Return on Investment and Compute Infrastructure when value of result is not quantified

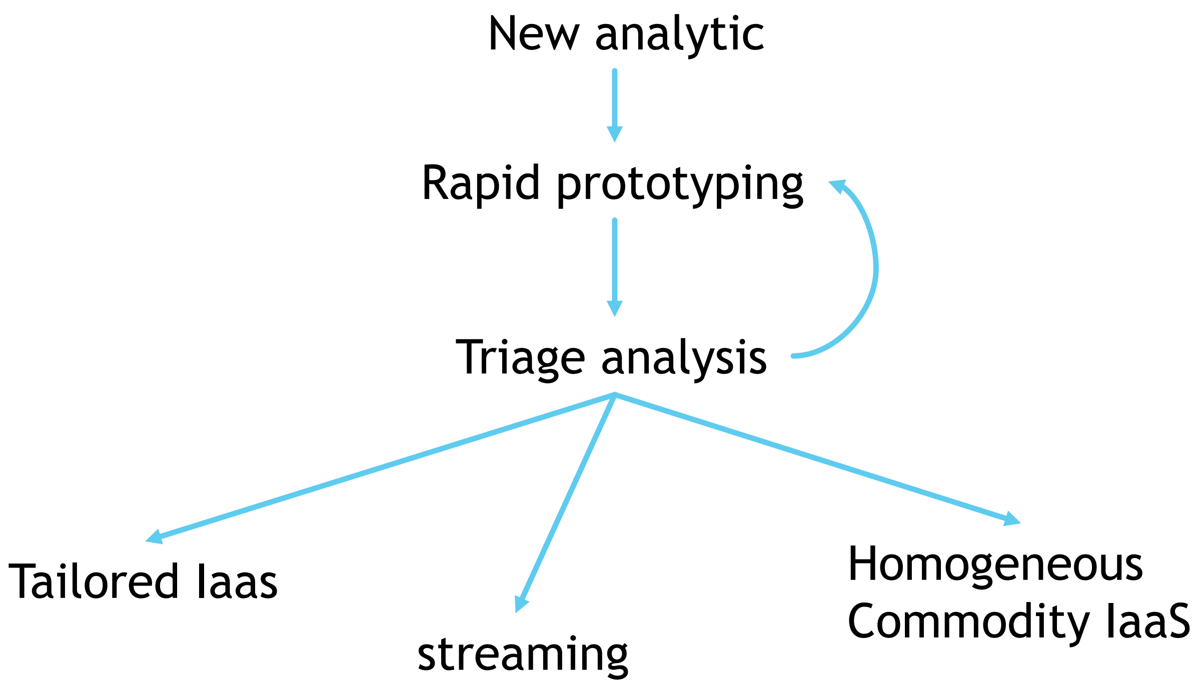
Ben Payne

20160506

Rapid Prototyping platforms

- Your desktop - Microsoft Excel, etc
- AWS
- Google Compute Engine
- Jupyter notebooks

Triage process design



New analytic

Rapid prototyping

Triage analysis

Tailored Iaas

streaming

Homogeneous Commodity IaaS

How to evaluate merit?

Two approaches

- Single analytic, compare RoI for multiple platforms over lifespan (ie 5 years)
- Many analytics, compare RoI for a single investment period (ie fiscal year)

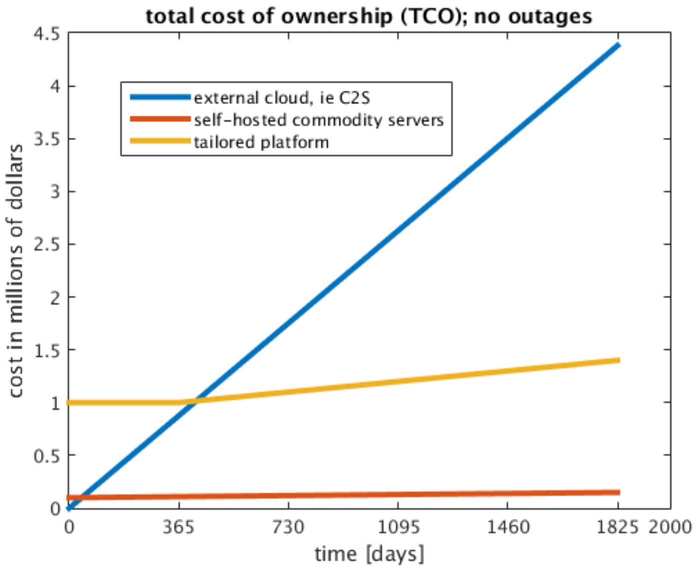
Comparison of platforms: “pay-as-you-go” vs “commodity” vs “tailored”

