When the world is not enough.

When the world is not enough.

Aka: Not cool enough to use an iPad to present from.

When the world is not enough.

Aka: Not cool enough to use an iPad to present from.
Aka:The mistakes we've made and will continue to make.

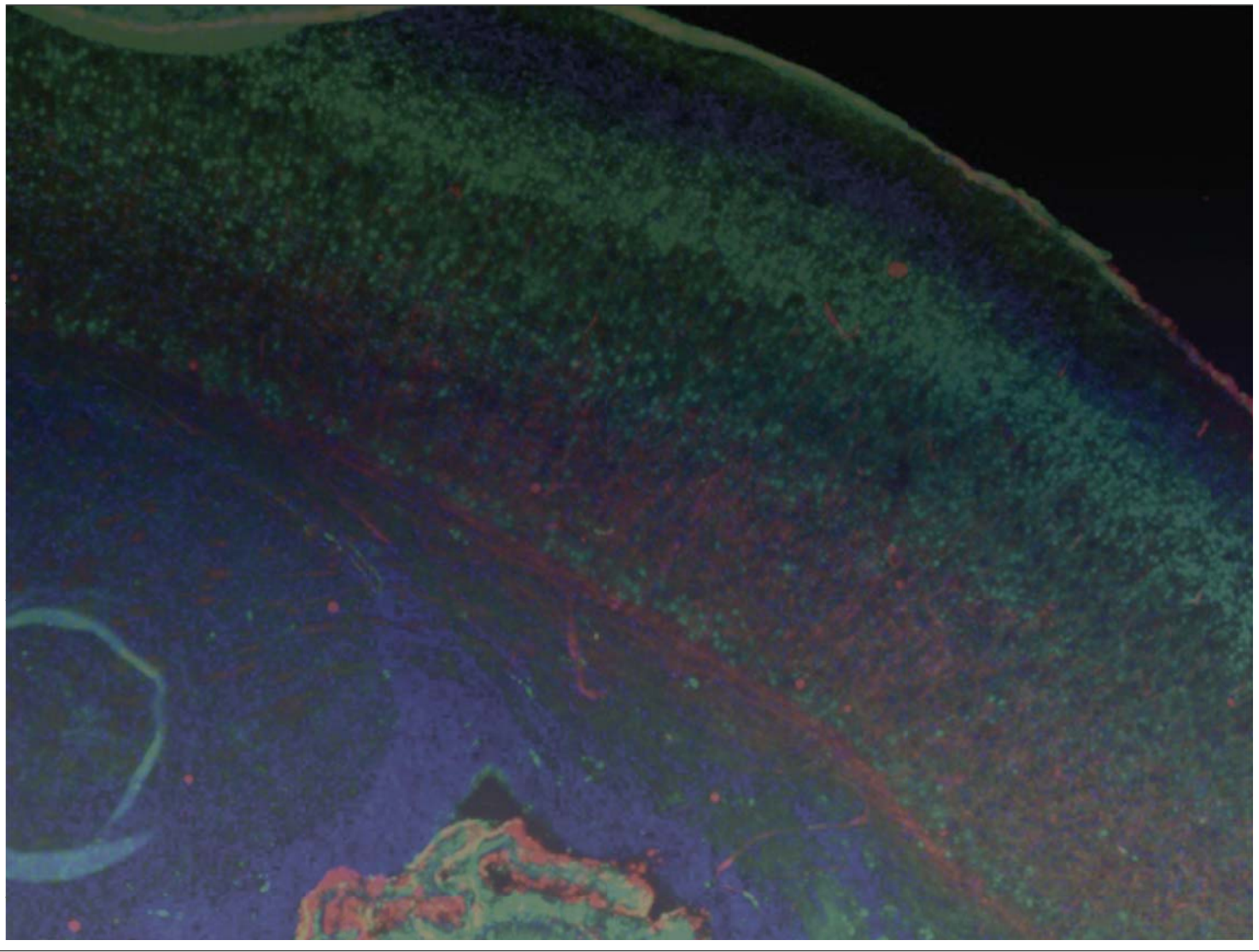
When the world is not enough.

