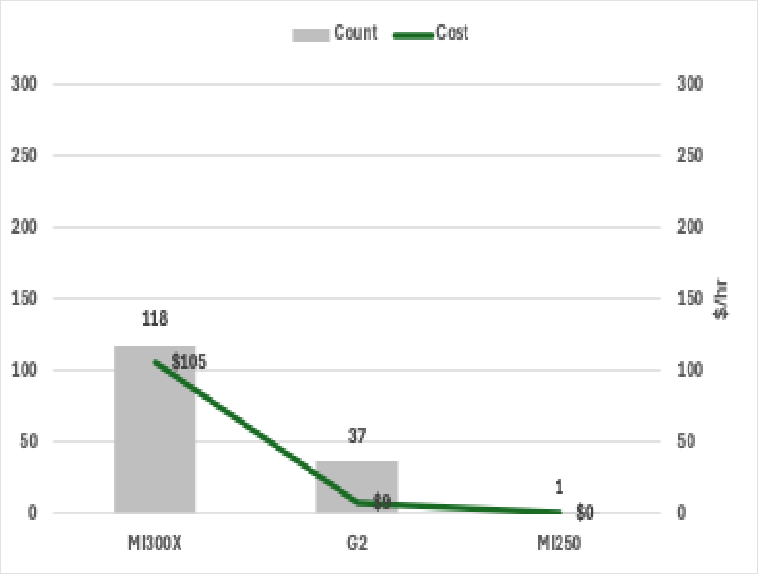
- change in request rates
- change in request lengths
- change/scale accelerators
- minimize change in accelerators and cost



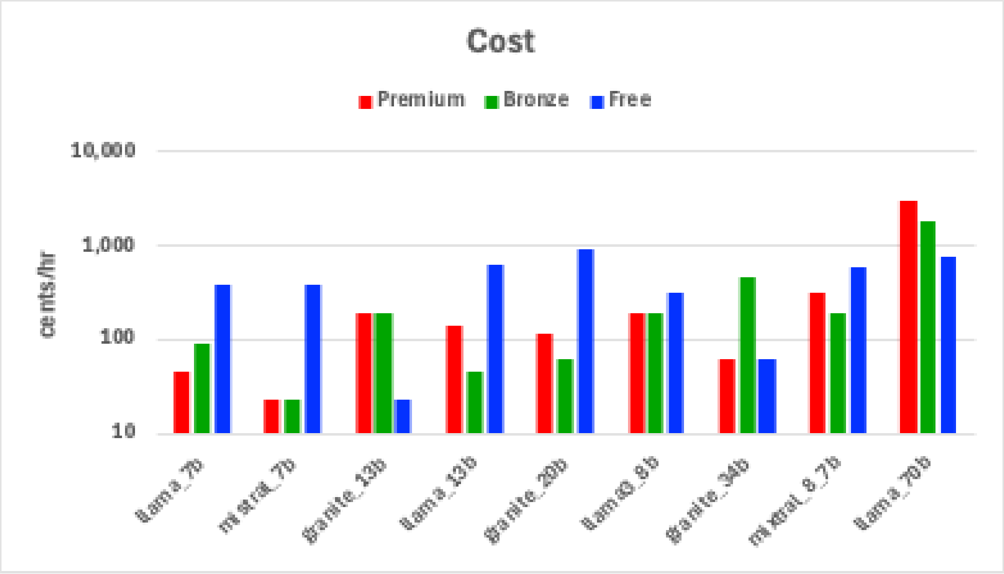


Accelerator Change

	Premium	Bronze	Free
llama_7b			
mistral_7b		MI300X->G2	
granite_13b			MI300X->G2
llama_13b		MI300X->MI250	
granite_20b	MI300X->G2		
llama3_8b			
granite_34b			
mixtral_8_7b			
llama_70b			