- AWS: assume zero capital cost and zero operations and maintenance cost. Fiscal cost is in number of hours used; I assume a rate of $100/hour. Time-to-market is the advantage; assume 1 day for acquisition and creating the analytic. Time-to-solution is assumed to be 100 minutes

- self-hosted commodity platform: capital cost is $100,000 (one-time investment), and the operations and maintenance is $10,000 per year (an on-going cost). Time-to-market is 5 days; time-to-solution is 60 minutes.

- tailored architecture is costly — $1,000,000 for capital, and $100,000 per year for O&M. Advantage is a tiny time-to-solution, 1 minute, but the time-to-market is 1 year. Time-to-market includes NRE and acquisition.

Total cost of ownership



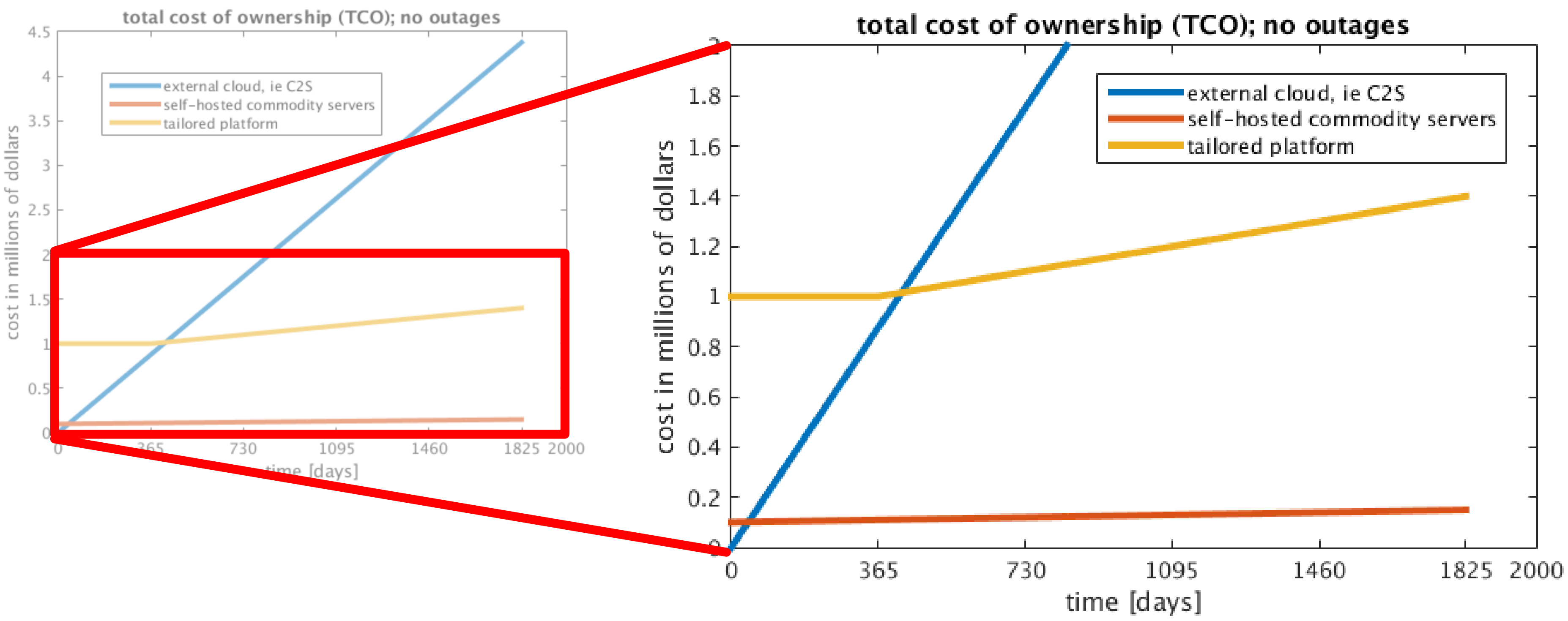
(Assume a single analytic)

(Lower is better)

Observations:

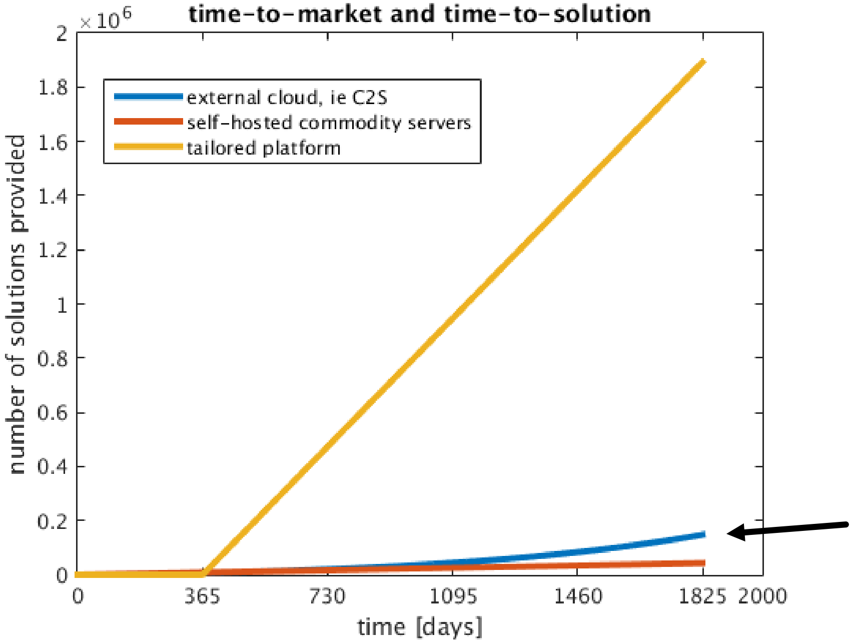
- No capital investment means initial cost is lowest
- Tailored architecture costs more than commodity
- Paying by the hour gets expensive

Total cost of ownership



For the current parameters, pay-by-the-hour is not as cost effective;
you do get access to compute faster

Time-to-market and time-to-solution



(due to Moore’s law)

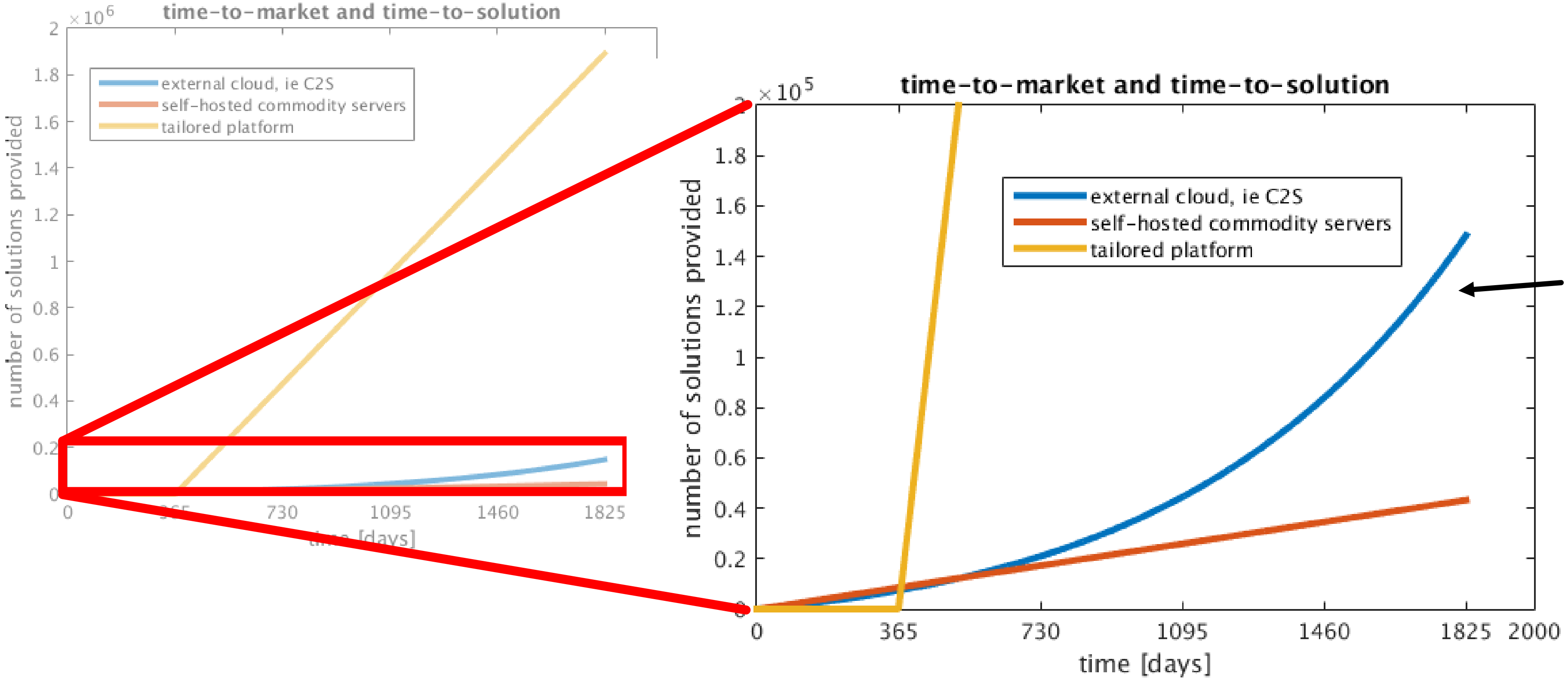
- The tailored architecture provides more solutions over lifespan

