EDA Workloads & Public Clouds

Can we really save money in the public cloud?

Mitch Richling

June 25, 2017

https://www.mitchr.me/SS/res/papers/celug2017MJR.pdf

TEXAS INSTRUMENTS

Outline

- Vocabulary
- Cloud Observations: The HPC EDA Perspective
- Basic criteria for saving money with public cloud
- Criteria applied to Single server
- Criteria applied to HPC clouds (bursting)
- Applications

Vocabulary

HPC Compute Environments

- ~~Supercomputer~~
- ~~Compute Cluster~~
- ~~Compute Farm~~
- ~~Compute Ranch~~
- ~~Compute Grid~~
- ~~The Grid~~
- HPC Cloud

Vocabulary

Internal vs. External Clouds

- *Internal Cloud*
  A group of compute servers acquired via long term lease or direct purchase.
- *External Cloud*
  Cloud infrastructure services (compute servers) purchased from an external entity on very short term lease (1 hour to 1 week).
- *Hybrid Cloud*
  Simultaneously using both *Internal Cloud* and *External Cloud*.

Cloud Observations: The HPC EDA Perspective

- Cost
- Requirements Mismatch
- Old Software
- EDA Software License Costs

Observation 1: Public Cloud Infrastructure Is Expensive

From an EDA HPC perspective, public cloud prices can be shockingly high.

Observation 1: Public Cloud Infrastructure Is Expensive

Unnecessary hardware features for EDA HPC

- Fully redundant networking
- Extremely flexible and re-configurable networking
- Rack level network security segmentation and scanning
- Fully redundant power
- Long term secondary power generators

Observation 1: Public Cloud Infrastructure Is Expensive

HPC Software Pricing

Typical HPC software discounts can be in the 98% range. For example, a $1,500 enterprise OS license might be $25 for an HPC shop.

While public cloud vendors no doubt enjoy deep discounts on high volume purchases, we do not see the kinds of discounts we normally get in HPC reflected in the final product price.

Observation 1: Public Cloud Infrastructure Is Expensive

PROFIT!

Texas Instruments

Observation 2: Cloud Offering Mismatch with EDA

Public cloud offerings are a bit of a mismatch when it comes to EDA HPC requirements.

Observation 2: Cloud Offering Mismatch with EDA

Network Attached Storage

- Most clouds provide better support, and far lower prices, for software defined storage.
- NAS sometimes feels like an afterthought.
  ex: One provider has a 1M file limit for NFS volumes!!
- Frequently "high performance NAS" offerings are a bit light on the "high" part.

Observation 2: Cloud Offering Mismatch with EDA

Cloud HPC Offerings

Cloud HPC offerings are not a match for EDA HPC requirements:

- EDA HPC makes little use of accelerators (GPUs)
- EDA HPC rarely uses low latency interconnects designed for MPI workloads
- Performance depends less on massive scale out of middle tier CPUs, and more on
  - Top bin speed CPUs
  - High bandwidth, low latency CPU/RAM interconnects!
- EDA HPC uses higher RAM/CPU than typical HPC

Observation 3: Antique EDA Software

Our 1980's vintage EDA HPC application architectures don't mesh well with modern, 12-factor cloud applications.

Observation 3: Antique EDA Software

FlexLM

- Server instances are installed, and locked down via host ID, on fixed hardware.
- Server instances run for a long time. Scale up is via sysadmins installing more servers.
- Tools like Veritas are required for truly seamless HA

Cloud App

- Server instances run wherever – under the control of the application, not a sysadmin.
- Dynamic server instance start-up/shutdown that adjust under application control.
- The application detects and corrects failures even across geographic regions.

Texas Instruments

Observation 3: Antique EDA Software

Note that FlexLM is just one example. Most software in EDA is simply not cloud aware.

Observation 4: Antique EDA Software Licensing

Licensing models are a barrier to fully utilizing available hardware!

Observation 4: Antique EDA Software Licensing

Cloud Hardware

- The cloud always has more CPUs when you need them – or so says the cloud sales guy!
- What wasn't said: You might have to use 2GHz CPUs!
- Slow CPUs are simply not economically viable because license cost fails to scale down with performance.

The Big Question

Question:

With higher per hour costs and generally lower performance in the public cloud, is it possible for a medium sized EDA HPC shop to get work done and actually save money in the cloud?

The Big Question

Hybrid Cloud

Across medium and large enterprises, the IT at large seems to be converging on an approach to minimize costs via a hybrid cloud methodology.

The Big Question

Hybrid Cloud – lean manufacturing applied to IT

The idea is familiar to manufacturing businesses. Long term leases or purchase for what you use all the time, and short term leases or outsourcing for what you use occasionally.

If your sales person needs a car 3 weeks of every month, then get a four year lease from Toyota. If your sales person needs a car 1 week of every month, send them to Enterprise. It is really that simple.

The Big Question

Hybrid Cloud – lean manufacturing applied to IT

The cloud works the same way. We have a higher cost per unit time external resource vs a lower cost per unit time internal resource.

Thus we can minimize total cost by utilizing resources in an internal cloud for what we need most of the time, and resources in an external cloud for short term requirements.

The Big Question

Question:

With per hour costs so high in the public cloud, is it possible for a medium sized HPC shop to actually save money?

MAYBE

Formulating the Question

This entire talk is about one equation... Really!

Texas Instruments

Formulating the Question

We want to know when the total internal cloud cost, T_I, is greater than the total external cloud cost, T_E. Said with math:

$$T_I > T_E$$

Formulating the Question

We want to know when the total internal cloud cost, T_I, is greater than the total external cloud cost, T_E. Said with math:

$$T_I > T_E$$

This simple equation is difficult to conceptualize because internal and external cloud costs are expressed in different terms.

$T_I > T_E$: Buy a server or run in a public cloud...

In the internal cloud we think in terms of lease cost and all the other stuff one must buy to support a server. In the external cloud we think in terms of CPU·hours and other consumables (storage, networking, etc...). We need to think in terms of common variables.

$T_I > T_E$: Buy a server or run in a public cloud...

As a first step, let's think about the costs over a full lease term.

$T_I > T_E$: Buy a server or run in a public cloud...

Internal Cost: T_I

L_C Server TCO (lease, etc...)

L_T Lease term (in hours)

$$T_I = L_C$$
$$T_I = \frac{L_C}{L_T} \cdot L_T$$

$$C_I = \frac{L_C}{L_T} = \text{Internal cost/hour}$$

External Cost: T_E

C_E External cost Per Hour

H Hours used

$$T_E = H \cdot C_E$$
$$T_E = \frac{H}{L_T} \cdot L_T \cdot C_E$$

$$U = \frac{H}{L_T} = \text{System utilization}$$

$T_I > T_E$: Buy a server or run in a public cloud...

A little high school algebra..

$$
\begin{aligned}
T_I &> T_E \\
\frac{L_C}{L_T} \cdot L_T &> \frac{H}{L_T} \cdot L_T \cdot C_E \\
C_I \cdot L_T &> U \cdot L_T \cdot C_E \\
C_I &> U \cdot C_E \\
\frac{C_I}{C_E} &> U
\end{aligned}
$$

$T_I > T_E$: Buy a server or run in a public cloud...

$$\frac{C_I}{C_E} > U$$

Where C_E is the external cost per hour, C_I is the internal cost per hour, and U is the system utilization.

$T_I > T_E$: Buy a server or run in a public cloud...

$$\frac{C_I}{C_E} > U$$

Where C_E is the external cost per hour, C_I is the internal cost per hour, and U is the system utilization.

This simple equation provides us with a *criteria* for deciding if a server would be more cost effective hosted in an external cloud!

$T_I > T_E$: Buy a server or run in a public cloud...

Example: A Frazzle server

The frazzle service is used by summer interns (June - August). Internal server cost: \$0.05/hour. Cloud provider cost: \$0.25/hour.

$$\frac{C_I}{C_E} > U$$

$$\frac{0.05}{0.25} > \frac{3}{12}$$

$$0.20 > 0.25 \quad \text{FALSE!}$$

No. It won't save us any money to put this service in the cloud.

Practical Matters

Cloud Cost: C_E

- Cloud providers charge for things other than CPUs
  - Network (this one is difficult to manage)
  - Storage (block, disks, object, etc...)
- Per hour costs can change – even over the life of a VM
- Prices change by time unit (hour, week, month, etc...)

Practical Matters

Cloud Cost: C_E

- Cloud providers charge for things other than CPUs
  - Network (this one is difficult to manage)
  - Storage (block, disks, object, etc...)
- Per hour costs can change – even over the life of a VM
- Prices change by time unit (hour, week, month, etc...)

Cloud billing feels a bit like a 1990's era cell phone bill. Complex by design, and leveraging that complexity to drive cloud profits.

Entire companies have appeared which do nothing but help enterprises understand cloud billing and control costs!

Practical Matters

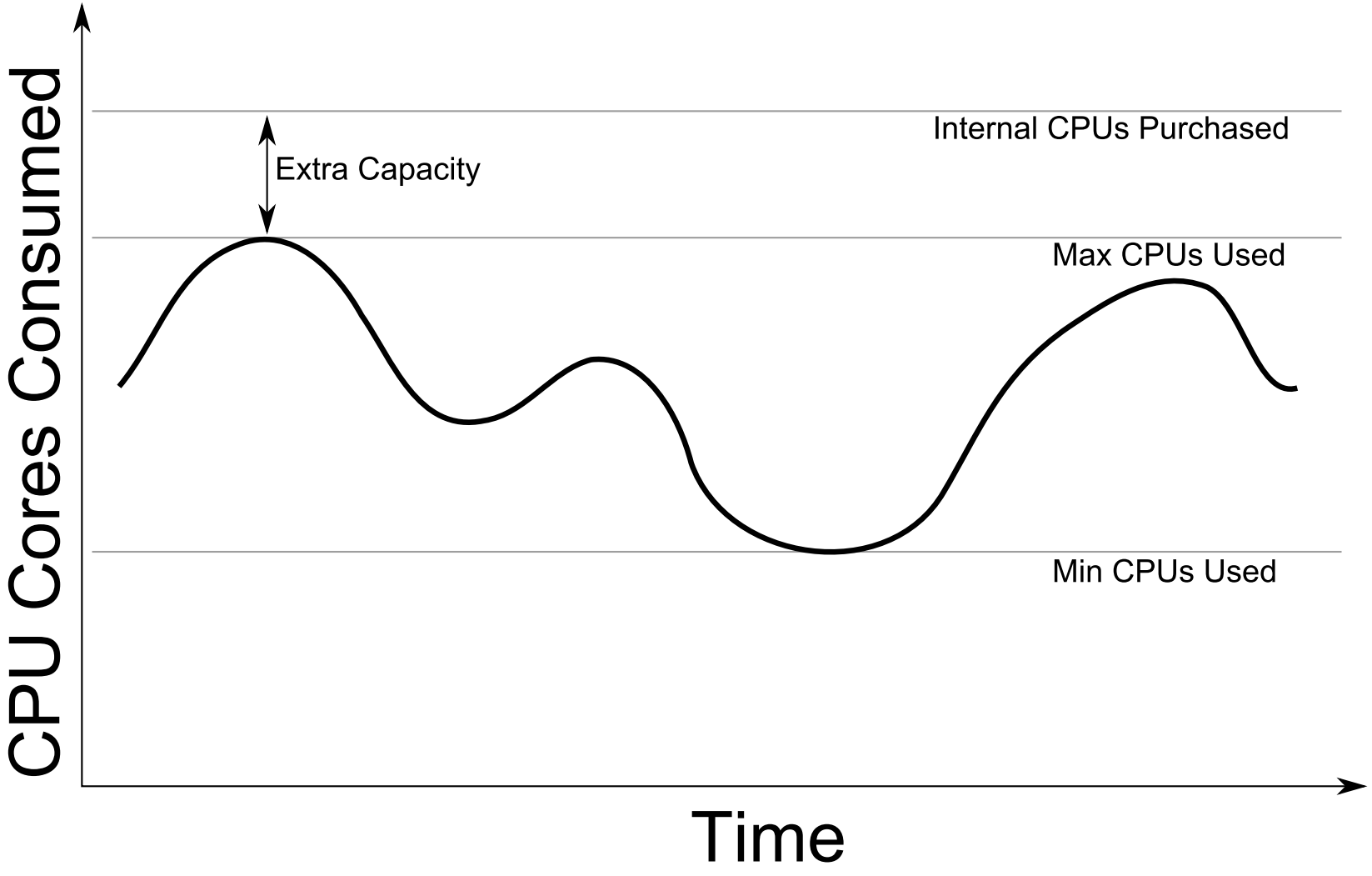
Server TCO: L_C

This one is hard, and requires a careful analysis of internal costs.