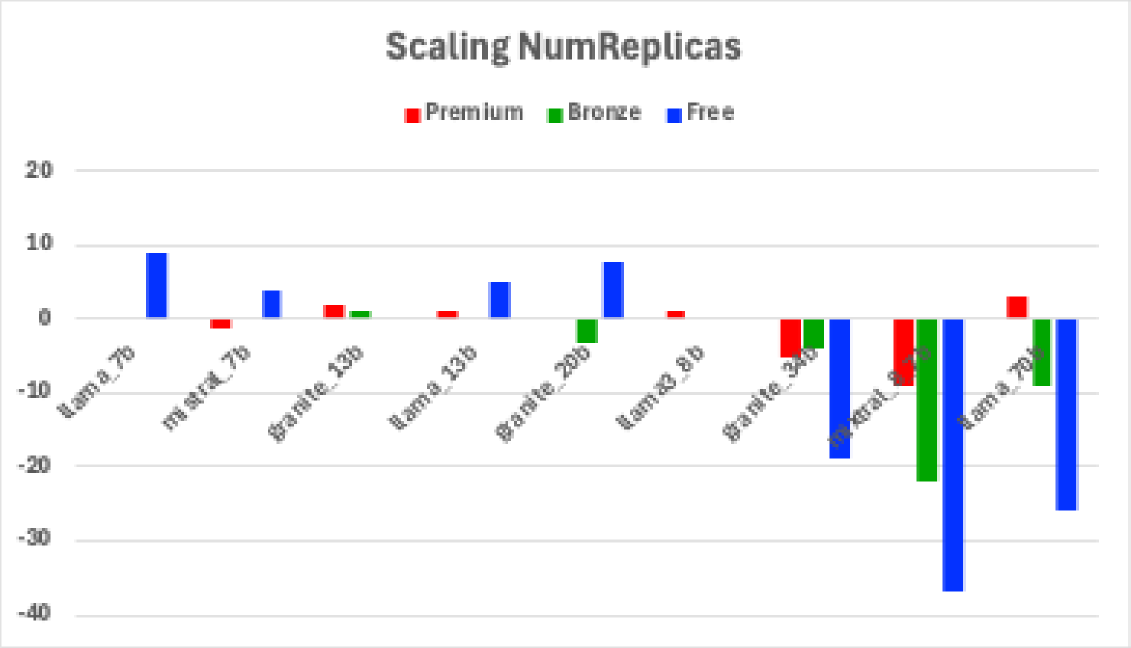


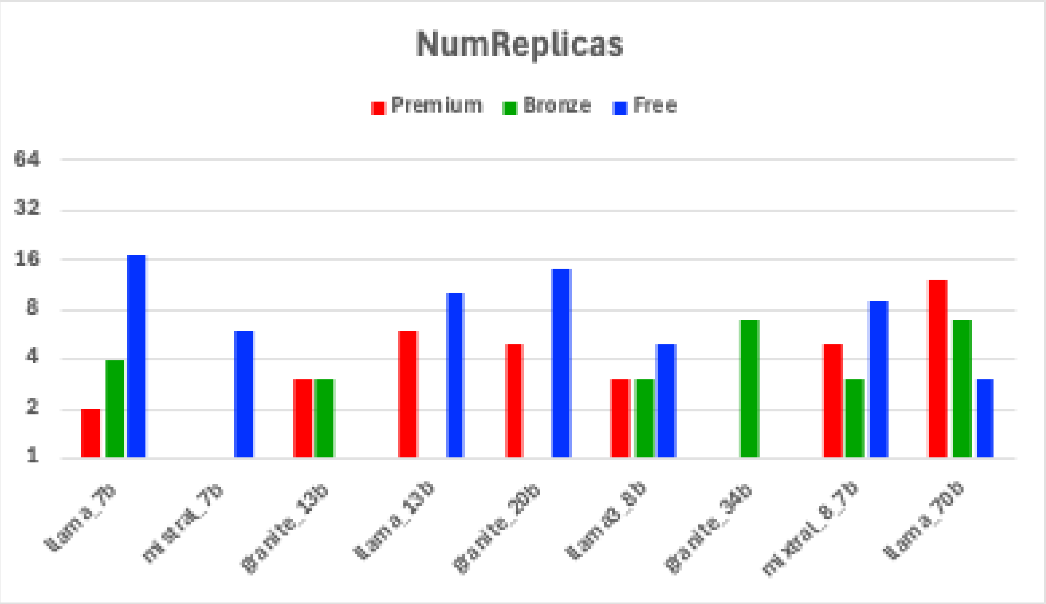
TotalCost
11,427.00

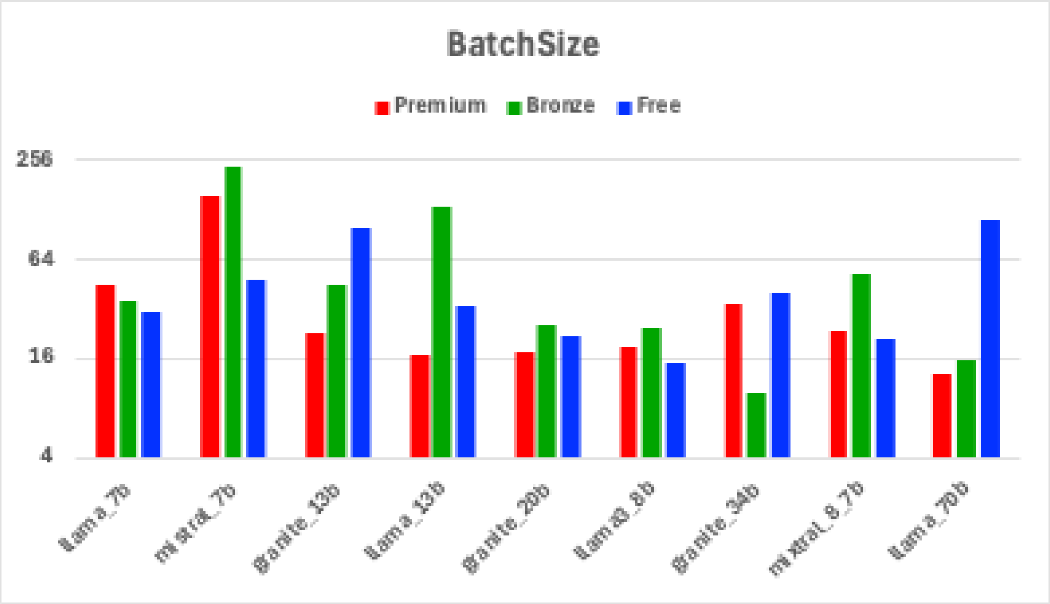


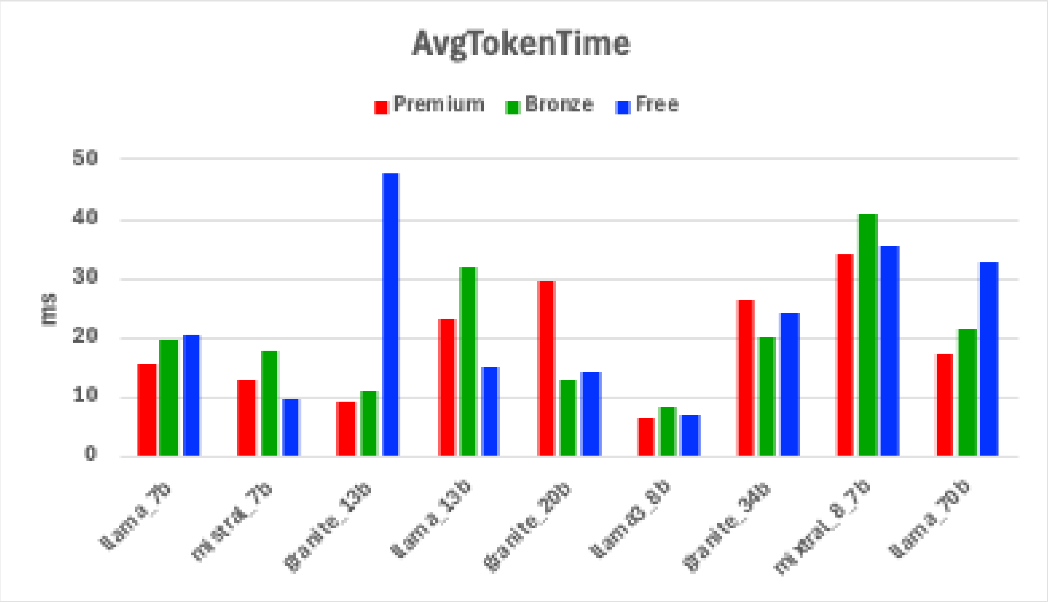
Accelerator Change			
	Premium	Bronze	Free
llama_7b			
mistral_7b		MI300X->G2	
granite_13b			MI300X->G2
llama_13b		MI300X->MI250	
granite_20b	MI300X->G2		
llama3_8b			
granite_34b			
mixtral_8_7b			
llama_70b			

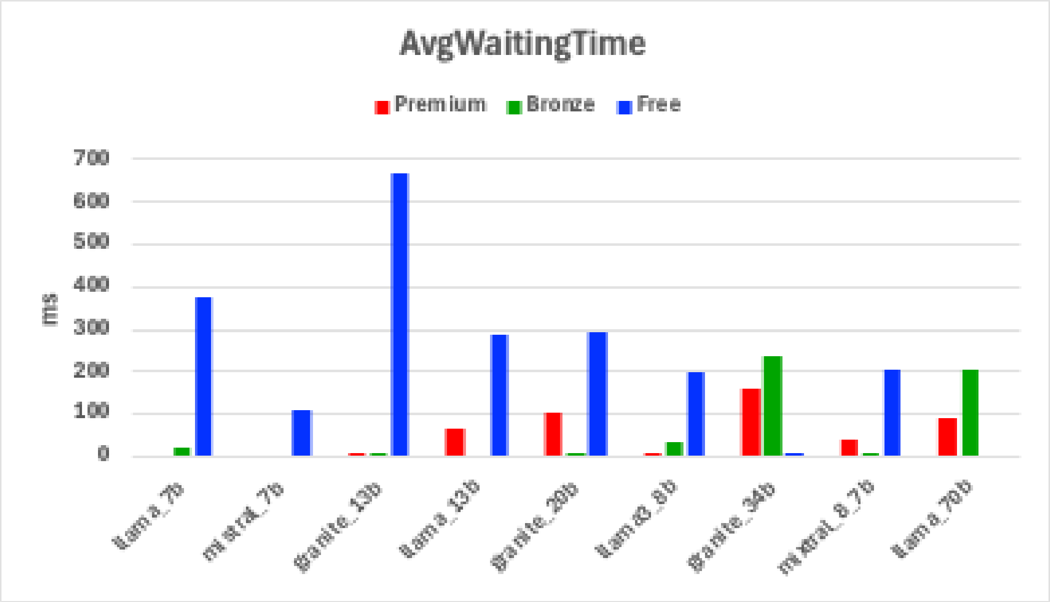
Scale			
	Premium	Bronze	Free
llama_7b	0	0	9
mistral_7b	-1		4
granite_13b	2	1	
llama_13b	1		5
granite_20b		-3	8
llama3_8b	1	0	0
granite_34b	-5	-4	-19
mixtral_8_7b	-9	-22	-37
llama_70b	3	-9	-26

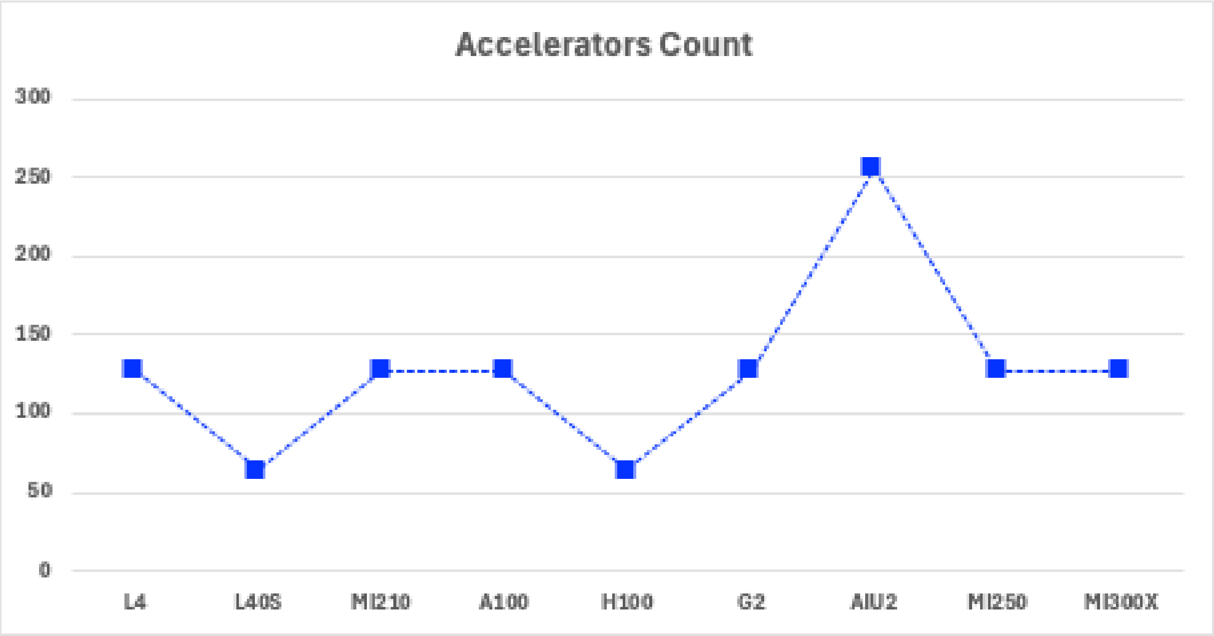



NumReplicas
Premium
Bronze
Free
64
32
16
8
4
2
1
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b