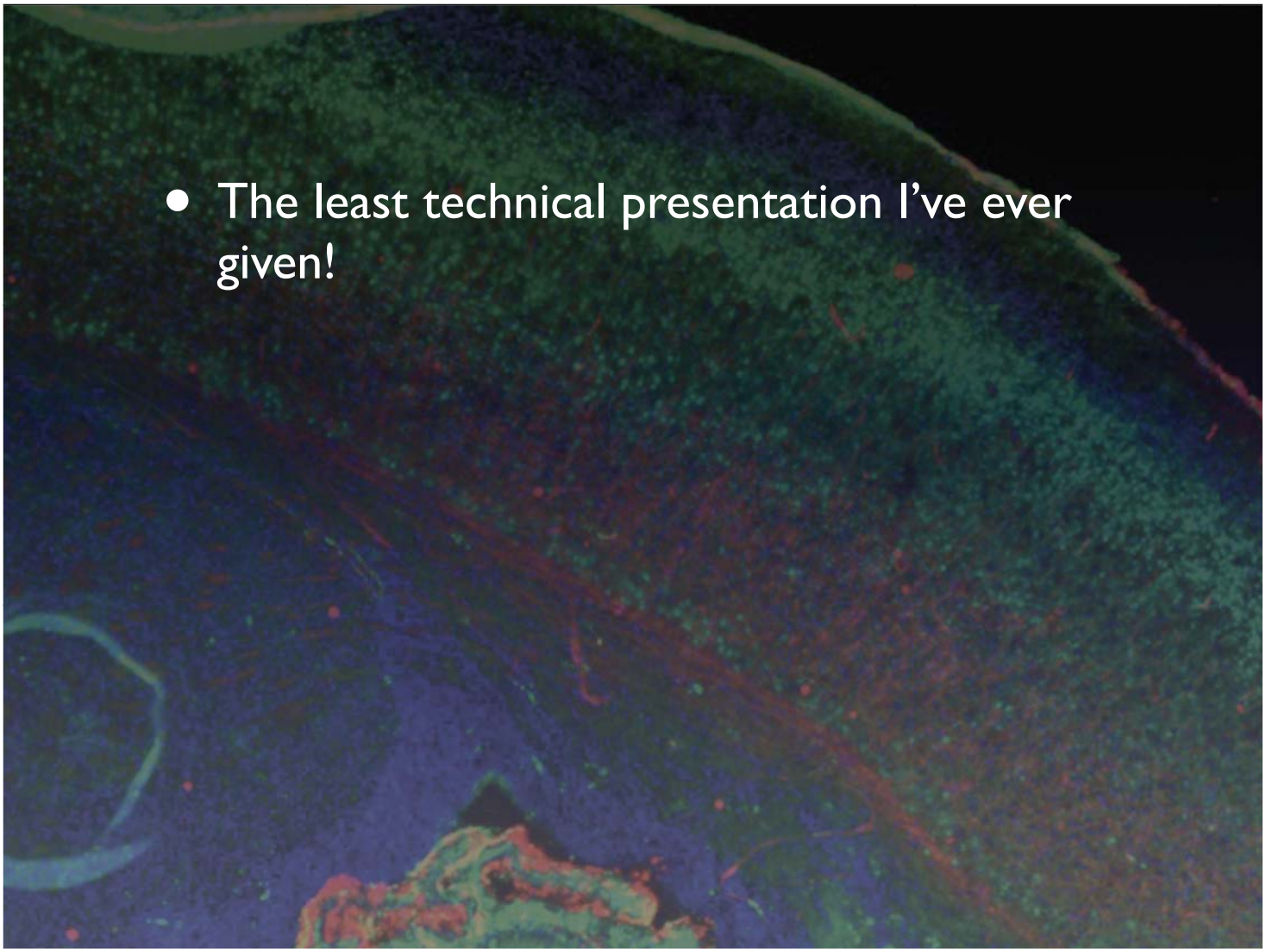
Aka: Not cool enough to use an iPad to present from.
Aka:The mistakes we've made and will continue to make.
Aka:We think we're pretty darn clever, but...