It is important to include everything you wouldn't have to pay for if you go to the cloud (Ethernet cables, switches, racks, cooling, power, console servers, etc....)

It can be difficult to determine what you can get rid of by moving to the cloud. Staff is a good example – i.e. at what hardware reduction level can you reduce staff?

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



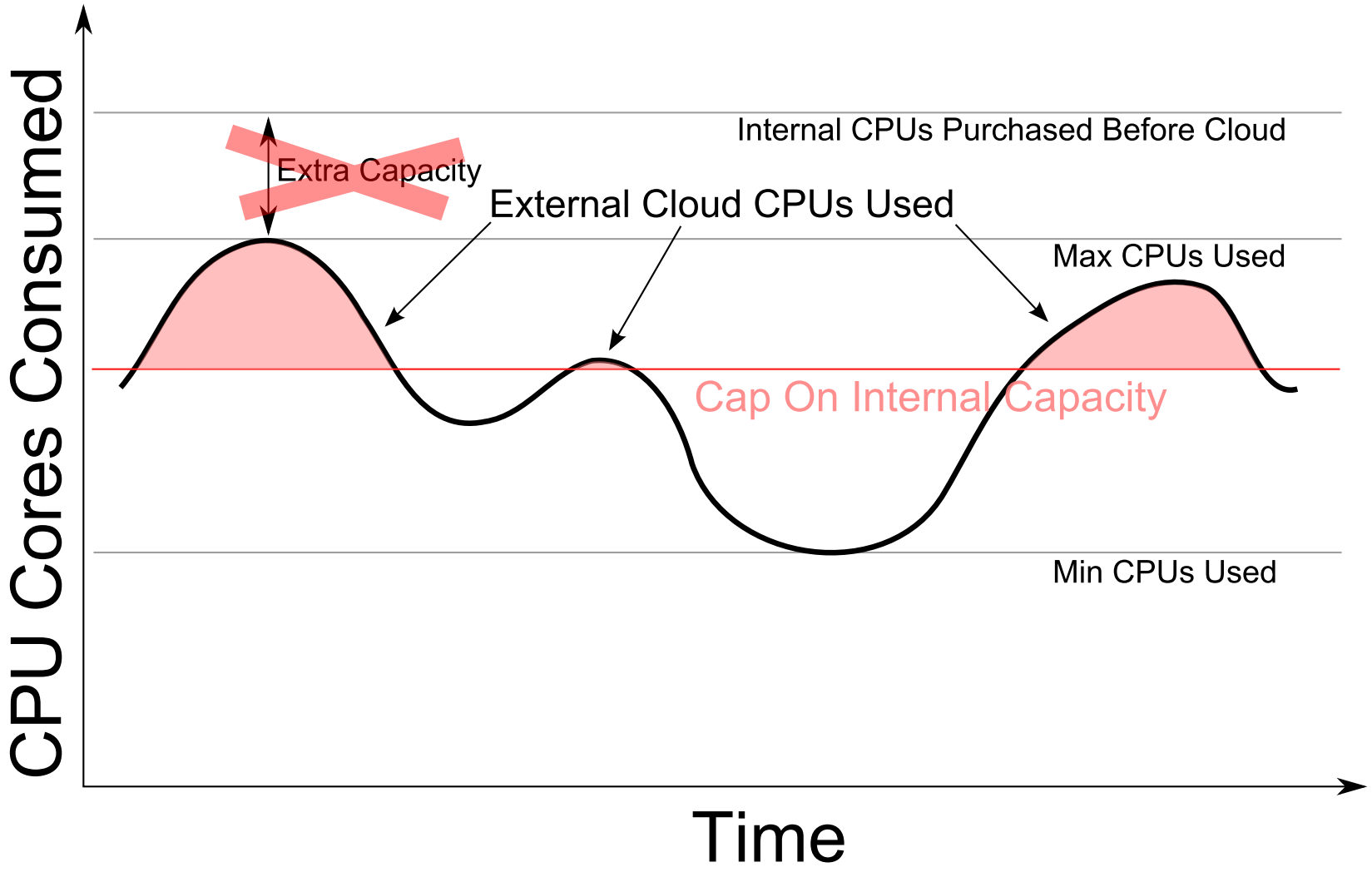
Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)

Cloud Bursting Defined

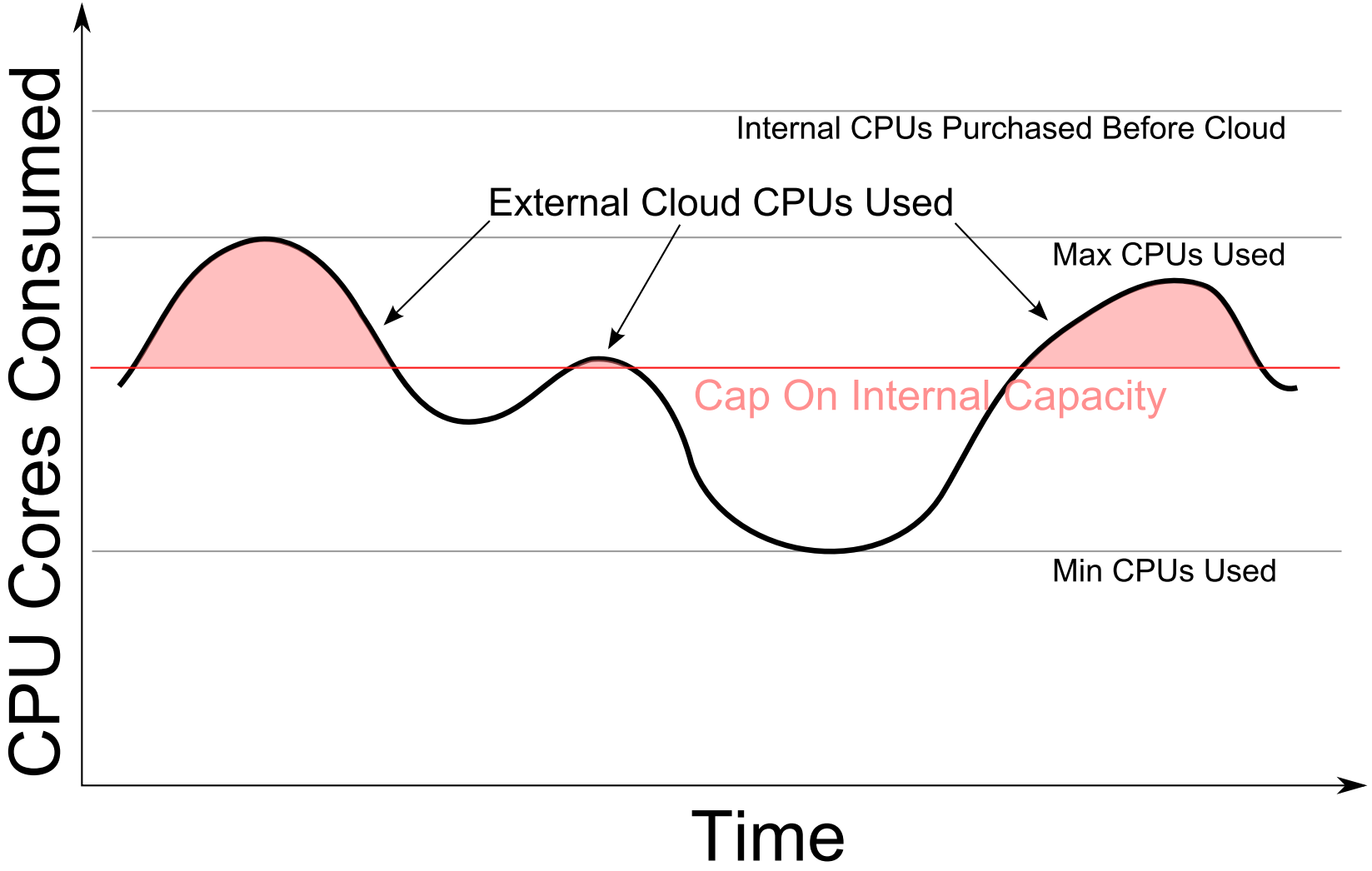
The idea is simple. Shave some "fluff" capacity from the top of an internal HPC cloud, and burst to an external cloud when necessary.

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



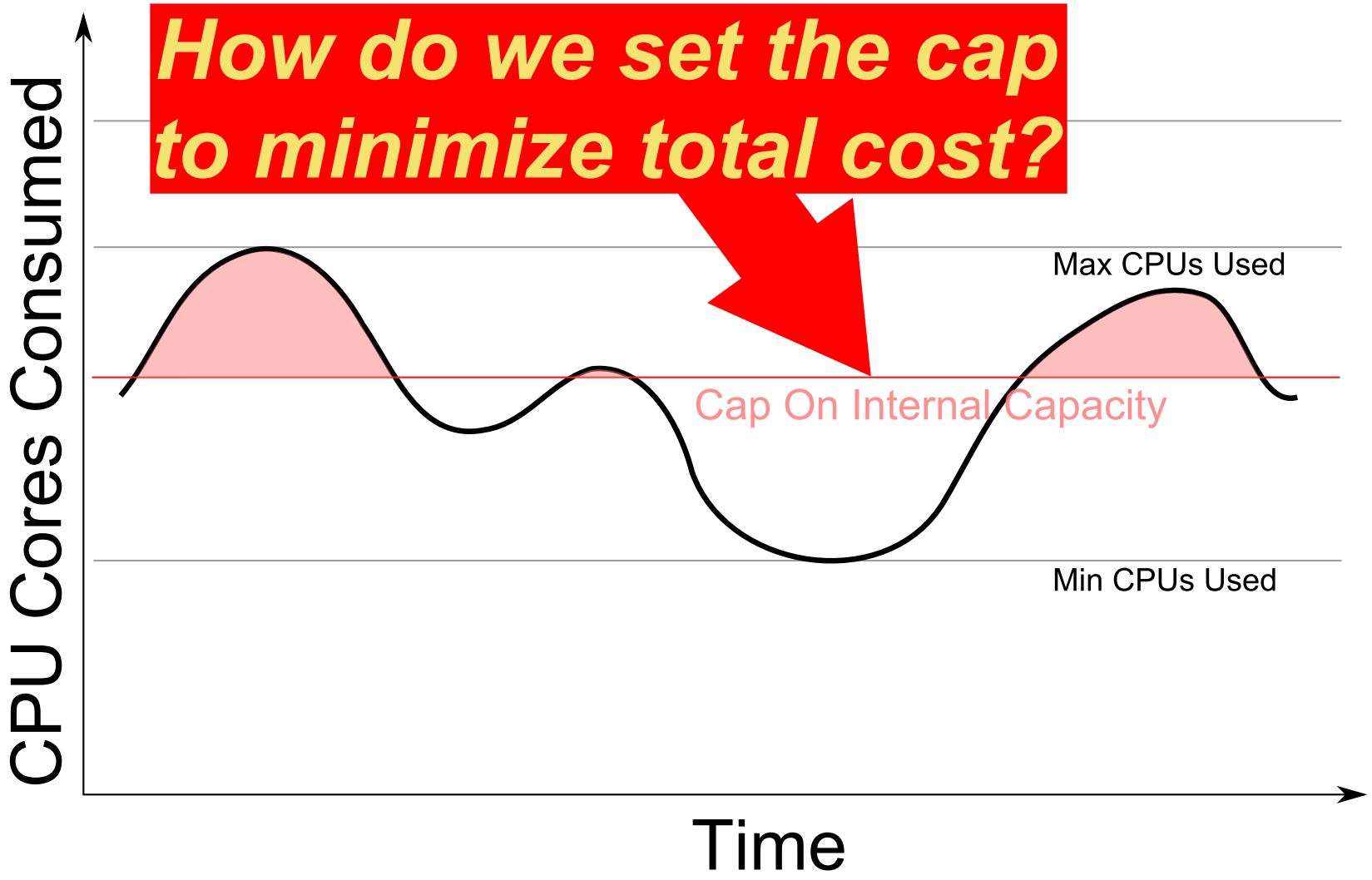
Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