(Higher is better)

System availability:

- External cloud: 99.9%
- Self-hosted cloud: 99%
- Tailored architecture: 90%

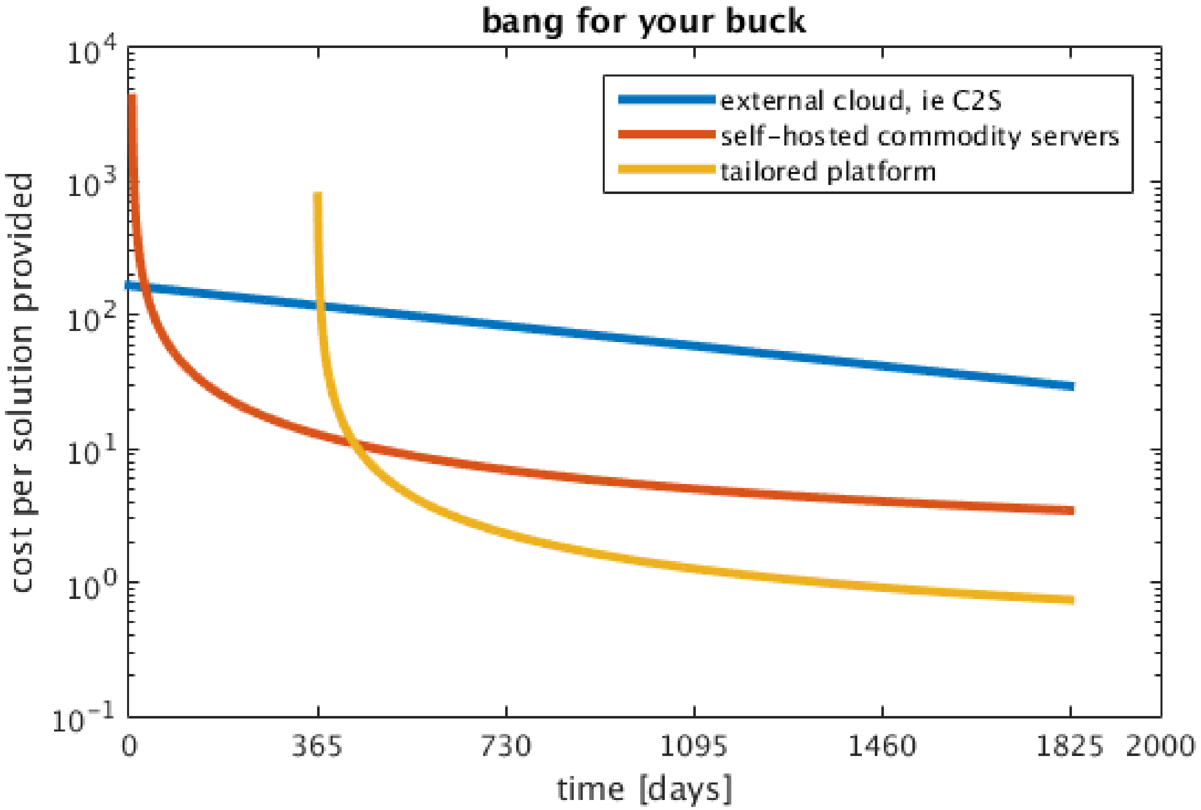
Time-to-market and time-to-solution



(due to Moore’s law)