When the world is not enough.

Aka: Not cool enough to use an iPad to present from.
Aka: The mistakes we've made and will continue to make.
Aka: We think we're pretty darn clever, but...

Jake Carroll, Senior IT Manager
The Queensland Brain Institute, UQ



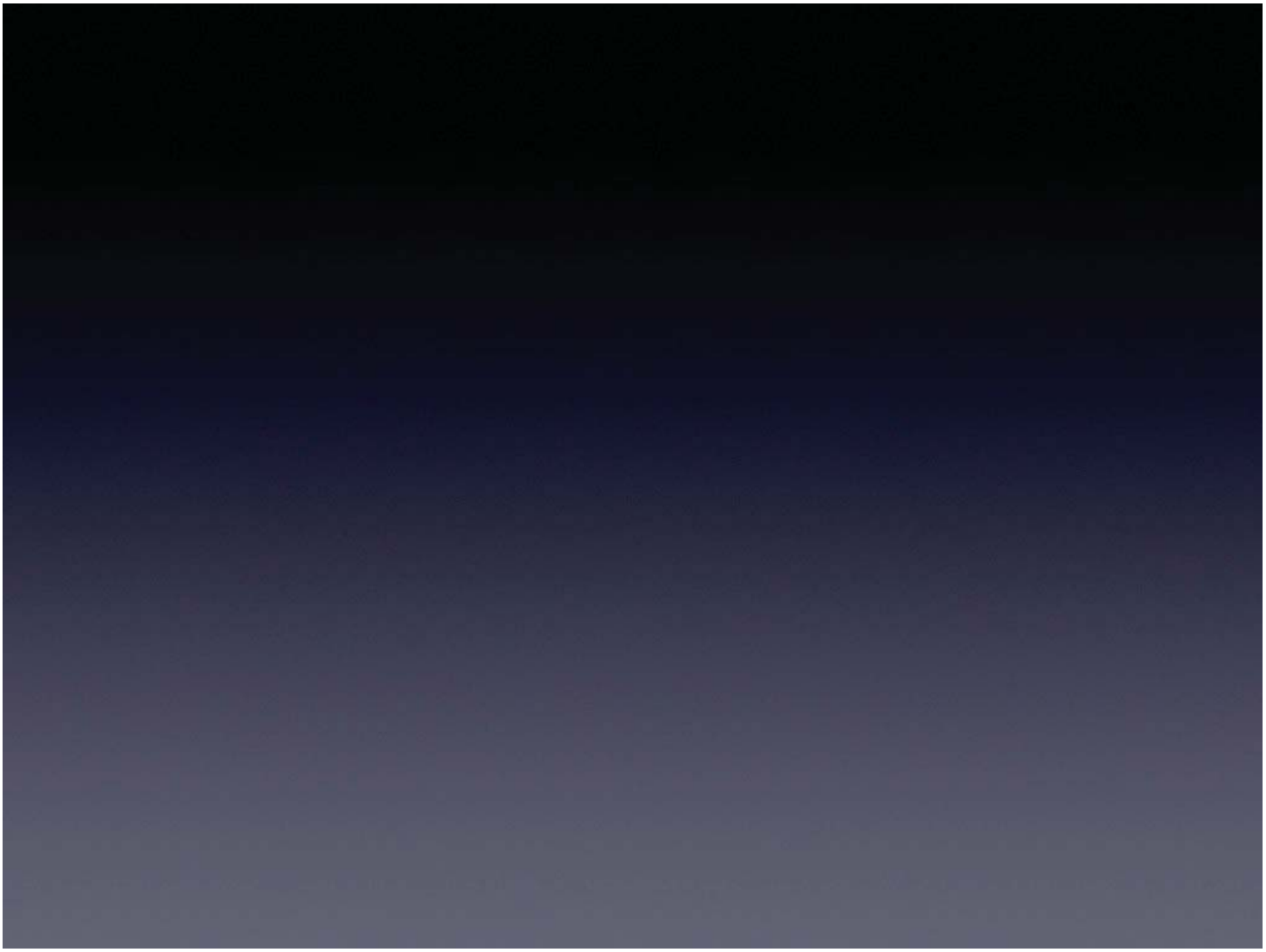

• The least technical presentation I've ever given!


• The least technical presentation I've ever given!
• This isn't about technology in the most blatant sense. This is the story of what we think we know about it, and how fragile what we think we know really is.

- The least technical presentation I've ever given!
- This isn't about technology in the most blatant sense. This is the story of what we think we know about it, and how fragile what we think we know really is.
- I hope it just helps people look at things slightly differently, if nothing else.

- The least technical presentation I've ever given!
- This isn't about technology in the most blatant sense. This is the story of what we think we know about it, and how fragile what we think we know really is.
- I hope it just helps people look at things slightly differently, if nothing else.

- Nothing comfortable about this zone.




True Stories

Episode 1

Episode 1

- A researcher and trusted colleague came to me and said *"I need you to build me a new specialised HPC system"*...

Episode 1

- A researcher and trusted colleague came to me and said *"I need you to build me a new specialised HPC system"*...
- I said: *"OK. Believe it or not, I can do that"*.

Episode 2

Episode 2

- We talked details:

Episode 2

- We talked details:
  - workload types

Episode 2

- We talked details:
  - workload types
  - floating point performance,

Episode 2

- We talked details:
  - workload types
  - floating point performance,
  - tightly coupled vs loosely coupled

Episode 2

- We talked details:
  - workload types
  - floating point performance,
  - tightly coupled vs loosely coupled