BatchSize
Premium
Bronze
Free
256
64
16
4
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

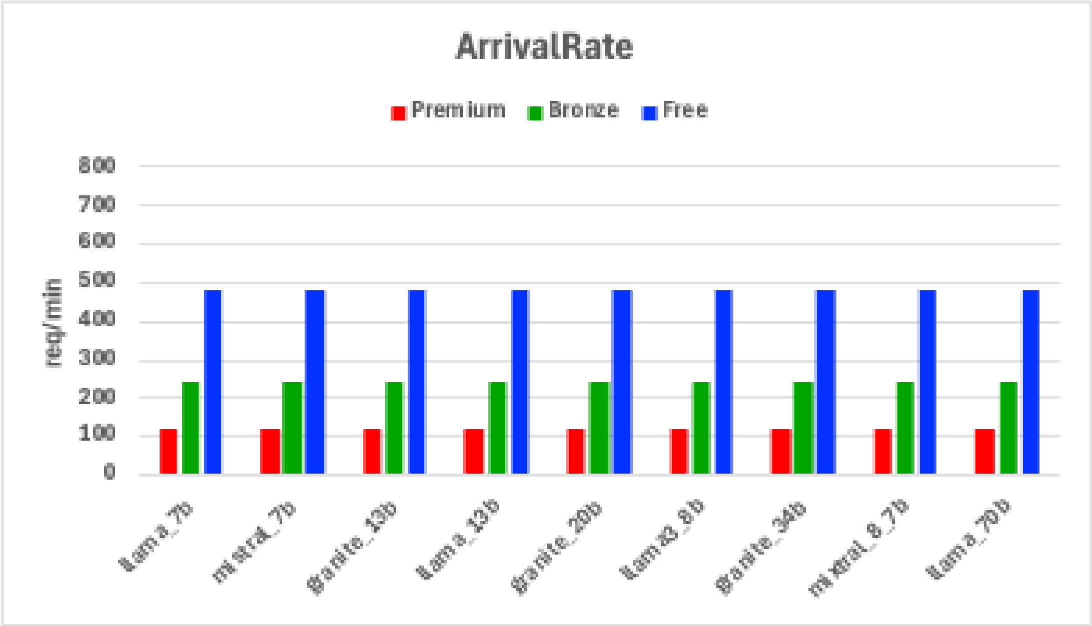

AvgTokenTime
Premium
Bronze
Free
50
40
30
20
10
0
ms
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

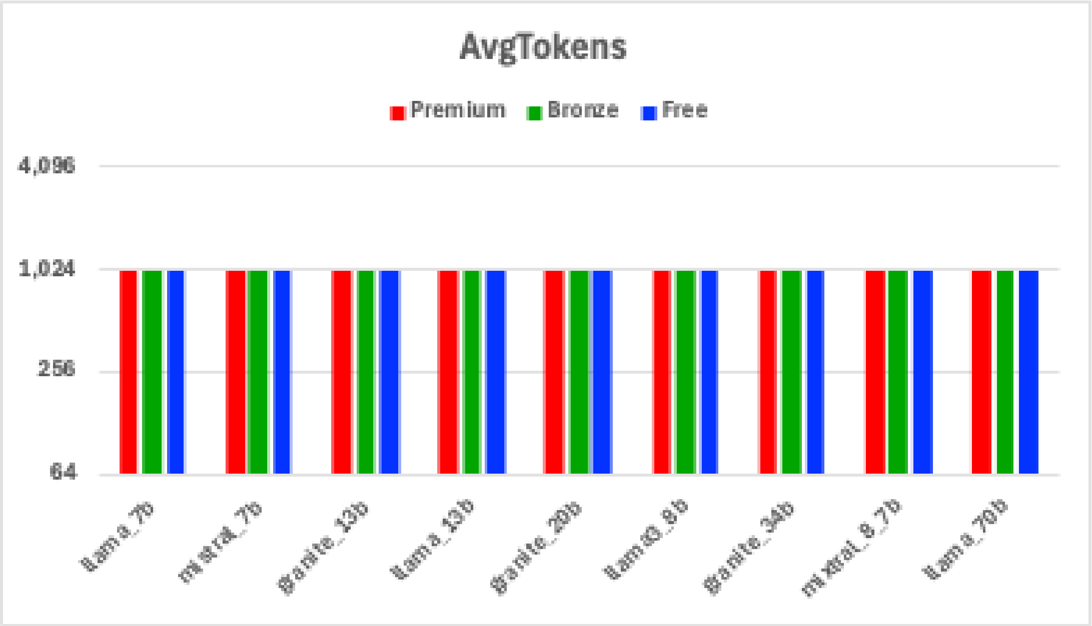

AvgWaitingTime
Premium
Bronze
Free
700
600
500
400
300
200
100
0
ms
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b