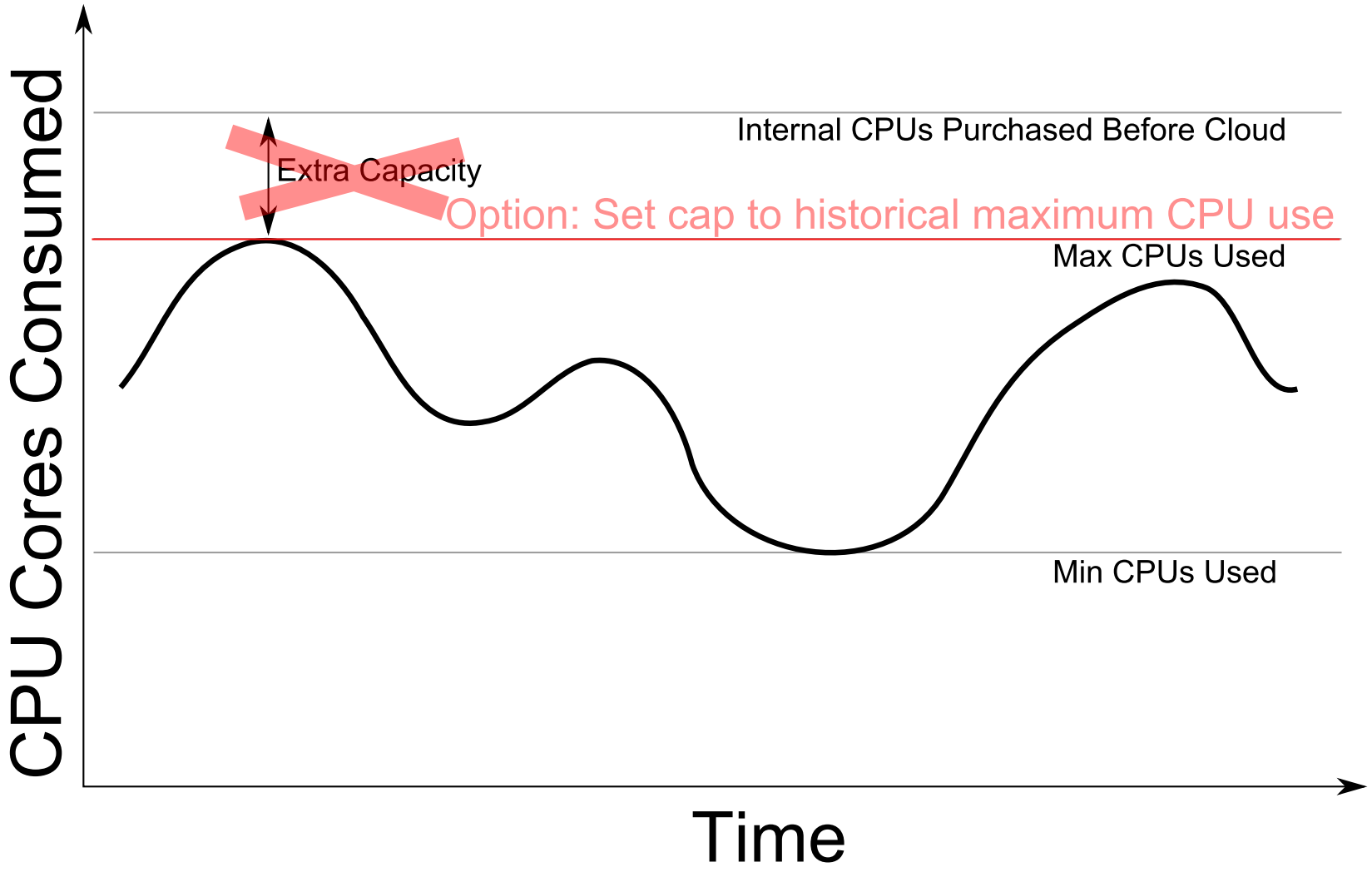
Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



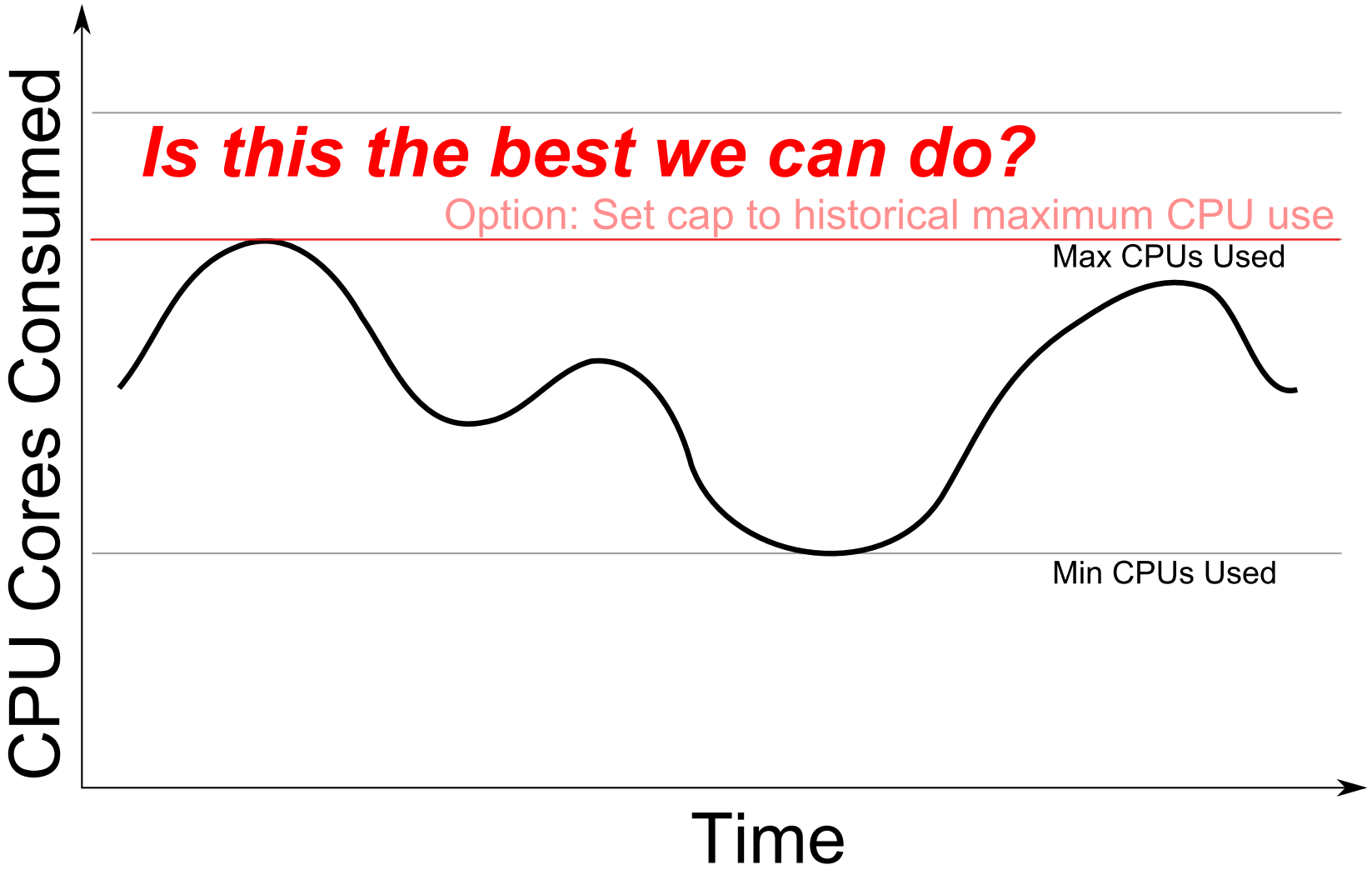
Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



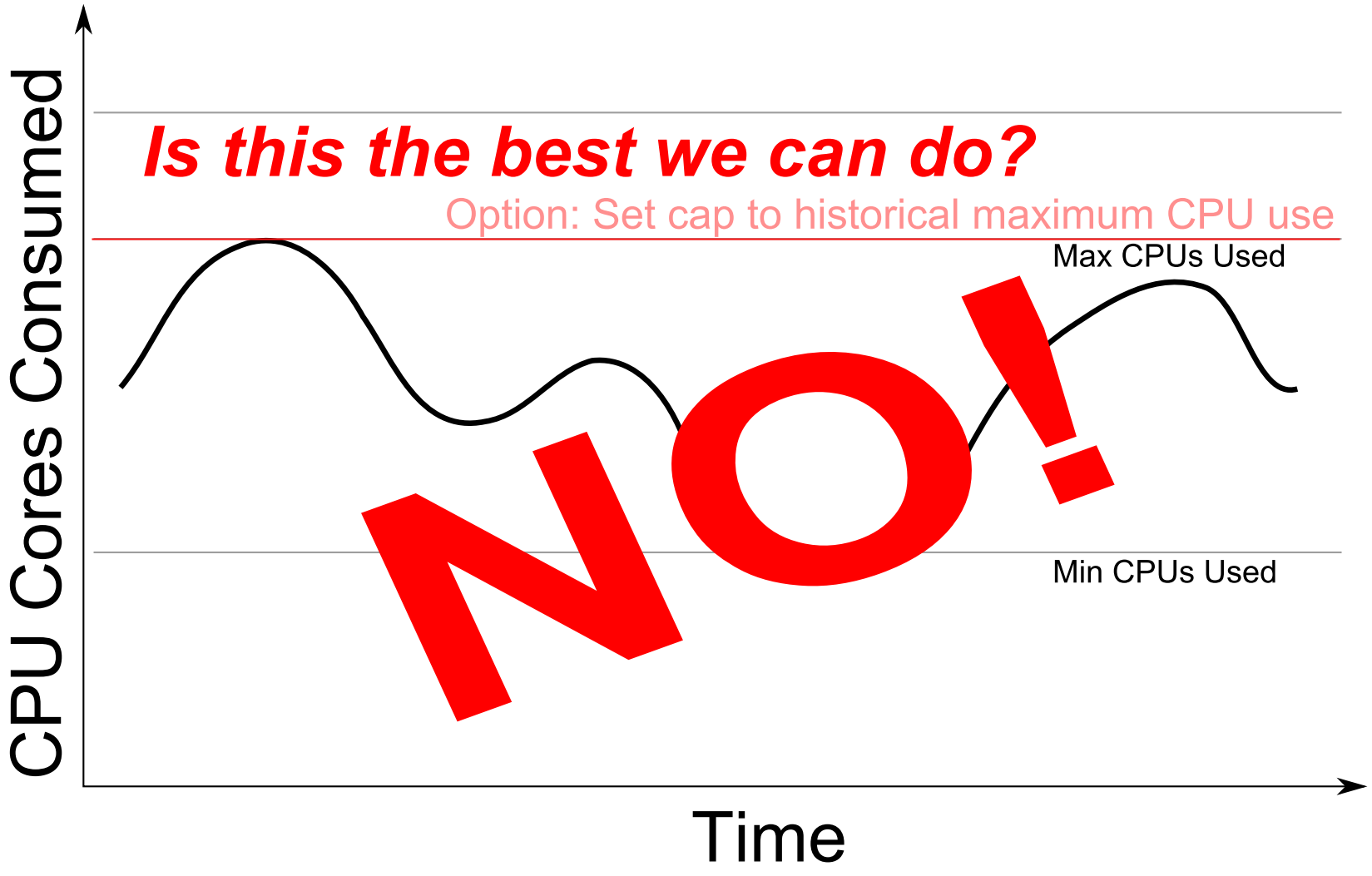
Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)



Texas Instruments

$T_I > T_E$: For HPC Clouds (Cloud Bursting)

So.. How do we set that cap?