  - MPI interconnectivity

Episode 2

- We talked details:
  - workload types
  - floating point performance,
  - tightly coupled vs loosely coupled
  - MPI interconnectivity
  - RDMA latency sensitivity,

Episode 2

- We talked details:
  - workload types
  - floating point performance,
  - tightly coupled vs loosely coupled
  - MPI interconnectivity
  - RDMA latency sensitivity,
  - IBVerbs and all those things I love.

Episode 3

Episode 3

- So we built a whompin' great HPC system with some really niche components.

Episode 3

- So we built a whompin' great HPC system with some really niche components.
- It went like a rocket in 42RU.

Episode 4

Episode 4

- Researcher friend says: *"You are brilliant. This thing makes the last cluster I used look positively tragic. You guys are the best things since sliced bread".*

Episode 4

- Researcher friend says: *"You are brilliant. This thing makes the last cluster I used look positively tragic. You guys are the best things since sliced bread".*
- I respond *"It's what we do, you can do bread slicing in 2RU now!".*

Episode 4

- Researcher friend says: *"You are brilliant. This thing makes the last cluster I used look positively tragic. You guys are the best things since sliced bread".*
- I respond *"It's what we do, you can do bread slicing in 2RU now!".*
- The joke fell flat.

Episode 5

Episode 5

- Researcher: *"Hey man. How come I can't do more than X-arithmetic transform operations on Y-dataset for n-iterations of Z at any given time, and why does it take so much extra time when it reaches time=w?"*

Episode 5

- Researcher: *"Hey man. How come I can't do more than X-arithmetic transform operations on Y-dataset for n-iterations of Z at any given time, and why does it take so much extra time when it reaches time=w?"*
- Me (half awake, at 5AM in the morning, staring at network load graphs): *"Because you've saturated the backplane of that 40GbE switch"*.