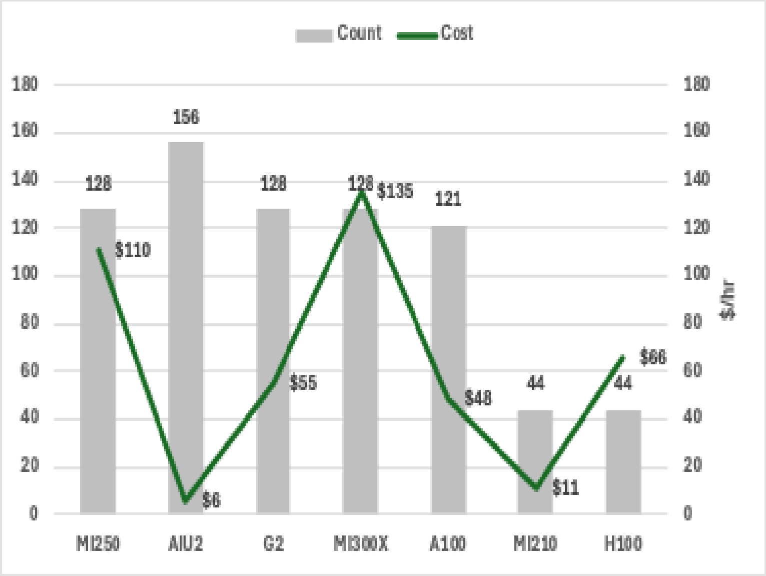
Limited accelerators

- cluster deployment
- greedy optimization

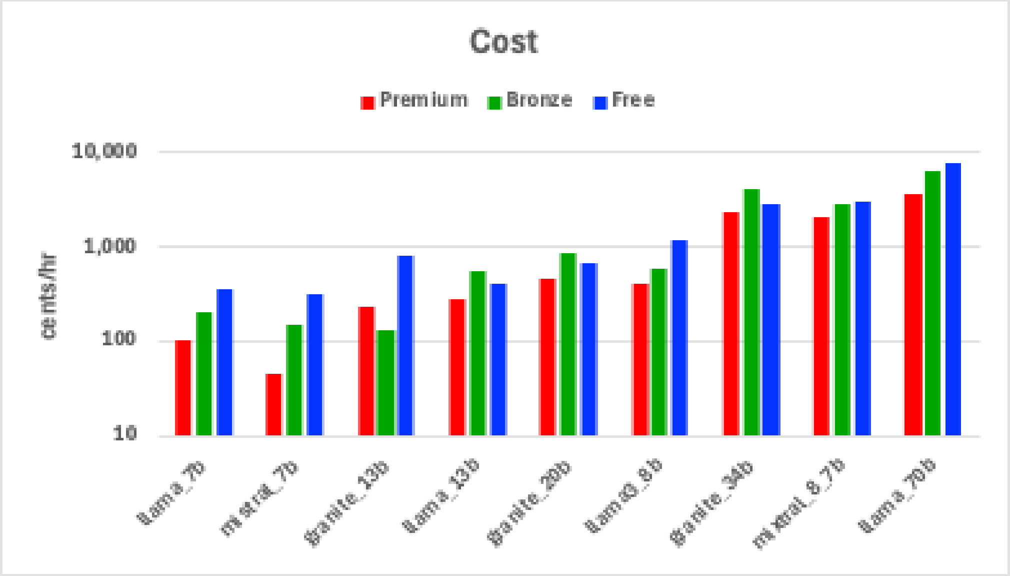


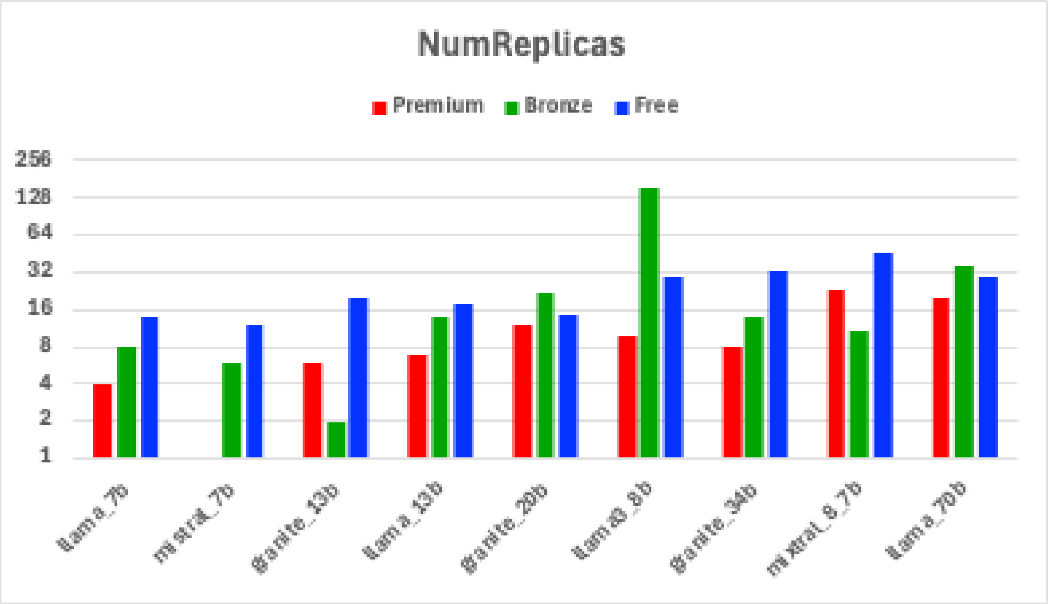


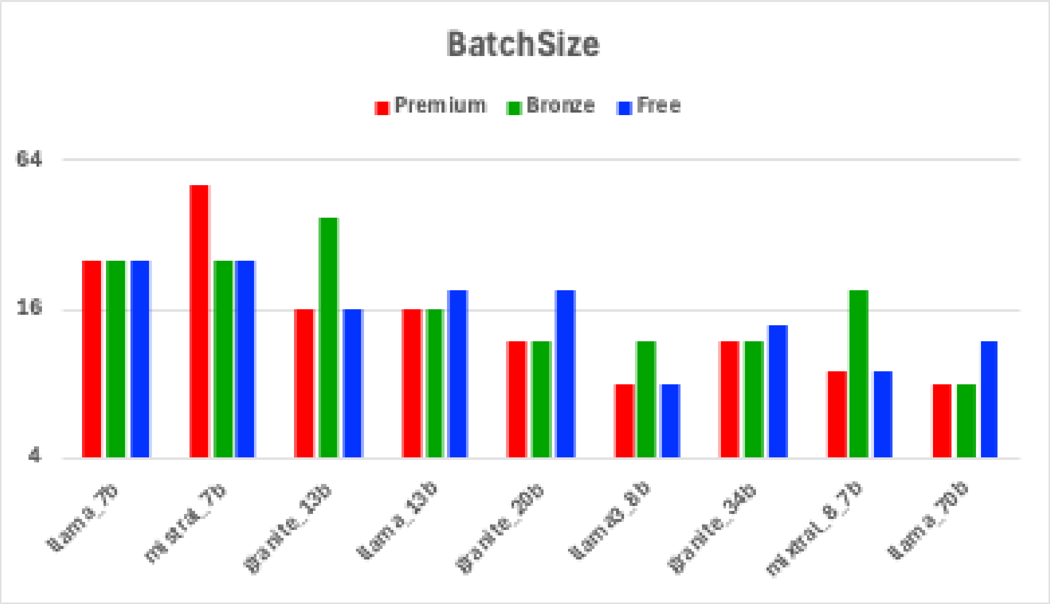
Accelerator			
	Premium	Bronze	Free
llama_7b	MI210	MI210	MI210
mistral_7b	MI250	MI210	MI210
granite_13b	A100	MI300X	A100
llama_13b	A100	A100	G2
granite_20b	A100	A100	MI250
llama3_8b	A100	AIU2	A100
granite_34b	2xH100	2xH100	2xG2
mixtral_8_7b	2xG2	2xMI300X	MI300X
llama_70b	2xMI250	2xMI250	2xMI300X