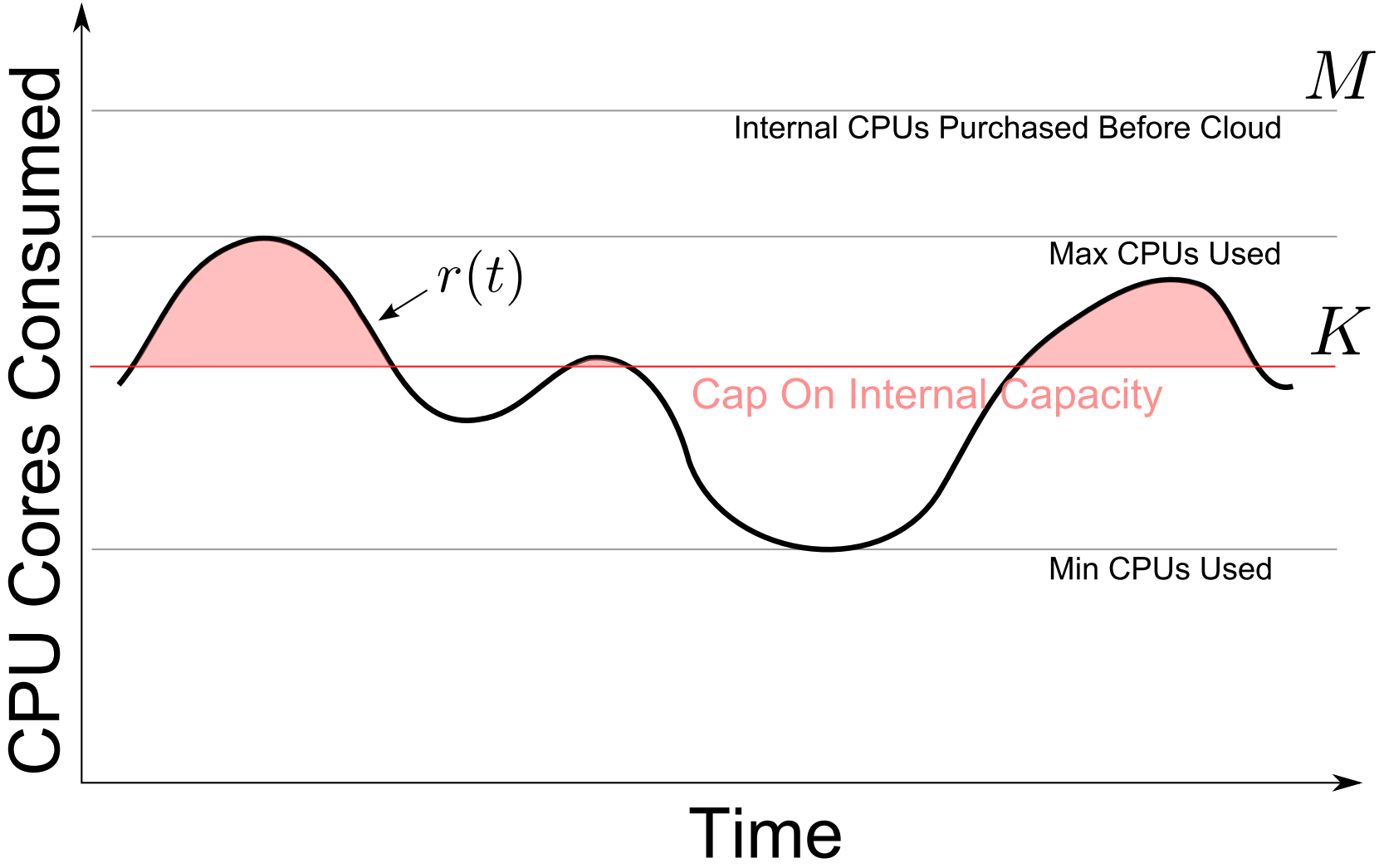
Setting the cap can be viewed as a classical optimization problem.

Setting the cap

First some variables

- M Number of cores purchased without cloud bursting
- K The cap: number of purchased using cloud bursting
- $r(t)$ The "cores running function". i.e. at time t the value $r(t)$ is the number of cores running jobs.
- S The support the cloudy part of r. i.e. $\{t|r(t) > K\} \subset \mathbb{R}$

Setting the cap



Setting the cap

A little manipulation..

$$
\begin{aligned}
T_I &> T_E \\
M \cdot C_I \cdot L_T &> K \cdot C_I \cdot L_T + C_E \int_0^{L_T} (r(t) - K)\, \mathbf{1}_S(t)\, dt \\
(M - K) \cdot C_I \cdot L_T + K \cdot C_I \cdot L_T &> K \cdot C_I \cdot L_T + C_E \int_0^{L_T} (r(t) - K)\, \mathbf{1}_S(t)\, dt \\
(M - K) \cdot C_I \cdot L_T &> C_E \int_0^{L_T} (r(t) - K)\, \mathbf{1}_S(t)\, dt \\
C_I \int_0^{L_T} (M - K)\, dt &> C_E \int_0^{L_T} (r(t) - K)\, \mathbf{1}_S(t)\, dt \\
\frac{C_I}{C_E} &> \frac{\int_0^{L_T} (r(t) - K)\, \mathbf{1}_S(t)}{\int_0^{L_T} (M - K)} dt \\
\frac{C_I}{C_E} &> \int_0^{L_T} \frac{(r(t) - K)\, \mathbf{1}_S(t)}{M - K} dt \quad \text{(bottom is constant)}
\end{aligned}
$$

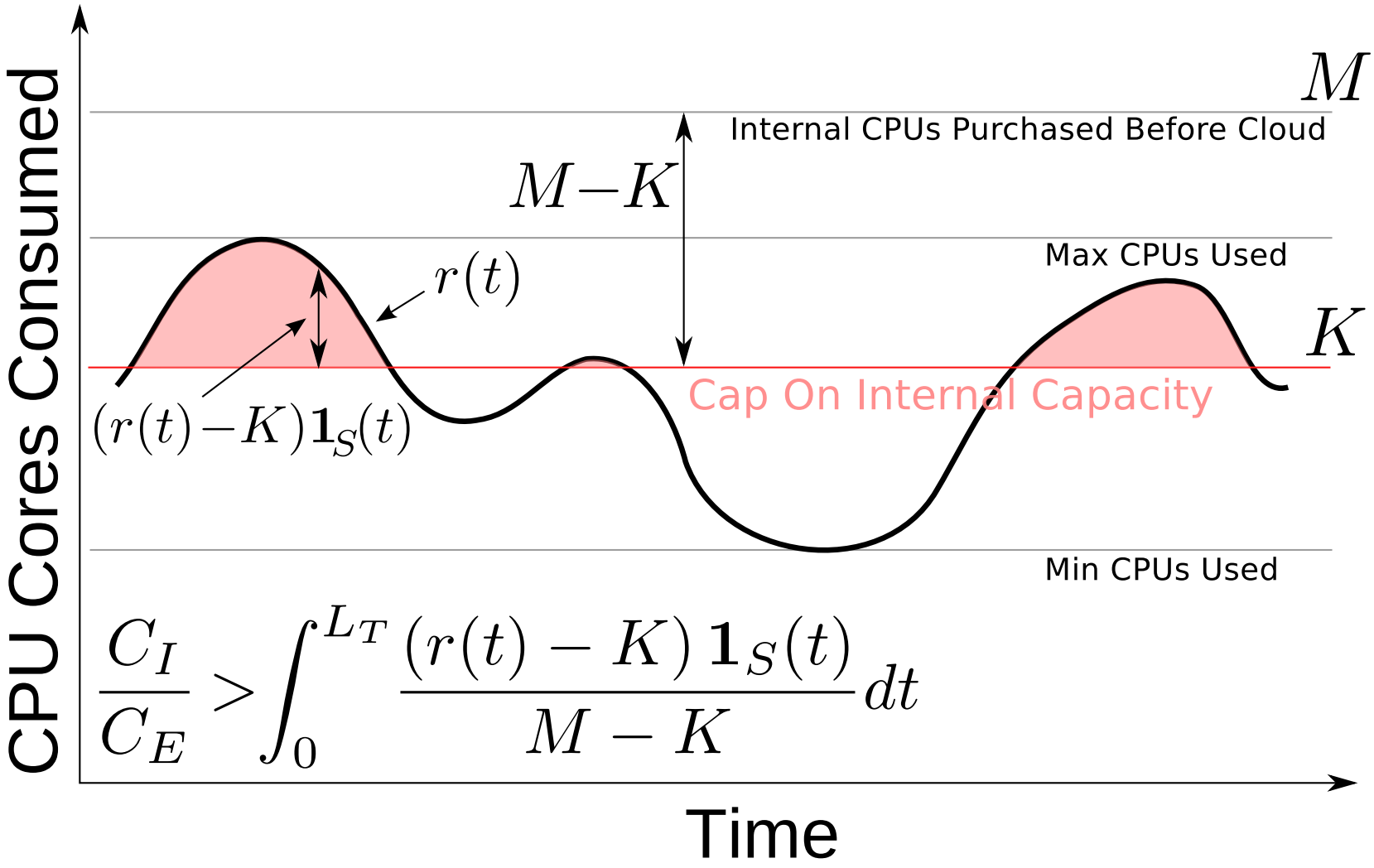
Texas Instruments

Setting the cap

What is this?

$$\frac{C_I}{C_E} > \int_0^{L_T} \frac{(r(t) - K)\,\mathbf{1}_S(t)}{M - K}\,dt$$

Setting the cap



Texas Instruments

Setting the cap

It is all about cloud utilization!

$$\frac{C_I}{C_E} > \int_0^{L_T} \frac{(r(t) - K)\,\mathbf{1}_S(t)}{M - K}\,dt$$

The RHS is the *average* utilization of the cloud resources – i.e. the time they spend on over the total time.