For the current parameters, ratio of time spent in NRE and acquisition to time-to-solution significantly impacts throughput

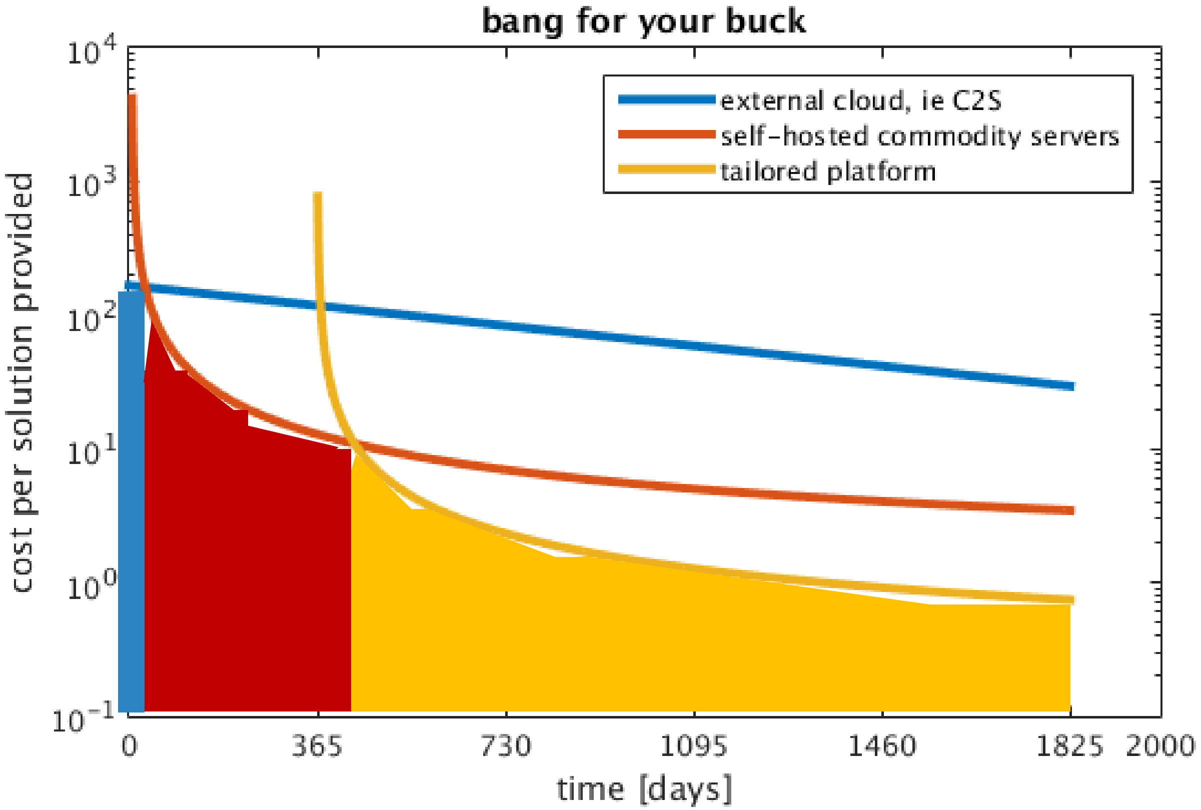
Bang-for-your-buck



(Lower is better)

Over the lifespan of the system, tailored IaaS is better ROI

Bang-for-your-buck



(Lower is better)

“best” depends on durability of analytic

Multi-tenancy on existing infrastructure: how to invest in next iteration?

- Suppose existing platform environment is homogenous
- Suppose you will get money to spend on compute for next fiscal period

- **Should that money be invested on more of the same,**
  or spend the money on a novel architecture?