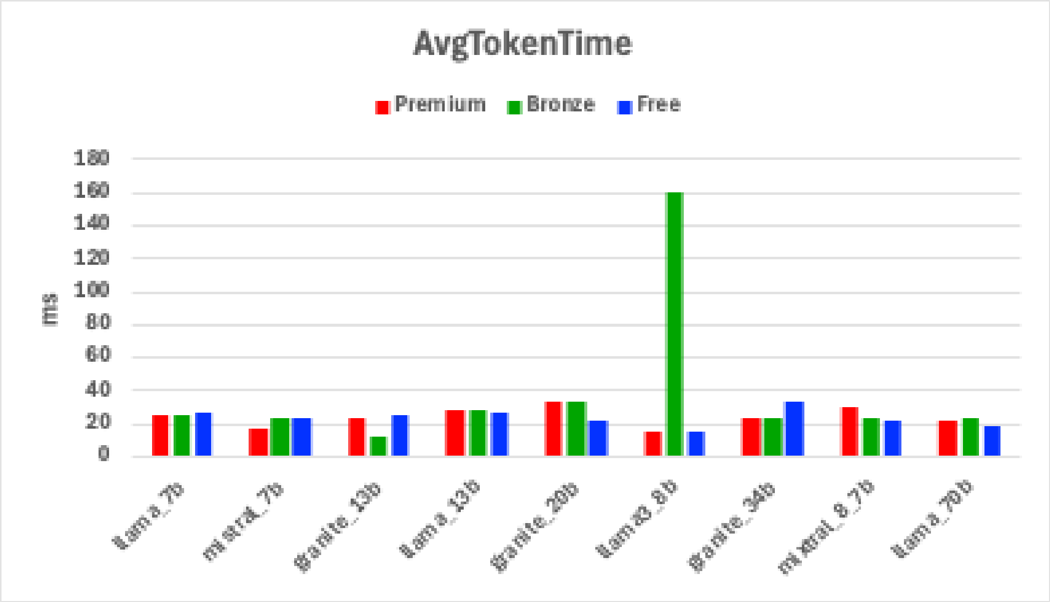


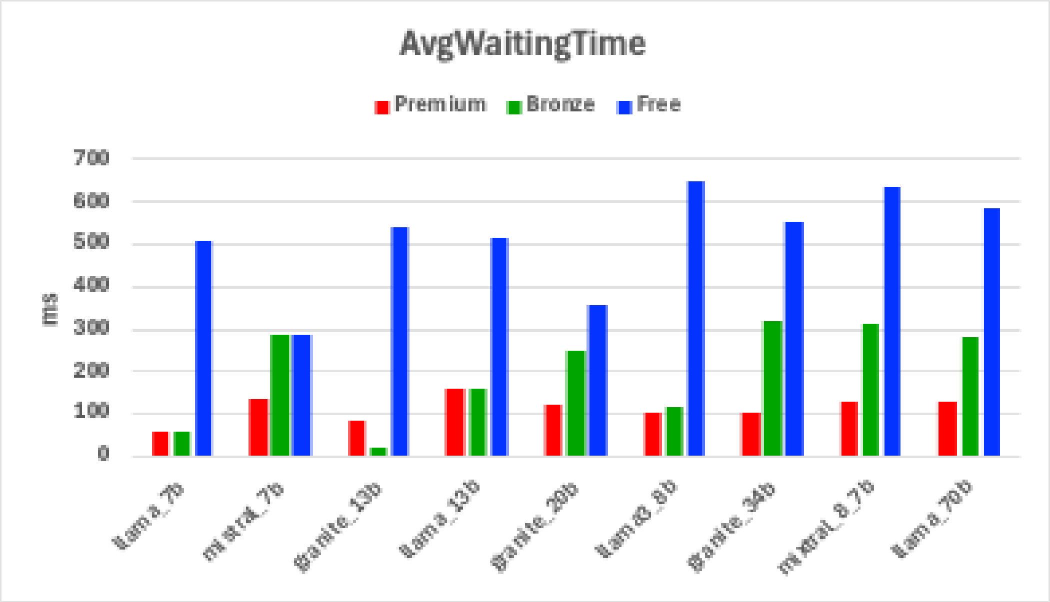
TotalCost
43,203.00

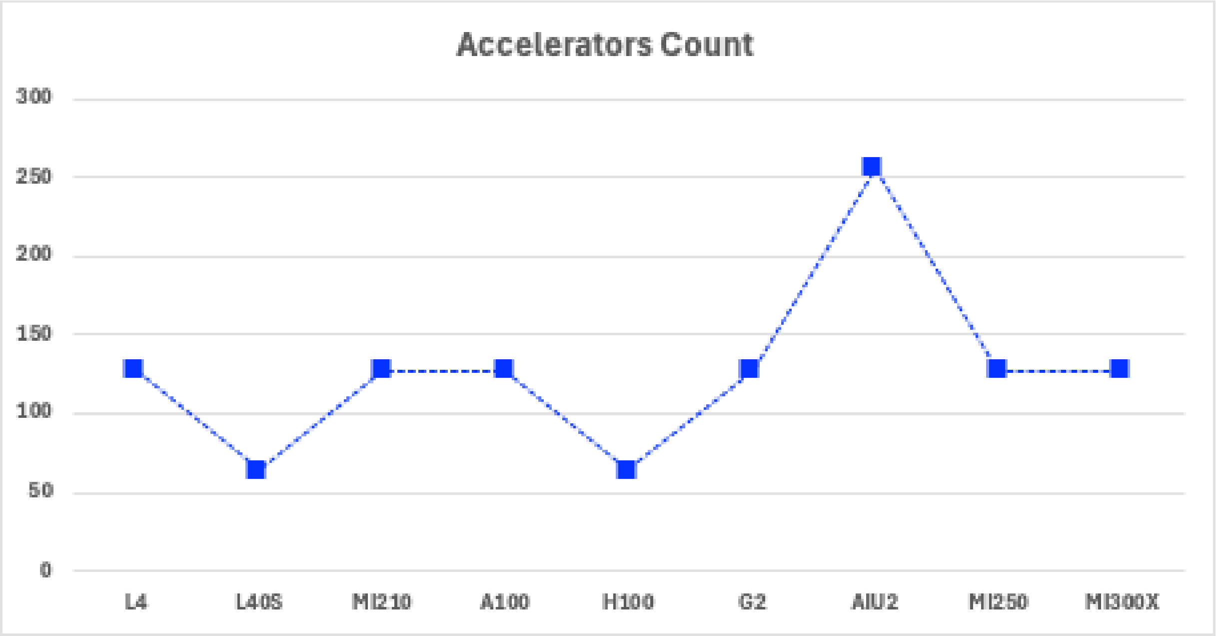



NumReplicas
Premium
Bronze
Free
256
128
64
32
16
8
4
2
1
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b


BatchSize
Premium
Bronze
Free
64
16
4
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

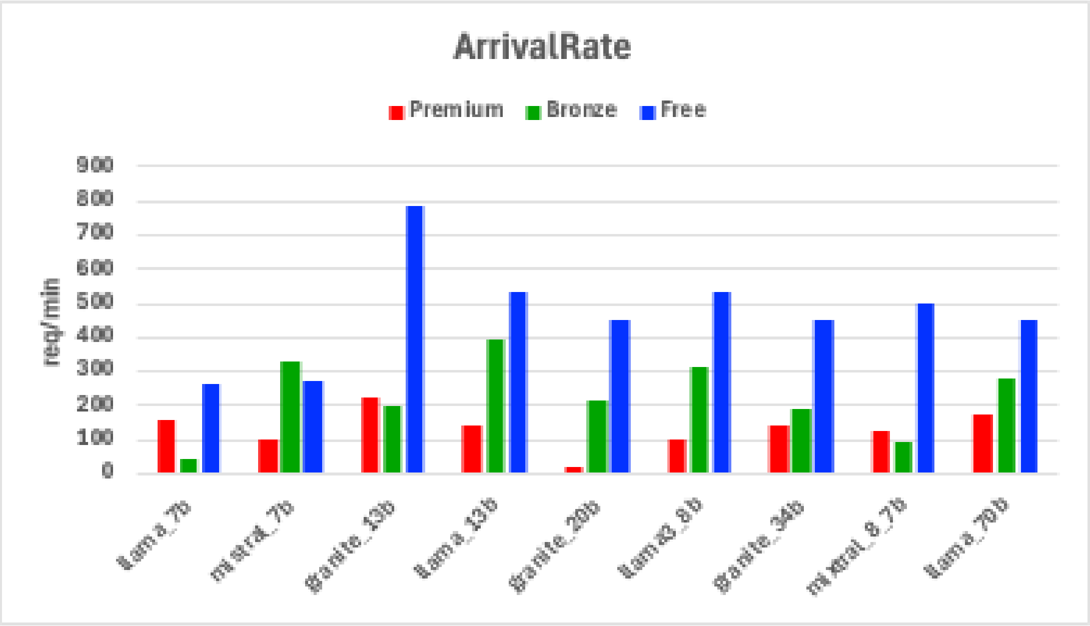

AvgTokenTime
Premium
Bronze
Free
ms
180
160
140
120
100
80
60
40
20
0
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b

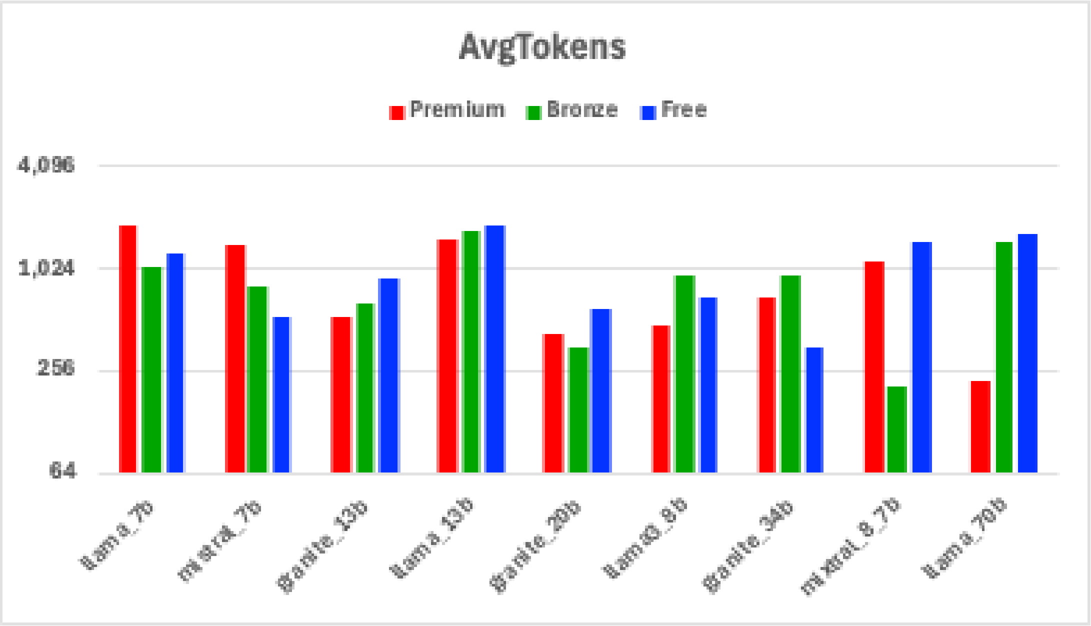

AvgWaitingTime
Premium
Bronze
Free
ms
700
600
500
400
300
200
100
0
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b



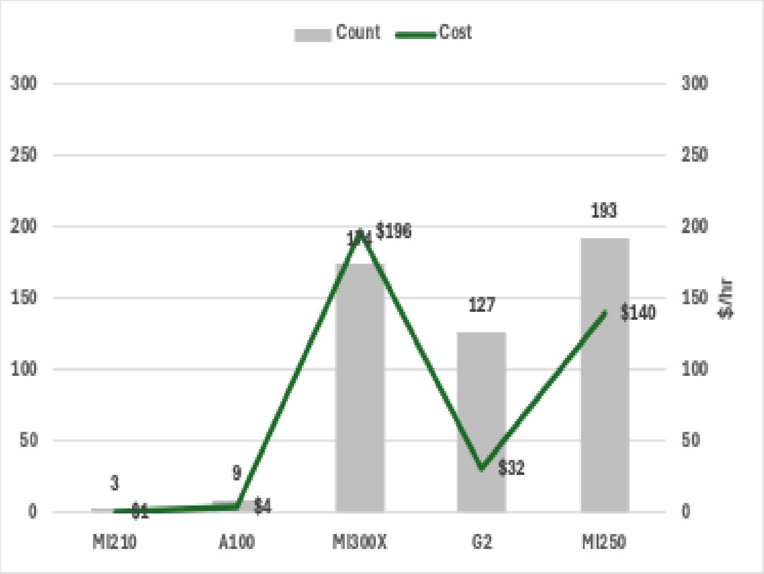
Limited accelerators - Dynamic

- change in request rates
- change in request lengths
- change/scale accelerators
- minimize change in accelerators and cost