He did WHAT?

o_0

He did WHAT?

o_0

He did WHAT?

o_0

...

...

- I had to replay that in my own head. He's flattened 40Gbit/sec of IO for 26 hours straight and the switches simply couldn't signal any quicker.

...

- I had to replay that in my own head. He's flattened 40Gbit/sec of IO for 26 hours straight and the switches simply couldn't signal any quicker.
- Just to simplify that. A guy hauled ~5GB/sec of IO through a cluster chassis for 26 hours and didn't realise he was doing it.

Change gears.

Change gears.

Change gears.

This isn't really a technical problem, honestly.

Change gears.

This isn't really a technical problem, honestly.

Thinking about it more deeply, there are a bunch of concepts that created this situation that are more to do with human behaviour.

Three primitives.

- There are three things I've found that seem unequivocally true in research computing, through experience, metrics and longitudinal study.

Problem 1- Illusions of scale.

Problem 1- Illusions of scale.

- Do you think for a second that a researcher is a researcher because he/she thinks in such a conventional way that they wouldn't try to push something further than it should theoretically go?

Problem 1- Illusions of scale.

- Do you think for a second that a researcher is a researcher because he/she thinks in such a conventional way that they wouldn't try to push something further than it should theoretically go?
- No. Of course not. So why would you expect him/her to not flatten your infrastructure?

- By **very definition** of what is done in cutting edge science, you can't keep up. You can enable, contain and even "make faster", but you can't keep up. Not unless you live in a world of 100% zero compromise and unlimited money.

- By **very definition** of what is done in cutting edge science, you can't keep up. You can enable, contain and even "make faster", but you can't keep up. Not unless you live in a world of 100% zero compromise and unlimited money.
- Many govt. funded projects talk about being an "answer to" the super science proposition.

Problem 2 - Tragedy of commons

- The tragedy of the commons (or tragedy of the unmanaged commons) is a dilemma arising from the situation in which multiple individuals, acting independently and rationally consulting their own self-interest, will ultimately deplete a shared limited resource, even when it is clear that it is not in anyone's long-term interest for this to happen. This dilemma was described in an influential article titled "The Tragedy of the Commons", written by ecologist Garrett Hardin and first published in the journal Science in 1968.

- Think about how a large scale computational cluster works using a grid or job scheduler.

- Think about how a large scale computational cluster works using a grid or job scheduler.
- Think about what happens when a very powerful cluster becomes popular.

- We get ourselves into an inescapable infrastructure battle at this point. Do we centralise and compromise, depleting outcomes, hoping that others won't, so we get a bigger slice of the pie, or do we build our own bridge and cost ourselves the earth and have horrific duplication of services?

Problem 3 - perpetual winners and losers.

Problem 3 - perpetual winners and losers.

- A horrible reality of the nature of funding and grants with regards to technology is that the 'big guys' always win the big funding and thus always end up with the big infrastructure to perpetuate those big grant winning funding runs.

Problem 3 - perpetual winners and losers.

- A horrible reality of the nature of funding and grants with regards to technology is that the 'big guys' always win the big funding and thus always end up with the big infrastructure to perpetuate those big grant winning funding runs.
- Don't kid yourselves, *please*. Equity is an illusion. This is not cynicism but realism.

Another story.

Episode 1

Episode 1

- Researcher: *"So, that new 400TB filesystem you gave us is amazing man!"*

Episode 1

- Researcher: *"So, that new 400TB filesystem you gave us is amazing man!"*
- Me: "It's a pleasure. We aim to please".

Episode 1

- Researcher: *"So, that new 400TB filesystem you gave us is amazing man!"*
- Me: "It's a pleasure. We aim to please".
- Researcher: *"I'm off to save the world, man!"*.

Episode 1

- Researcher: *"So, that new 400TB filesystem you gave us is amazing man!"*
- Me: "It's a pleasure. We aim to please".
- Researcher: *"I'm off to save the world, man!"*.
- Me: "/me eats yellow sponge cake".