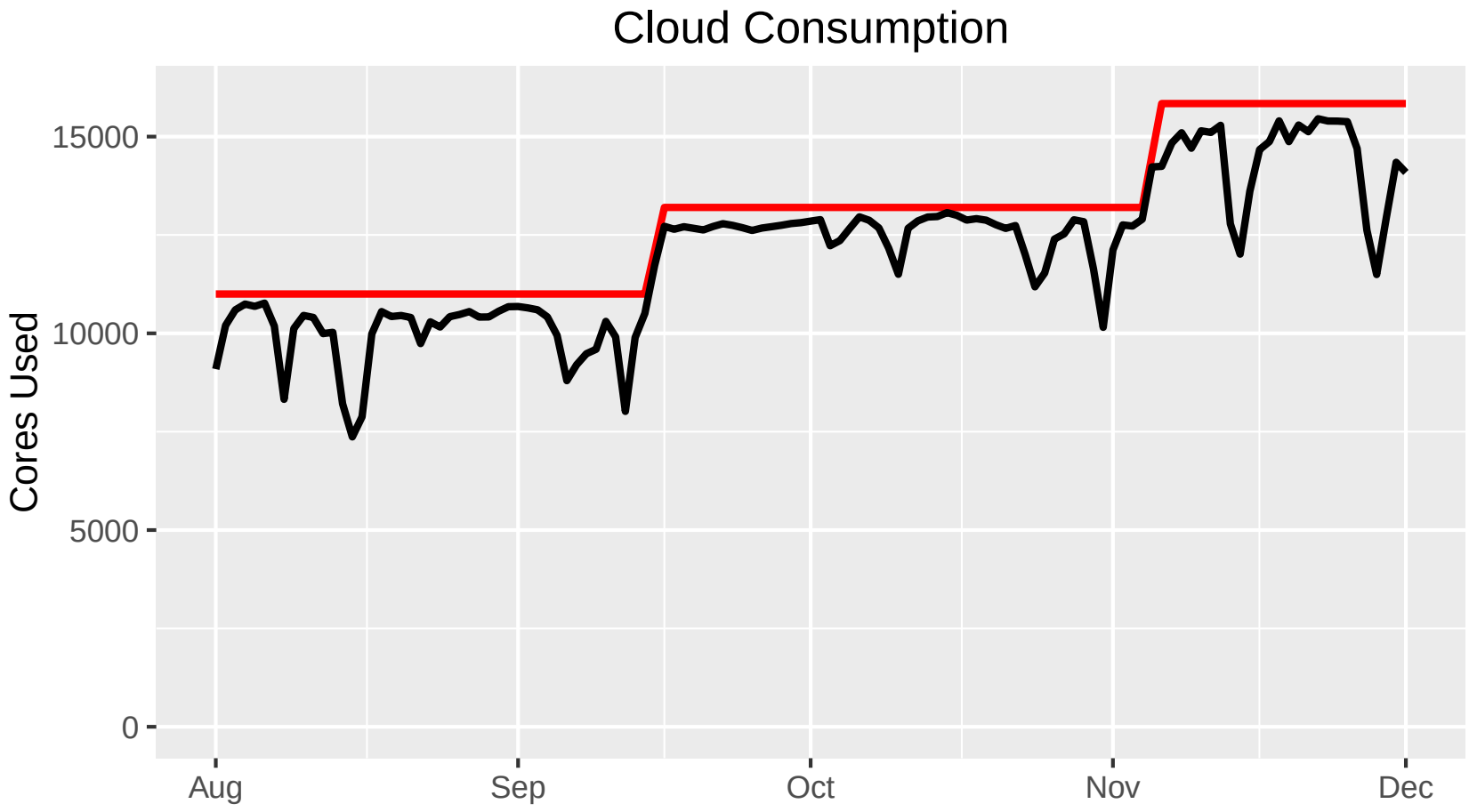
Note that $(r(t) - K)\,\mathbf{1}_S(t)$ is the instantaneous number of cores used and $\frac{(r(t)-K)\mathbf{1}_S(t)}{M-K}$ is the instantaneous utilization of of the cloud CPUs.

Setting the cap

In fact. It is the same equation we had before!

$$
\begin{aligned}
T_I &> T_E \\
\frac{C_I}{C_E} &> \int_0^{L_T} \frac{(r(t) - K)\,\mathbf{1}_S(t)}{M - K}\,dt \\
\frac{C_I}{C_E} &> U
\end{aligned}
$$

Real World Example of $r(t)$



Texas Instruments

Smoke & Mirrors

Some troubling details...

- The number of cores available is not constant over time!!
- The cutoff level (K) must change with capacity! Right?
- What is the form of $r(t)$? Can I even put it in an integral?
- Wait! What? How do I even compute $r(t)$?
- How can I guess what it will be in the future?

Smoke & Mirrors

A glimmer of hope...

- The final formula involves utilization – a unit-less value.
- If we rephrase the other quantities as percentages, perhaps we can make progress.
- Instead of $r(t)$, let's take a look at $u(t)$ – instantaneous utilization.

$u(t)$

Data Sources

- Sampled Host Utilization
  - CPU Utilization
  - Slot Utilization (number of running job slots)
  - Clipped Slot Utilization (ratio of number CPUs with at least 1 job dispatched to number of CPUs)
- Sampled running job slots
- Batch log data
  - Use job start/end time to compute number of running jobs for each second of the study interval

Data Source: Suggestion

- Combination of CPU and clipped slot utilization (5min samples):

$$\frac{\max(\texttt{num\_cores} \cdot \texttt{cpu\_util}, \texttt{alloc\_cores})}{\texttt{num\_cores}}$$

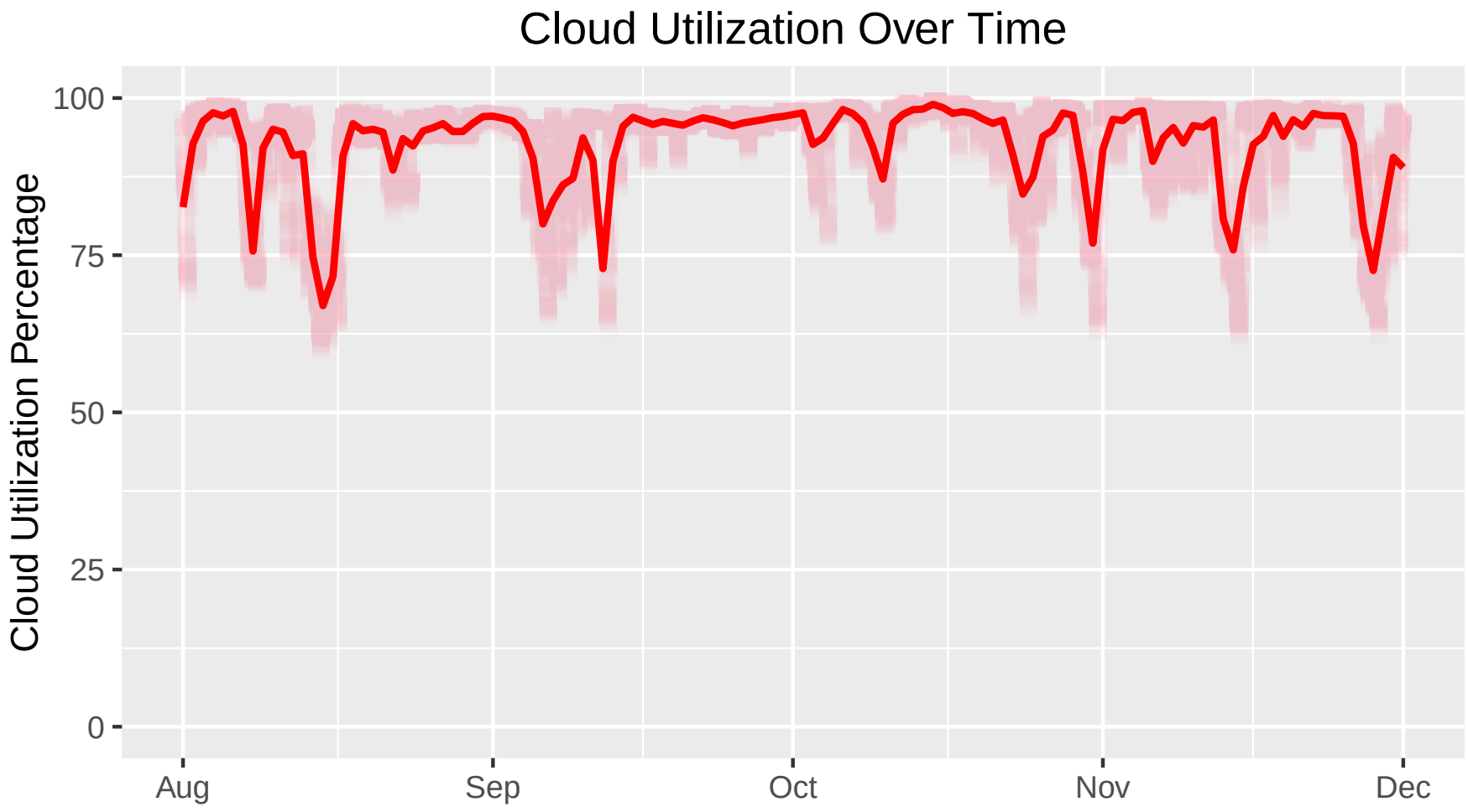
- Batch log data
  - To include/exclude hosts with cloud worthy jobs
  - Higher resolution job data
  - To focus on special job populations

$u(t)$

Data Source: WARNINGS!!

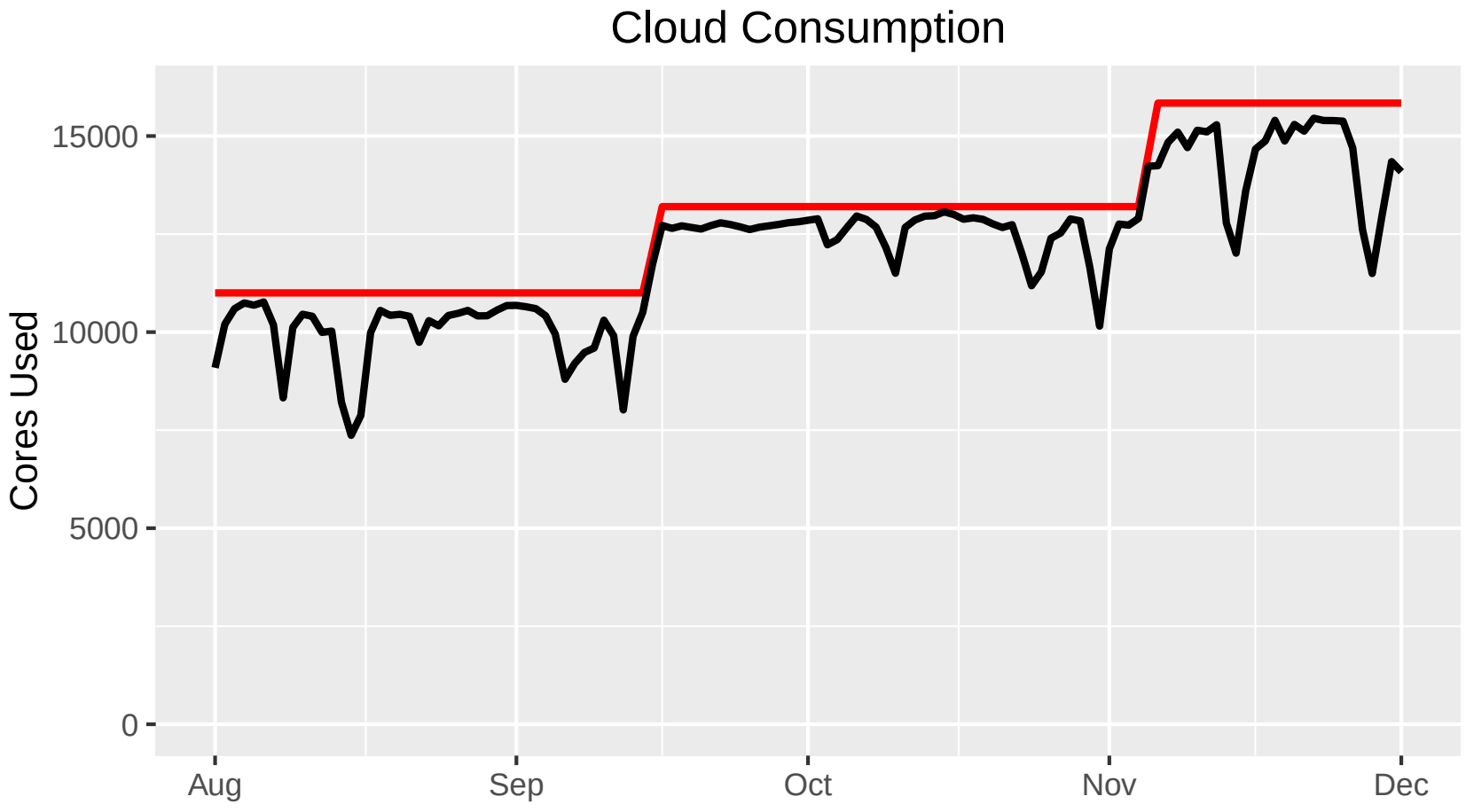
- Averages of averages
  - Correctly weight utilization measures with core count.
  - CPU utilization is frequently reported as an *exponentially weighted average*!
- Rounding & Hot Spots
  - Biased rounding is common in scheduling tools.
  - Reported running slot numbers can be sticky.