- Assumption: known tasks in the time period
- Assumption: known amount of work in the period

Input Parameters

- number of existing compute units for architecture A: 5

- capital cost per compute unit for architecture A: $20
- capital cost per compute unit for architecture B: $50
- O&M cost per compute unit for architecture A: $5
- O&M cost per compute unit for architecture B: $8
- Money to invest in next fiscal period: $100

- For each analytic, CPU hours for architecture A: {40, 100, 10, 5, 2}
- For each analytic, CPU hours for architecture B: {30, 10, 8, 4, 1}

Result of analysis

```
current_tts =
   31.4000
tts_homogeneous =
   19.6250
min_tts_heterogeneous =
   21.4000
Best analytic distribution, platform A =
    40     0    10     5     2
best analytic distribution, platform B =
     0    10     0     0     0
```

Result: maintaining homogeneous architecture yields better throughput

```
current_tts =
   31.4000
tts_homogeneous =
   19.6250
min_tts_heterogeneous =
   ~~21.4000~~
Best analytic distribution, platform A =
    {40     0    10     5     2}
best analytic distribution, platform B =
     {0    10     0     0     0}
```

- For these parameters, change of architecture doesn’t outweigh the cost of the acquisition + O&M
- Even though architecture B is always faster for every analytic, the overall mission throughput is lower for the amount of money spent. Therefore, **stick with homogeneous architecture**

Result: different initial conditions yield opposite outcome

capital cost per compute unit for architecture A: $20

```
current_tts =
   31.4000
tts_homogeneous =
   19.6250
min_tts_heterogeneous =
   21.4000
Best analytic distribution, platform A =
    {40     0     10     5     2}
best analytic distribution, platform B =
     {0    10     0     0     0}
```

capital cost per compute unit for architecture A: ***$30***

```
current_tts =
   31.4000
tts_homogeneous =
   22.4286
min_tts_heterogeneous =
   21.4000
Best analytic distribution, platform A =
    {40     0     10     5     2}
Best analytic distribution, platform B =
     {0    10     0     0     0}
```

This methodology applies to any situation

- I have a homogenous commodity IaaS and want to know whether to buy invest in a novel architecture
- I have tailored IaaS and want to know whether to invest in commodity IaaS
- I have both tailored and commodity IaaS and want to know what future distribution of funds maximizes throughtput
- Any platform, and set of analytics

- Need capital cost, O&M cost, and job runtimes per platform
- Need existing infrastructure, job list, and amount of money to be invested
- Output: which platform acquisition maximizes mission throughput