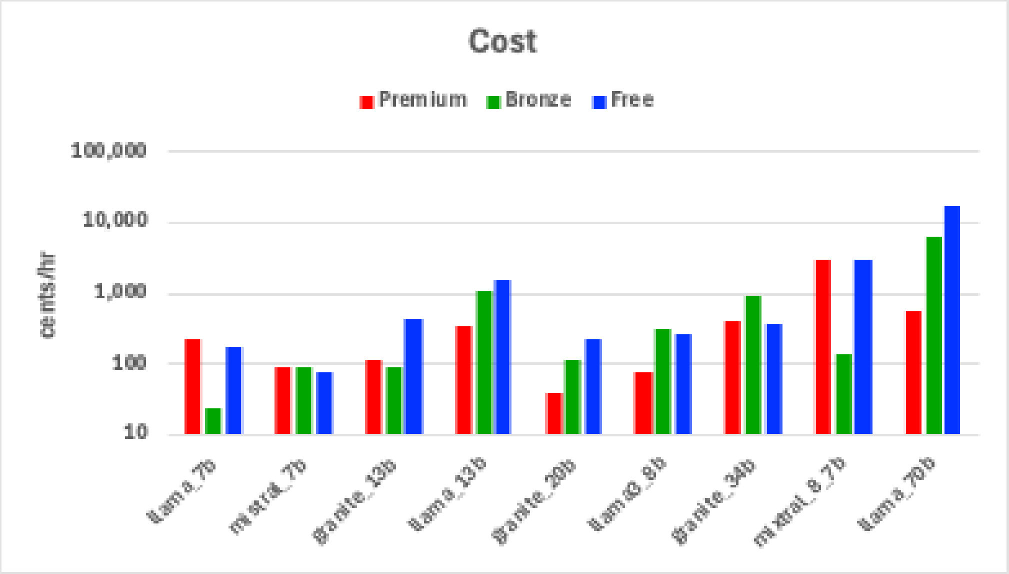




Accelerator Change			
	Premium	Bronze	Free
llama_7b	MI210->MI250	MI210->G2	MI210->MI250
mistral_7b		MI210->MI250	
granite_13b		MI300X->MI250	A100->G2
llama_13b	A100->G2	A100->MI250	
granite_20b			
llama3_8b		AIU2->MI250	A100->G2
granite_34b	2xH100->MI250	2xH100->MI250	
mixtral_8_7b	2xG2->2xMI250	2xMI300X->G2	
llama_70b			