Cloud Utilization



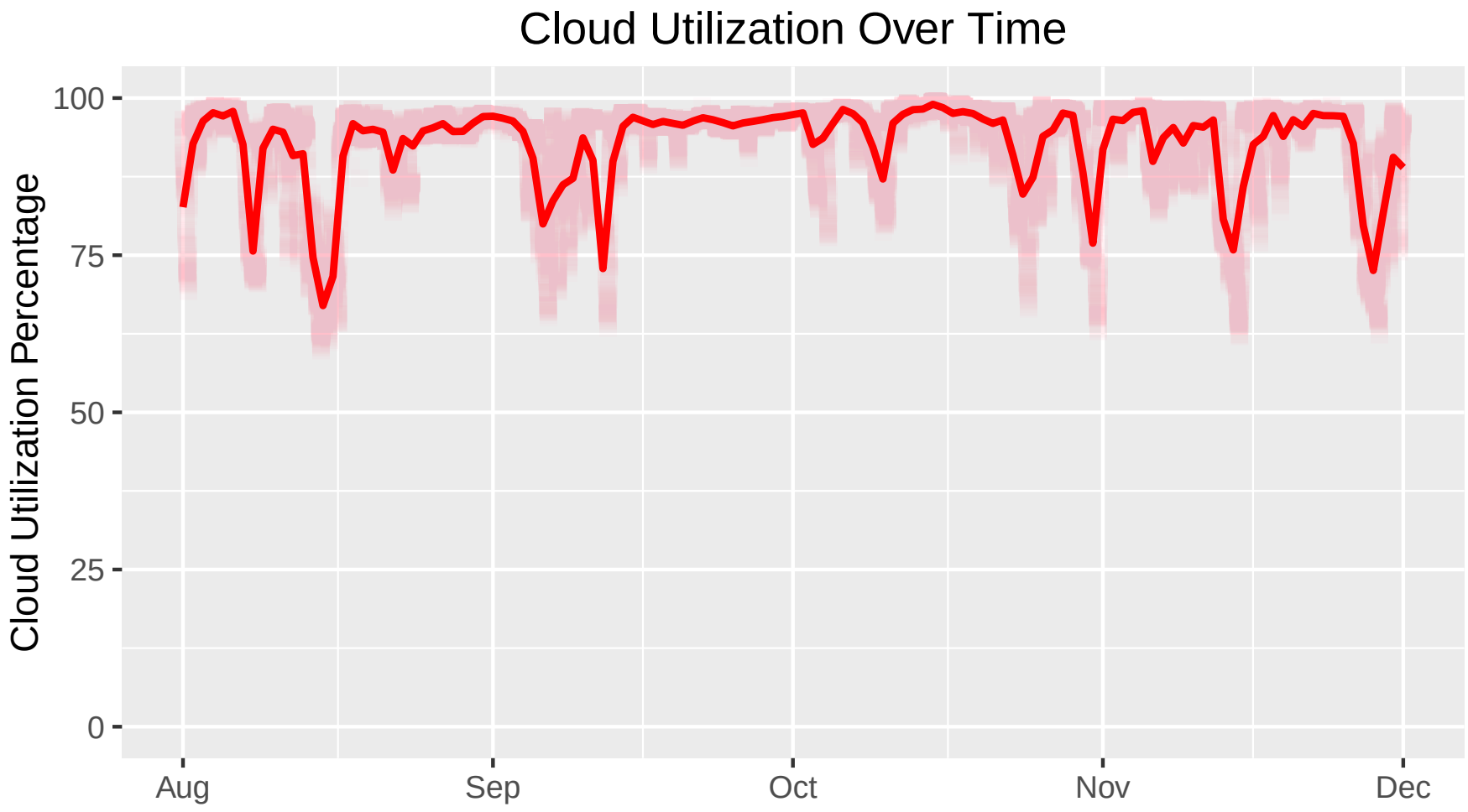
Texas Instruments

Cloud Utilization



Texas Instruments

Cloud Utilization



Texas Instruments

Cloud Utilization

An Observation

The utilization, $u(t)$, seems stationary with respect to capacity – over the illustrated interval several capacity changes occurred.

This is no accident. Most sane cluster managers increase capacity when queue time begins to edge up.

Cloud Utilization

Some troubling details...

- ~~The number of cores available is not constant over time!!~~
- ~~The cutoff level (K) must change with capacity! Right?~~
- What is the form of $r(t)$? Can I even put it in an integral?
- ~~Wait! What? How do I even compute $r(t)$?~~
- How can I guess what it will be in the future?

Cloud Utilization

A Technical Detail

If we expect to use this $u(t)$ function as a proxy for $r(t)$, then it is important to note the domain and range of this function:

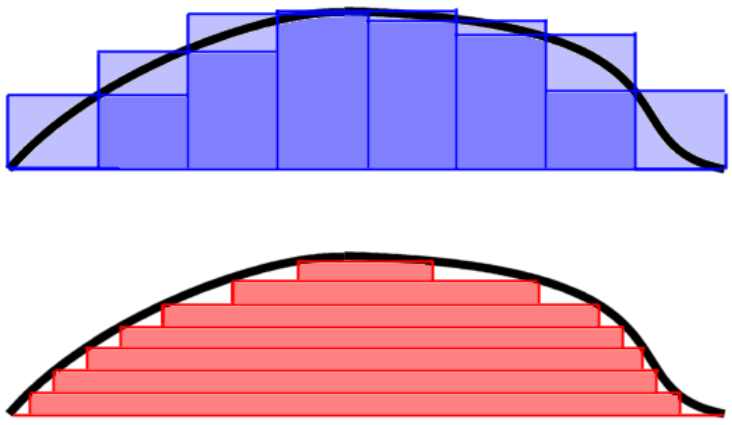
- Domain: Integers (POSIX Time Integers)
- Range: Integers or scaled integers depending upon data source

That is to say both domain and range are discrete sets which renders the previously presented integrals in a dubious light when viewed through the lens of undergraduate calculus.

Cloud Utilization

A Technical Detail & A Hint

While $u(t)$ may be problematic for the Riemann integral from undergraduate calculus, the more advanced Lebesgue integral works just fine. Unlike the Riemann integral, the Lebesgue integral chops functions up not into vertical chunks but horizontal ones.



Texas Instruments

Cloud Utilization

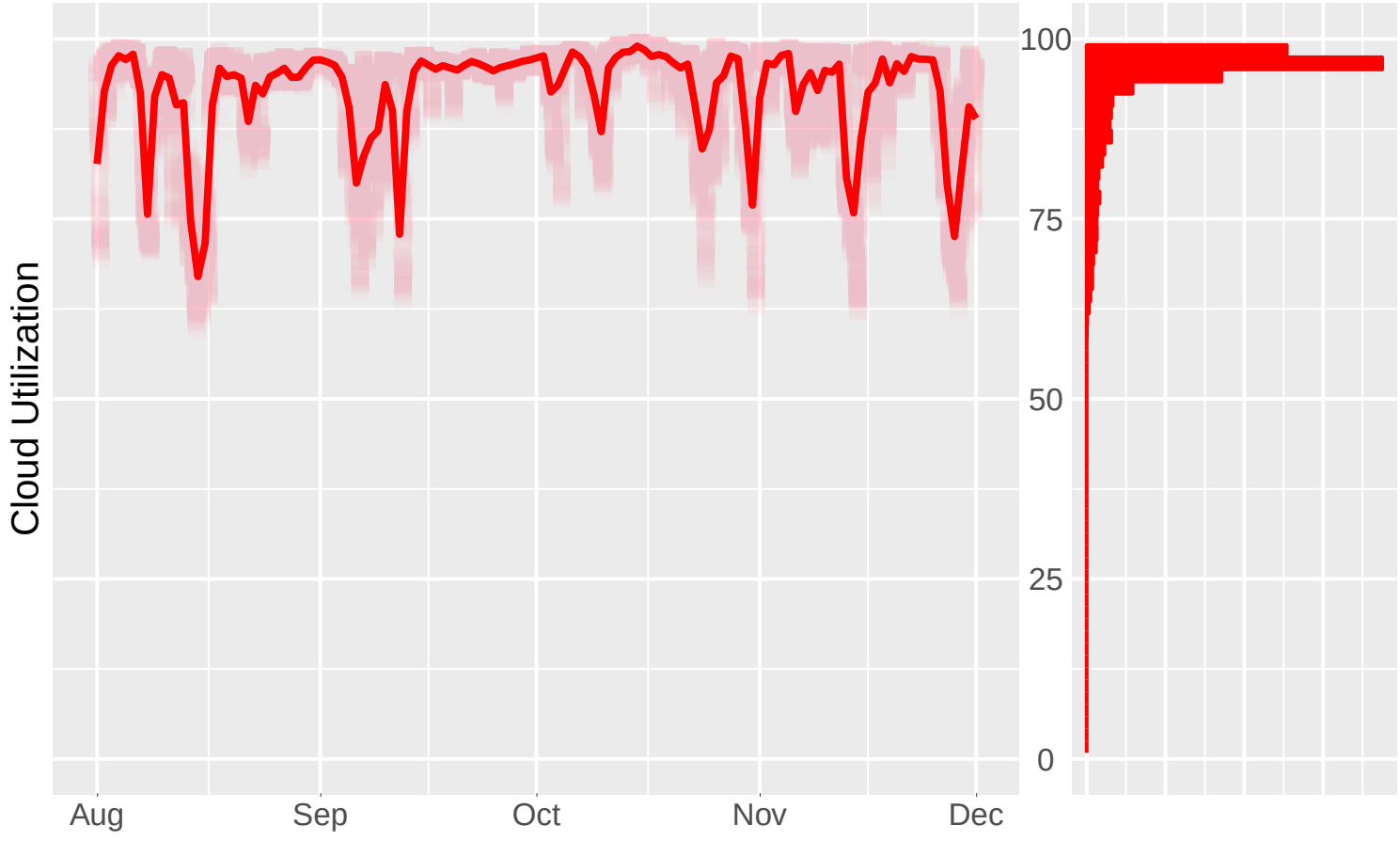
Eureka!

The stability of the delta between total capacity and used capacity combined with the idea of chopping the function up horizontally strongly suggests using the probability distribution of the delta in our model!

This provides a neat solution to our problem of variable total capacity: Simply use the $u(t)$ function in the model, and set K as a percentage delta from total capacity.