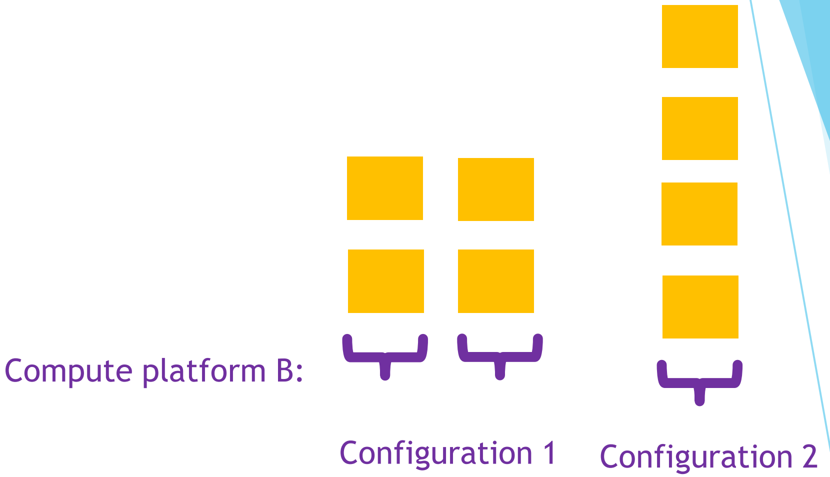
What I’m looking for

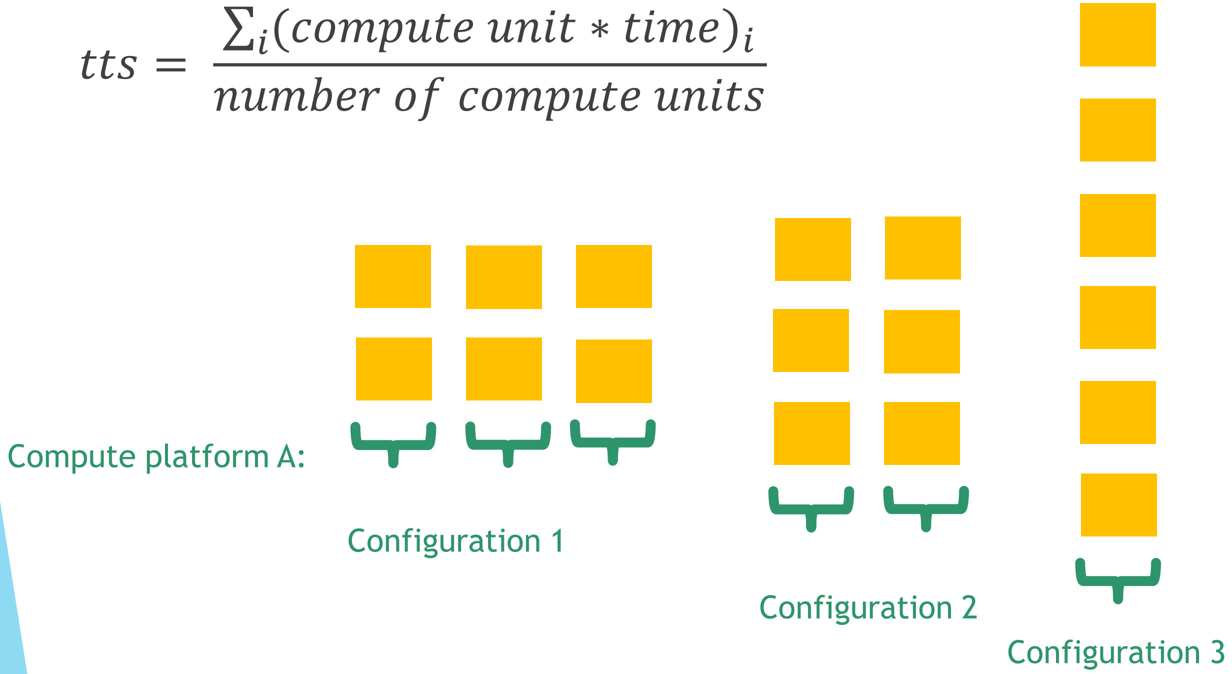
- **Measurements** from current platforms (time-to-solution for every analytic) over some time period
- Candidate platforms for acquisition
- Cost of system, number of racks or nodes, both capital and O&M

BACKUP

Attack the assumptions: space-time trade-off

$$tts = \frac{\sum_i (compute\ unit * time)_i}{number\ of\ compute\ units}$$





- Space-time trade-off of real applications is not linear; see Amdahl’s law
- Scaling depends on how much data is exchanged; synchronous communication

Attack the assumptions: space-time trade-off

- Each job has a scaling curve
- A scheduler searches for the layout of jobs on N compute resources which minimizes time-to-solution
- Scheduler output would yields the time-to-solution per platform

Attack the assumptions:
known tasks; known amount of work

- Projection is needed for the fiscal period being modeled
- We have data for what was done in the previous period
- Extrapolate historical data using a scaling factor

Getting faster doesn’t matter; the problem is constrained by meatspace

- Investigating different platform architectures wrt time-to-solution

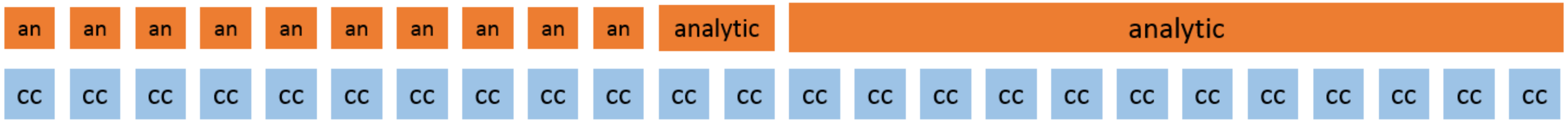
- Providing users more data is not helpful
- Providing users with data faster is not helpful

- Replace “speed-up” with “lower cost to solution”

Stick with homogeneous IaaS, or augment with tailored IaaS?

Goal: balance consumers and resources (avoid both idle and inability to do mission)

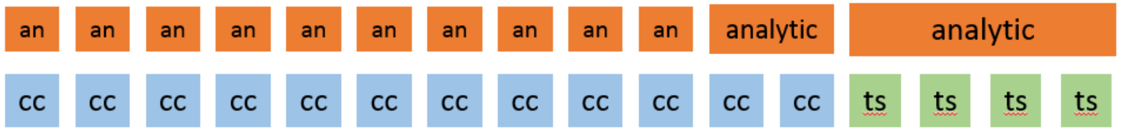
Homogeneous environment:



an an an an an an an an an an analytic analytic (compute consumers)

cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc (compute resources)

Heterogeneous environment:



an an an an an an an an an an analytic analytic

cc cc cc cc cc cc cc cc cc cc cc cc ts ts ts ts

These alternatives could provide equivalent mission throughput;
→ What about $?