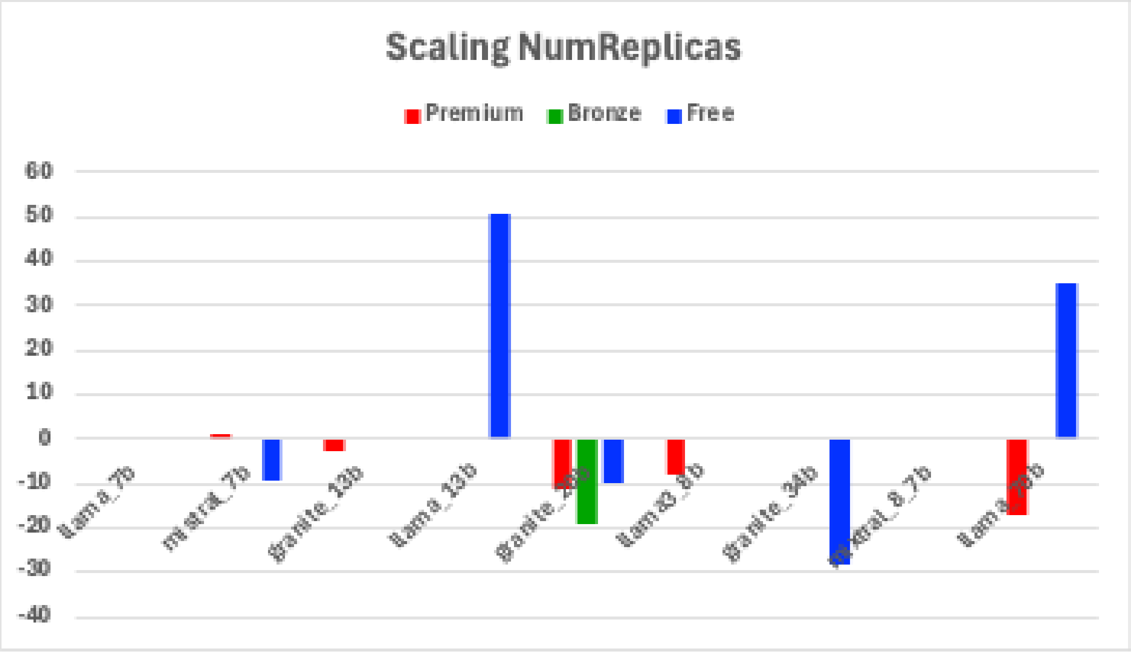


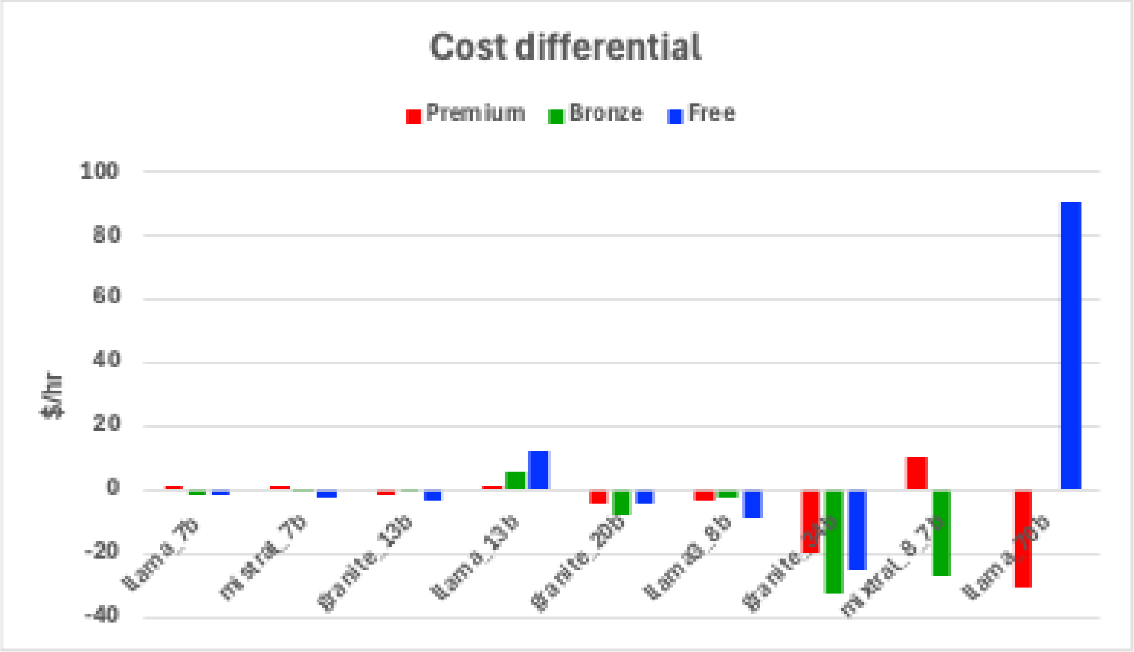
TotalCost
37,272.00

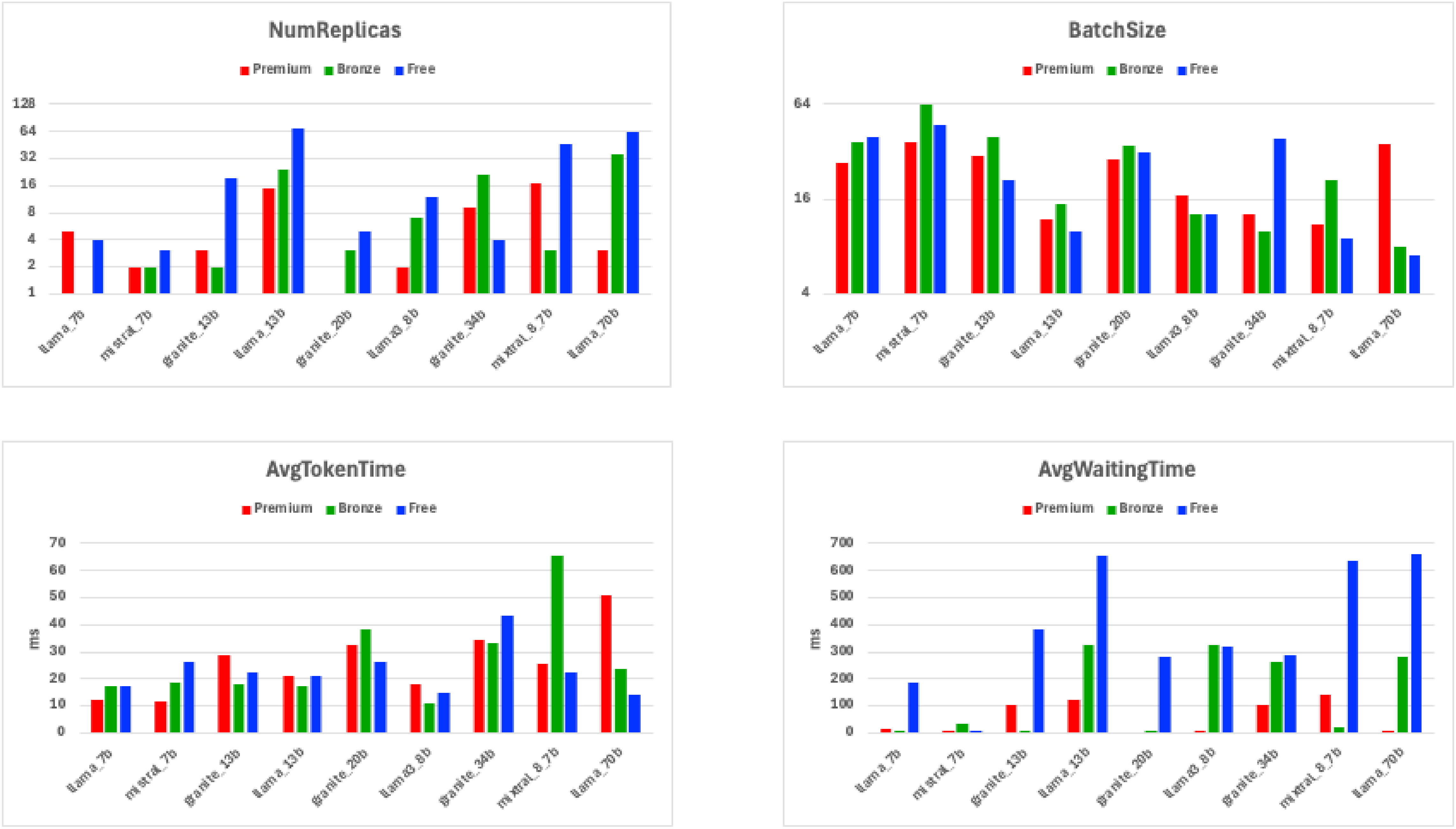


Accelerator Change			
	Premium	Bronze	Free
llama_7b	MI210->MI250	MI210->G2	MI210->MI250
mistral_7b		MI210->MI250	
granite_13b		MI300X->MI250	A100->G2
llama_13b	A100->G2	A100->MI250	
granite_20b			
llama3_8b		AIU2->MI250	A100->G2
granite_34b	2xH100->MI250	2xH100->MI250	
mixtral_8_7b	2xG2->2xMI250	2xMI300X->G2	
llama_70b			

Scale			
	Premium	Bronze	Free
llama_7b			
mistral_7b	1		-9
granite_13b	-3		
llama_13b			51
granite_20b	-11	-19	-10
llama3_8b	-8		
granite_34b			-28
mixtral_8_7b			0
llama_70b	-17	0	35



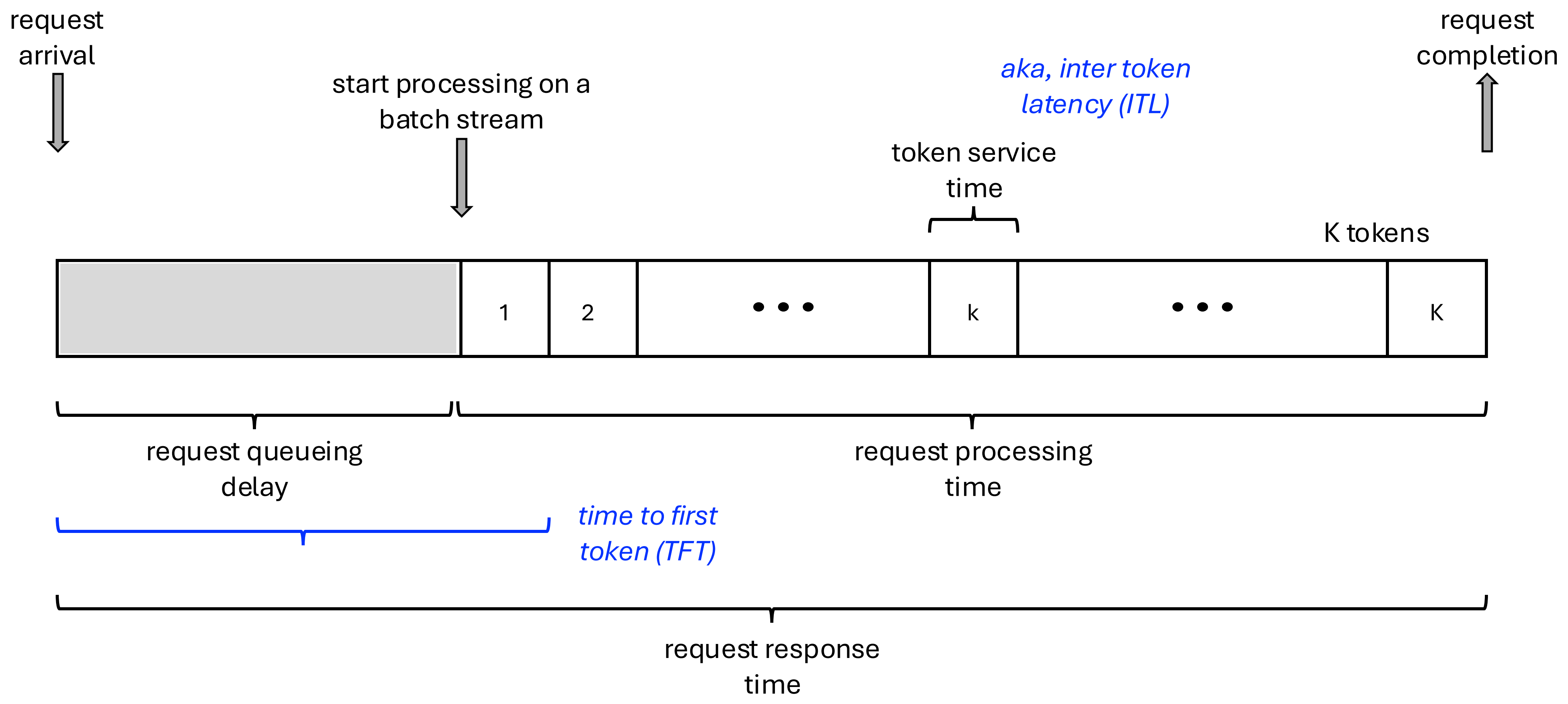




NumReplicas
Premium
Bronze
Free
128
64
32
16
8
4
2
1
llama_7b
mistral_7b
granite_13b
llama_13b
granite_20b
llama3_8b
granite_34b
mixtral_8_7b
llama_70b
BatchSize
64
16
4
AvgTokenTime
ms
70
60
50
40
30
20
10
0
AvgWaitingTime
700
600
500
400
300
200
100