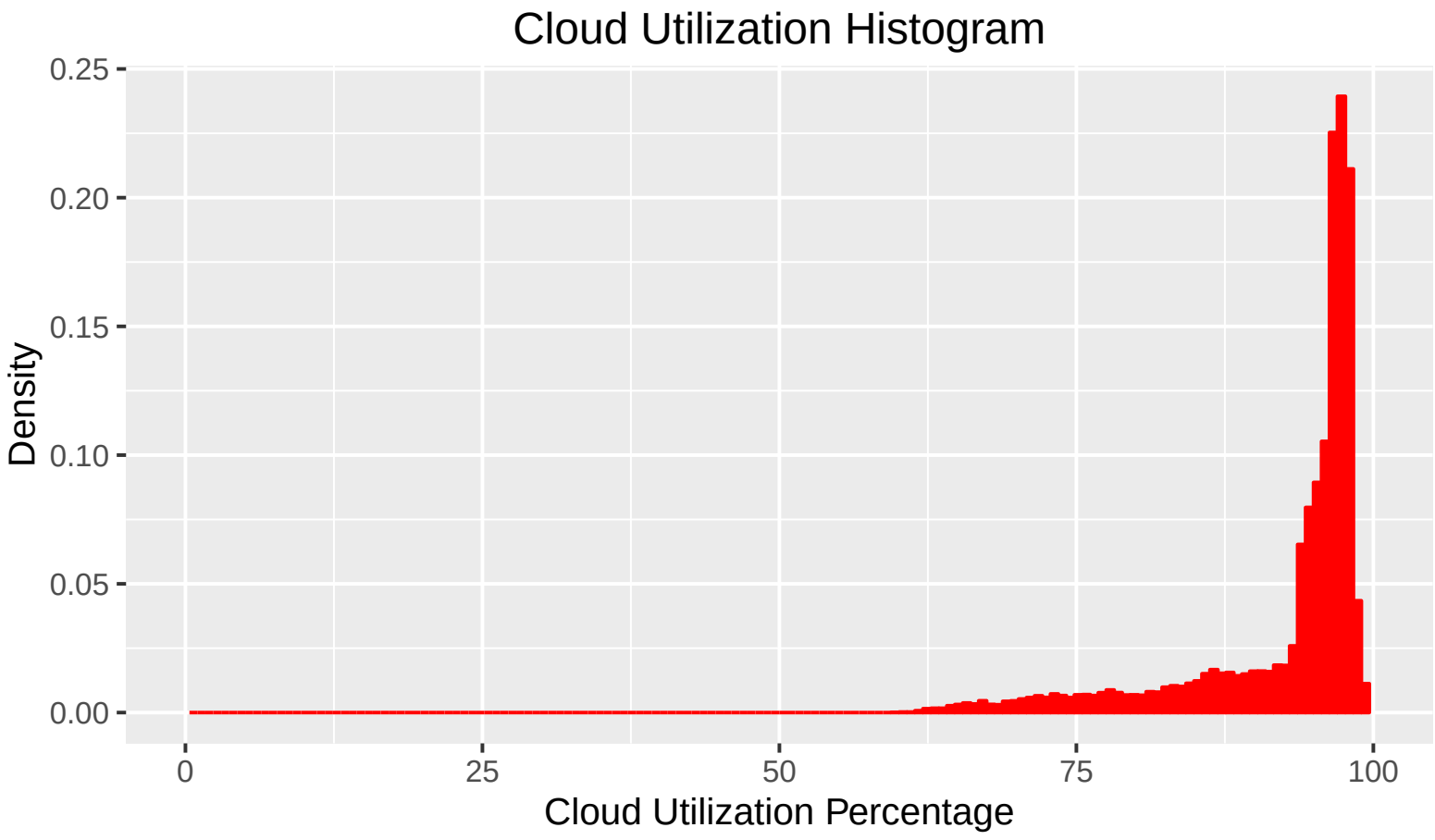
Finally, we transform the very hard problem of predicting future values of $r(t)$ with a simple probability problem.

Cloud Utilization



Texas Instruments

Cloud Utilization



Texas Instruments

Cost Reduction vs. Farm Size Reduction

The Idea

Perhaps the most obvious computation is to do a "what if" with the time series data. i.e. set a limit, and compute what the costs would be for the data you have.

Cost Reduction vs. Farm Size Reduction

How

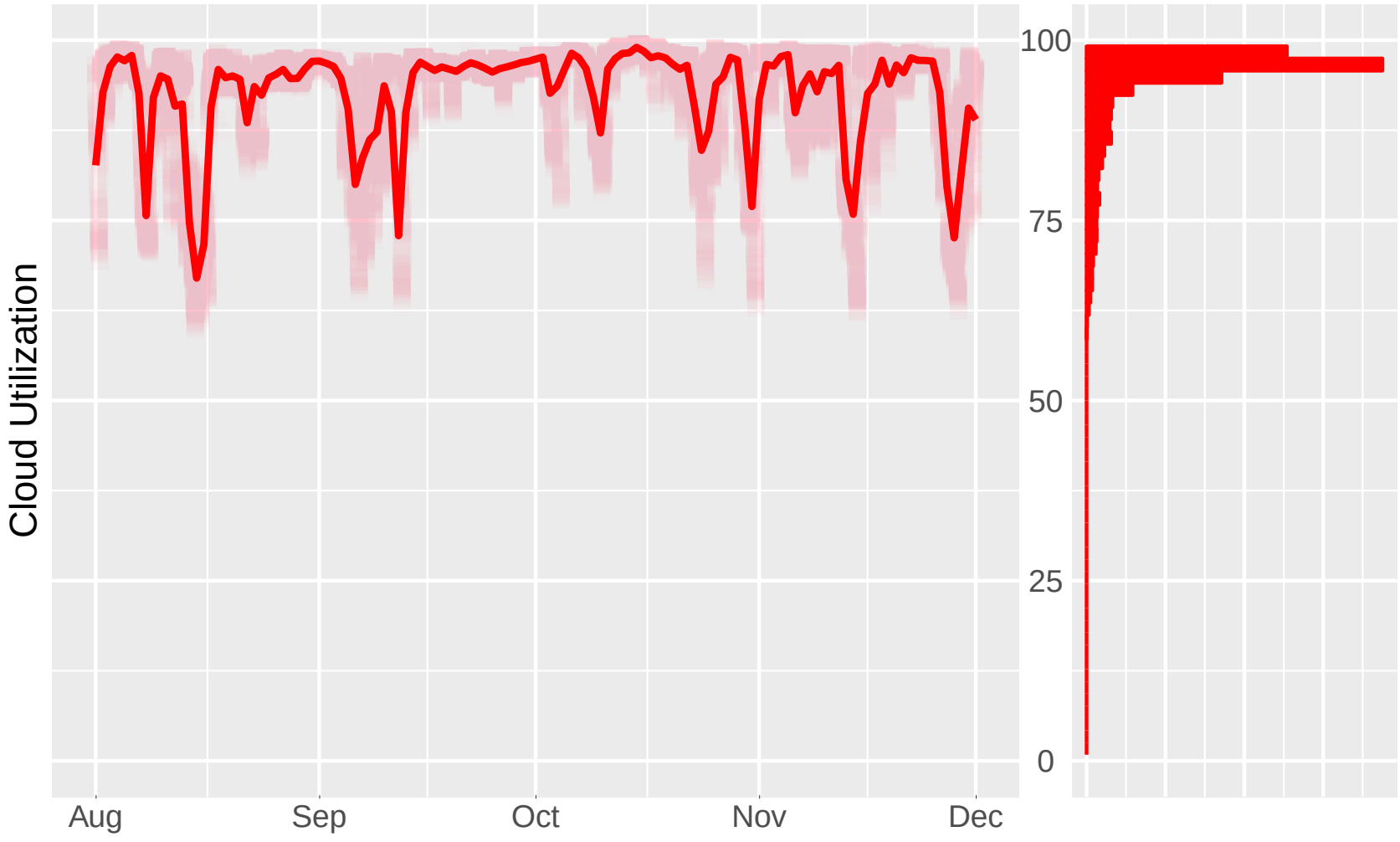
Option 1: Simply run through the data, and compute the costs. That is to say, we compute the value of the integral in our formula.

Cost Reduction vs. Farm Size Reduction

How

Option 2: Use the histogram. This is the same computation because the histogram can be thought of as the value of the integral you want to compute.

Cost Reduction vs. Farm Size Reduction



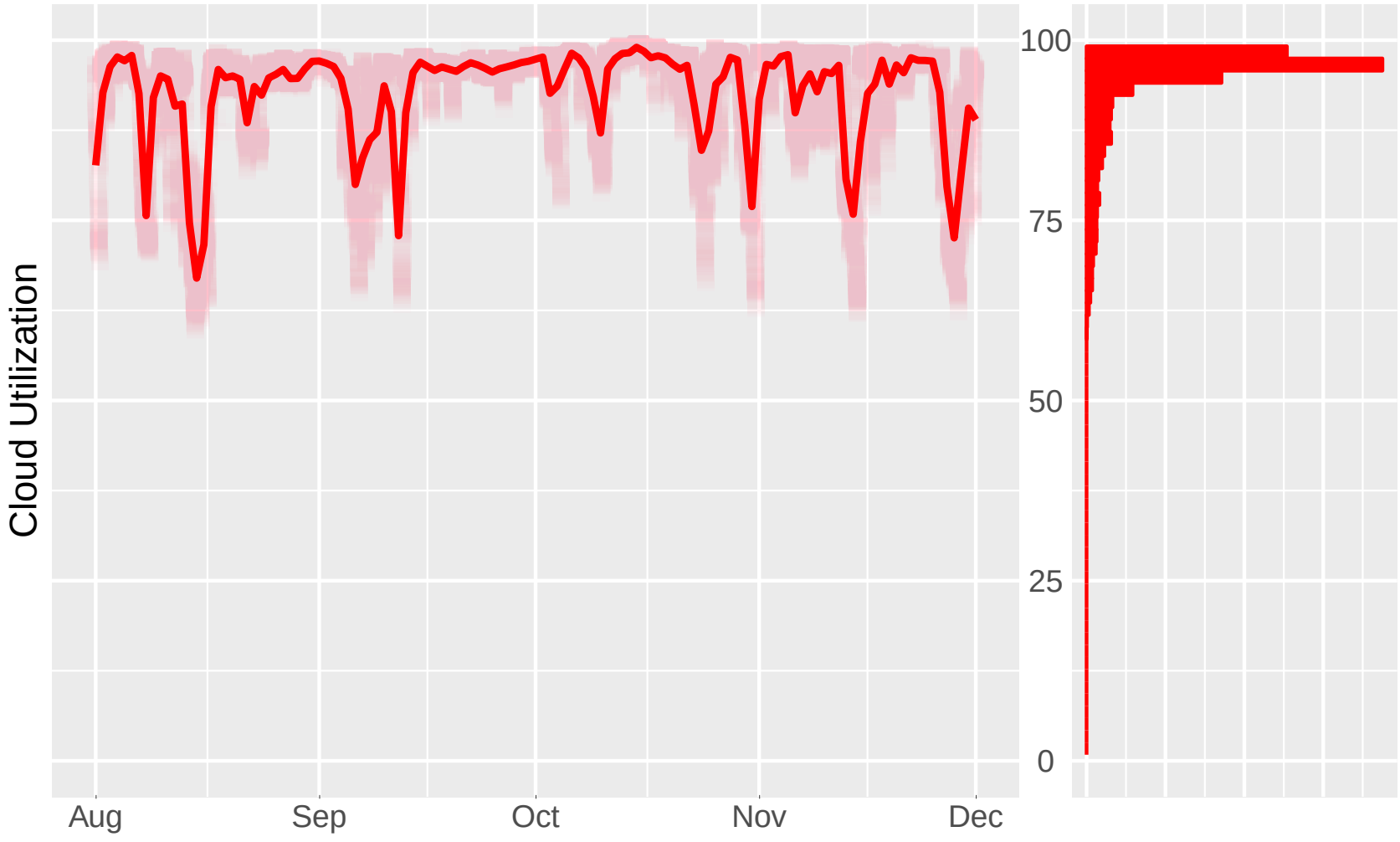
Texas Instruments

Cost Reduction vs. Farm Size Reduction

How

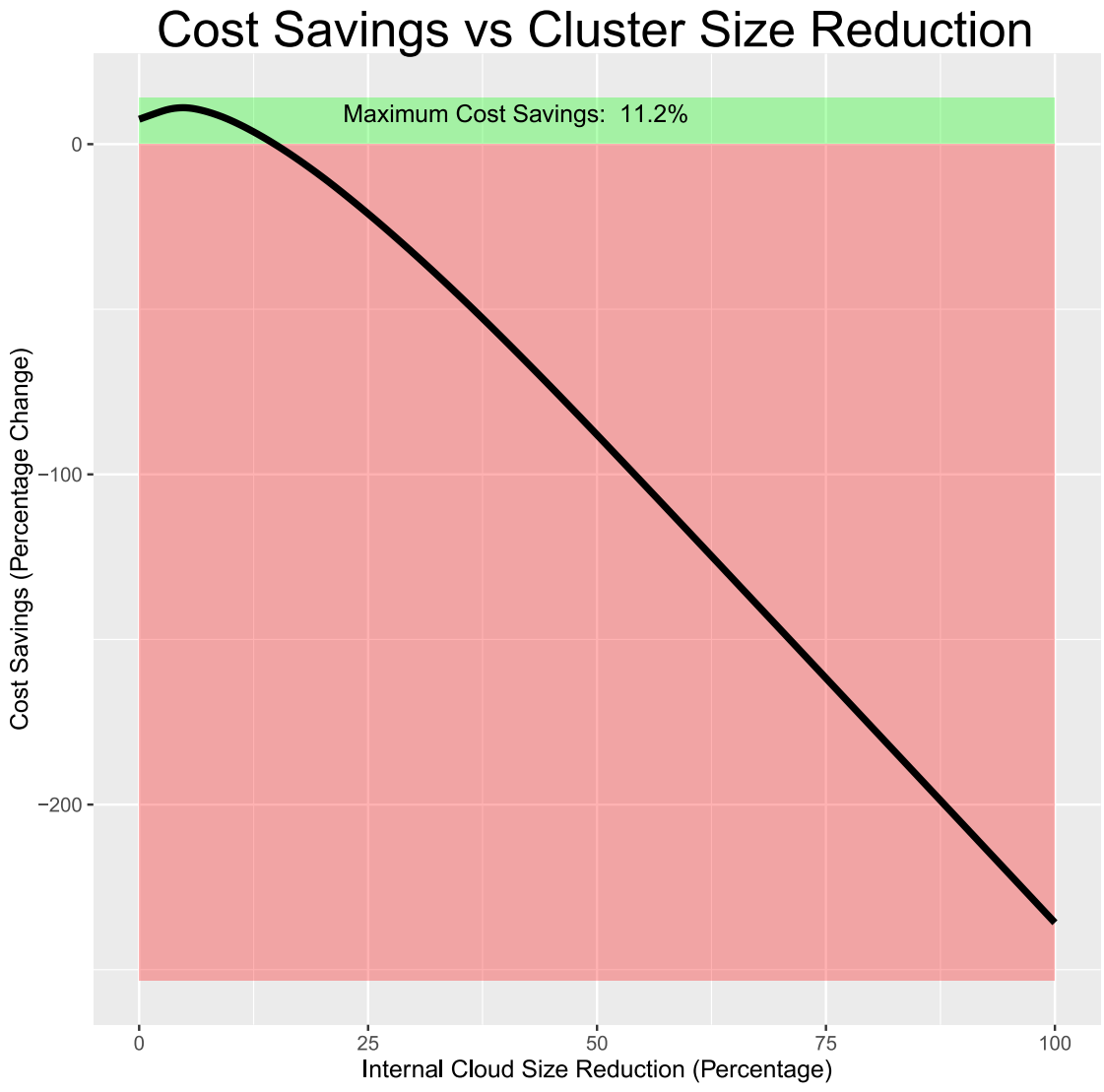
Using the histogram method is faster, and makes it easy to compute the savings for hundreds of different choices of internal cloud size cutoff. i.e. why do the what-if for only one choice when we can do it for all choices!

Cost Reduction vs. Farm Size Reduction



Texas Instruments

Cost Reduction vs. Farm Size Reduction



Texas Instruments

Cost Reduction vs. Farm Size Reduction

Extra Credit

A third way of creating the previous graph is to do a Monte Carlo simulation for each choice K. This has the advantage of providing a full probability distribution of events for each choice! We could use this distribution to add an envelope to the graph showing the 90% probability interval.

Confidence Intervals

What are the odds?

One question very common question:

"What is the chance we actually spend more?"

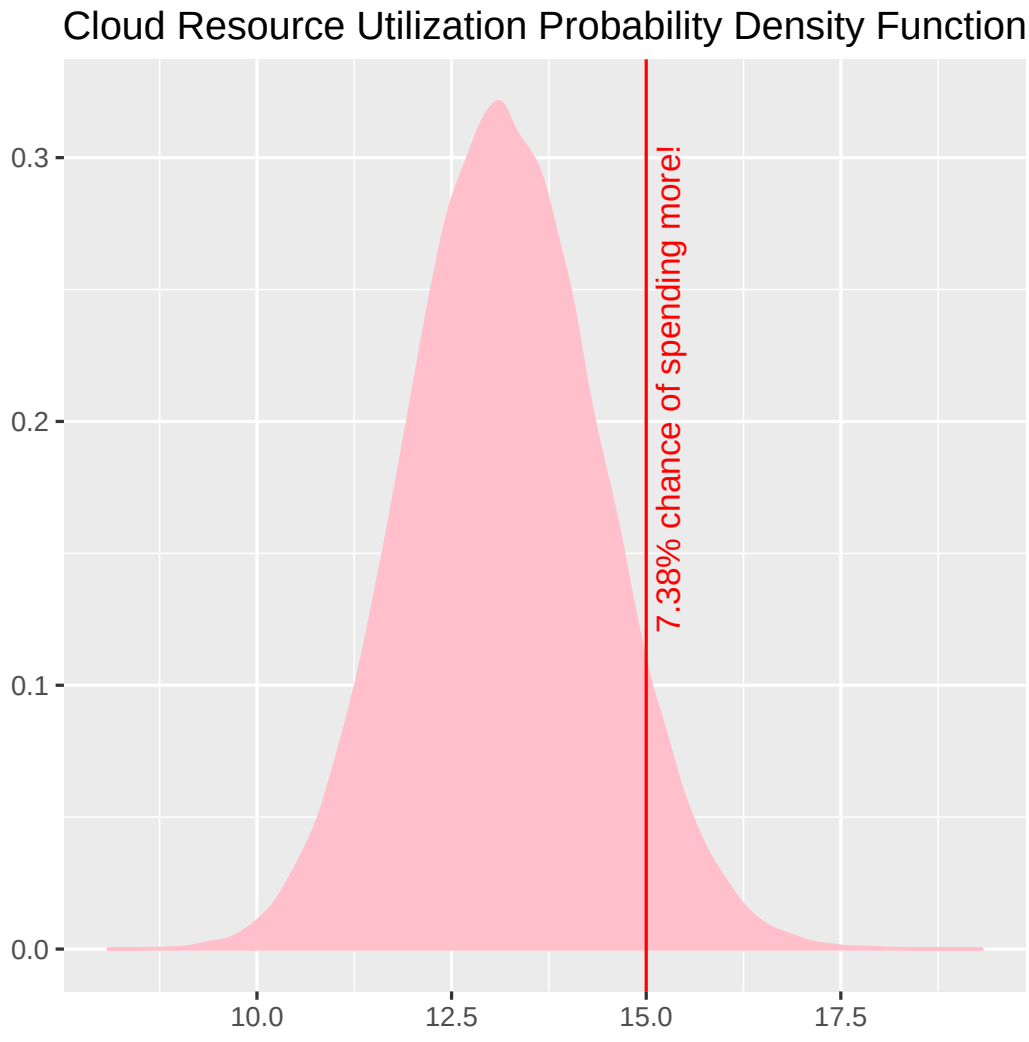
Confidence Intervals

Example

Suppose your $\frac{C_I}{C_E} = 15\%$, and you have set your internal capacity cap at 95%.

Assuming demand stays the same, what are the odds you actually spend more with a hybrid cloud strategy due to simple bad luck?

Confidence Intervals



Texas Instruments

Confidence Intervals

Monte Carlo: WARNINGS!!

- Pseudo Monte Carlo will normally work just as well as true Monte Carlo.
- Because of biased rounding and artificial hot spots in some data sources, some smoothing is important for most data sources!
  - Density estimators work fine.
  - Simple convolution filters or moving average schemes (like `lowess`) work well.
  - Regardless of technique, tune bandwidth to remove artifacts. Simple graphical checks are usually sufficient.

Thank You!

TEXAS INSTRUMENTS

Questions

The following slides document questions received regarding the previous material.

Questions

Is that Lebesgue stuff really necessary?

No. The Lebesgue integration approach is natural for someone well versed in statistical and probability theory; however, this is not strictly necessary. A computer scientist would likely take a rather different, more direct, approach as $u(t)$ is, or may be coerced to be, discrete. Both approaches are mathematically equivalent. In fact, the "bag-o-sums" approach is a useful world view when it comes time to actually implement these ideas in code.

Note that the final simulations might not be with a discrete PDF depending on the technique used to compute the density approximation.

Questions

What about data glitches?

Most of this work is predicated on the existence of reasonably curated compute cloud data. Most of that curation can be done in an automated fashion via standard methods for outlier detection and some sanity checks before data is consumed.

That said, this particular technique (Monte Carlo on an empirical PDF) is inherently more robust to small numbers of errant data points than some traditional techniques. One critical point where bad data can have a multiplicative effect is the histogram smoothing or density approximation step – use a robust technique if you have messy data!

Questions

What about initial costs?

Initial investments are required for the cloud (firewalls, setup fees, etc...). One may simply add these to the savings curve; however, your finance team is likely to want to see that data amortized over a depreciation cycle. In this case it is appropriate to simply adjust the C_E value for the first depreciation cycle.

Note that if lease periods and depreciation cycles frequently don't match, so some care is required.

Questions

What about break delta costs?

The analysis changes significantly at infliction points requiring point expenditures – i.e. that point when you need to buy a bigger firewall, need to add a mains feed, etc... For me these tend to be the exception rather than the rule, and so I normally handle these as a special case requiring a manual analysis. For automated reporting it is important to put an envelope around the filter options available so as to fence off such cases.

Questions

What if my PDF isn't stable over time?

This happens! I have had very good luck using standard time series techniques in this case to extract "random", "cyclic" and "trend" components from the time series:

`https://www.mitchr.me/SS/exampleR/rcode/timeSeriesDecomp.html`

The random and high frequency cyclic components may be combined together to create the PDF. In this context "high frequency" is with respect to the time period under analysis. For example daily cycles can be pushed into the PDF for an analysis over two years. Then do the analysis as before, and add the trend component. One useful graph is the savings adjusted over time. For example a monthly graph with savings at the optimal K value for that month.

Questions

Storing all the histograms will take a lot of space!

Instead of storing the raw histogram data, store the coefficients of a Chebyshev interpolation polynomial. Experience has shown that most of the time you can get away with 6 points, but may need as many as 20. Note that this creates a continuous function for your PDF which makes some simulation software less difficult to use. BTW, it is also faster to compute the values via the polynomial than look them up in a table every time.

Questions

Have a question?

Feel free to contact me with questions, and I'll do my best to answer them.