Backup

Request timing & definitions



amortize delay over all tokens

$$token\ delay = \frac{request\ response\ time}{K}$$

$$token\ delay\ \cong ITL + \frac{TFT}{K}$$

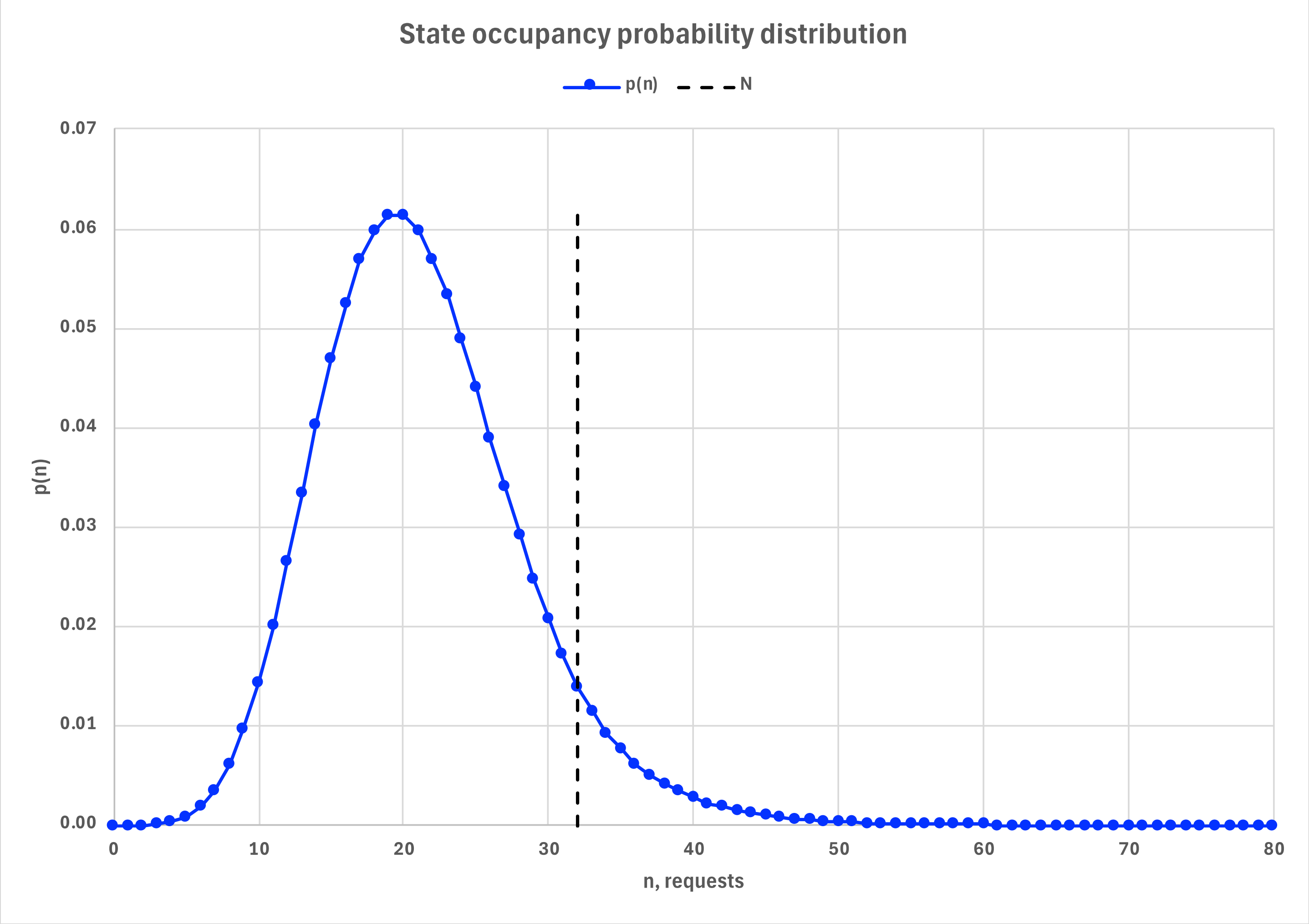
Objectives

- Optimize operational cost/performance/power
  - assignment of accelerator types
  - model replicas
  - model batch decision
  - meet SLOs
  - min cost and power
- Capacity planning and/or model placement & accelerator allocation

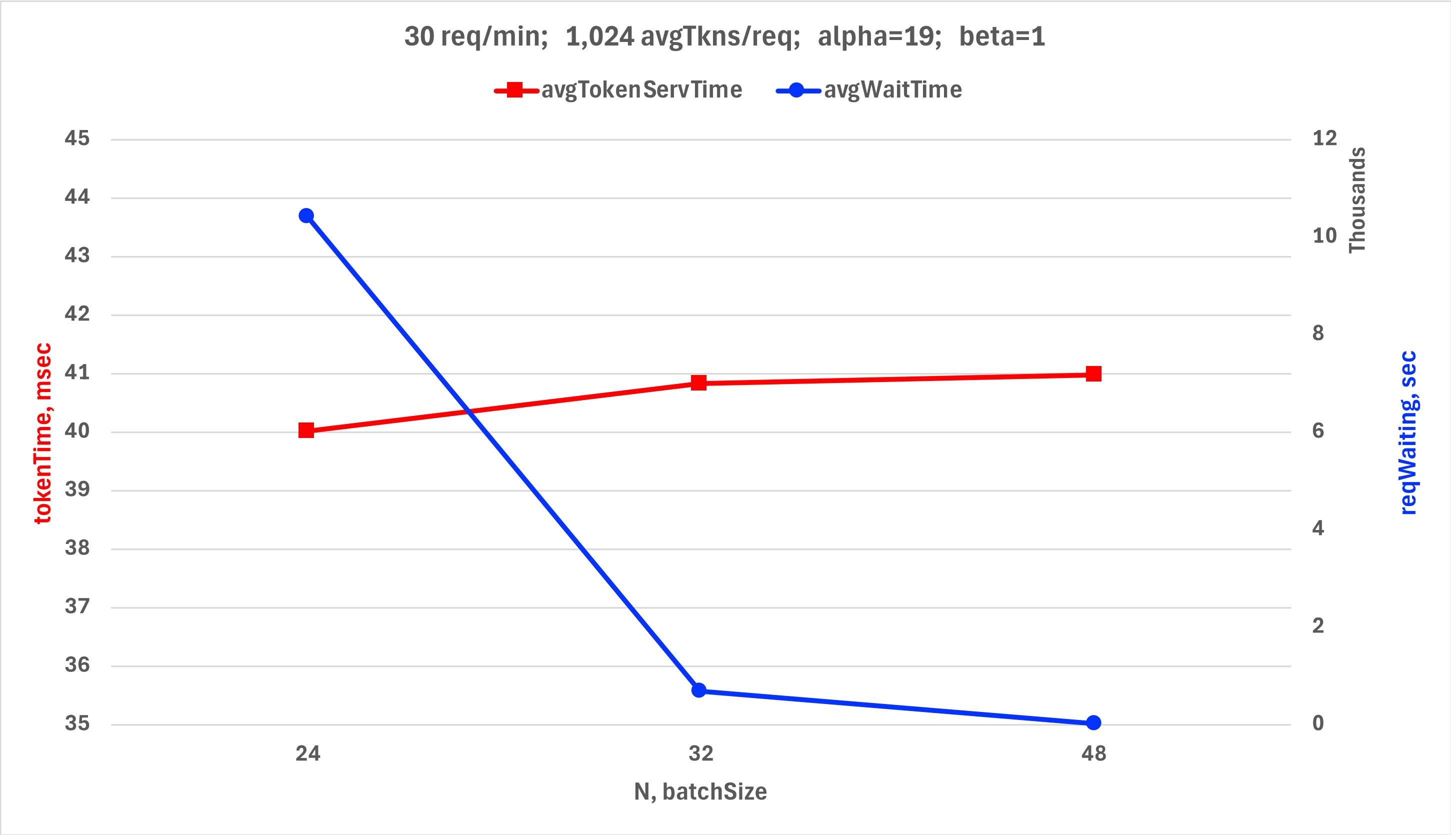
- Inference model optimization
- Improve model accuracy, efficiency, compute/memory requirements, …
- Explore mechanisms for request queueing, balancing, …
- Model sharing, adapters, multiplexing, …
- GPU slicing mechanisms, DRA, migration, …

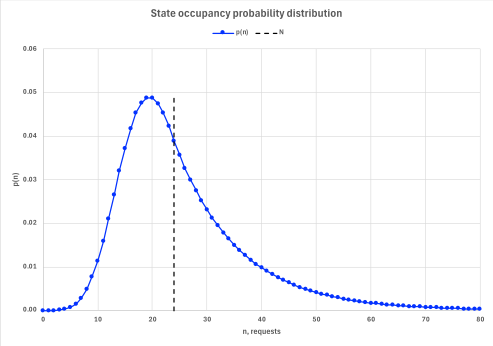
sample case

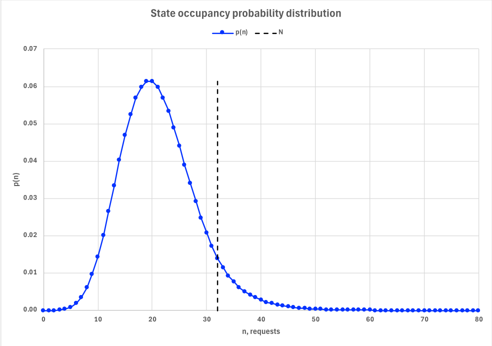
requestRate	30	/min	p(0)	5.7982E-07		avgNumSystem	21.24
lambda	0.0005	/msec	maxN	200		avgNumServer	20.90
K	1024		pFull	2.0382E-17		avgNumQueue	0.34
N	32		effecTput	30.00000	req/m	utilization	0.653
alpha	19			0.00050	/msec	avgRespTime	42,481
beta	1		P(N)	0.93803		avgSrvTime	41,808
lambda*K	0.512					avgWaitTime	674
						avgTokenServTime	40.8

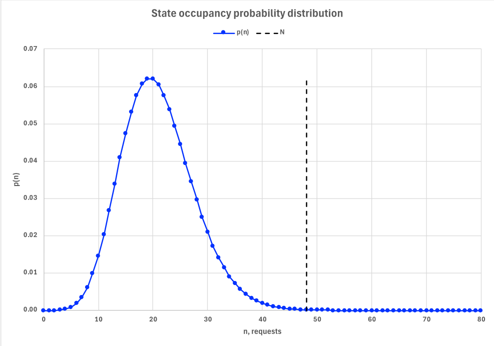


impact of batch size









ITL limiting

requestRate	35.200	/min	p(0)	1.0542E-08		avgNumSystem	30.23
lambda	0.00059	/msec	maxN	200		avgNumServer	30.04
K	1024		pFull	1.4442E-14		avgNumQueue	0.20
N	48		effecTput	35.20031	req/m	utilization	0.626
alpha	19			0.00059	/msec	avgRespTime	51,533
beta	1		P(N)	0.96847		avgSrvTime	51,200
lambda*K	0.600752					avgWaitTime	333
						avgTokenServTime	50.0

Find

maximum request rate

s.t.

avgTokenServTime <= target

target = 50 msec
requestRate = 35.2 req/min