Episode 2

Episode 2

- Researcher: *"Hey, directory listings are taking aaaaages man. I need to be on Oslow with results on Wednesday. I've only got 5 days to get this done. Little help?"*

Episode 2

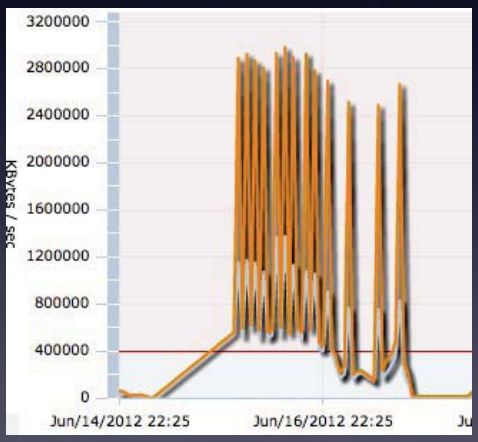
- Researcher: *"Hey, directory listings are taking aaaaages man. I need to be on Oslow with results on Wednesday. I've only got 5 days to get this done. Little help?"*
- Me: *"I'll get the dudes onto it to take a look..."*

Episode 2

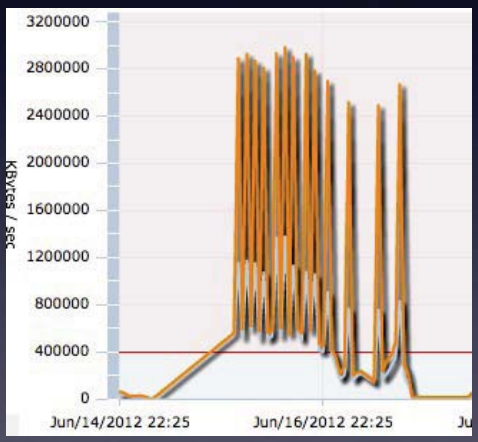
- Researcher: *"Hey, directory listings are taking aaaaages man. I need to be on Oslow with results on Wednesday. I've only got 5 days to get this done. Little help?"*
- Me: *"I'll get the dudes onto it to take a look..."*
- Dudes: *"Problem is one of contention - here are the numbers".*

The numbers.

The numbers.



The numbers.



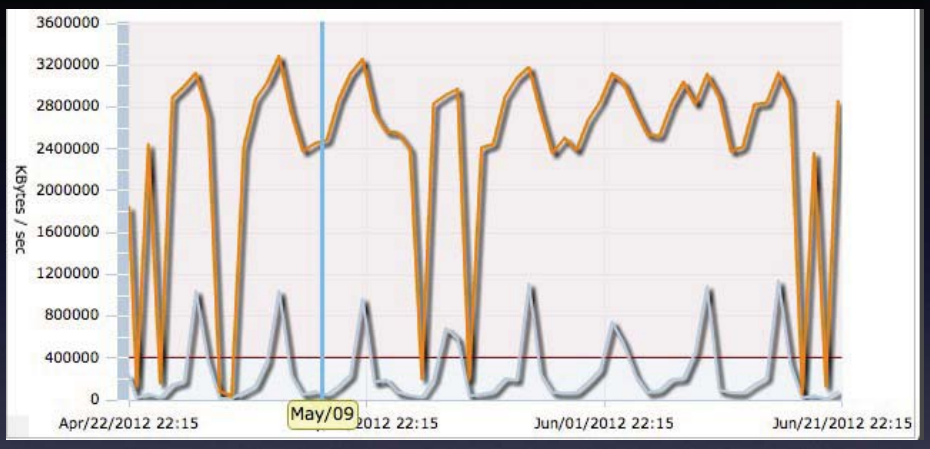
Genome-wide-association-study
in flight, manipulating 1.1TB of data in RAM.

The numbers.

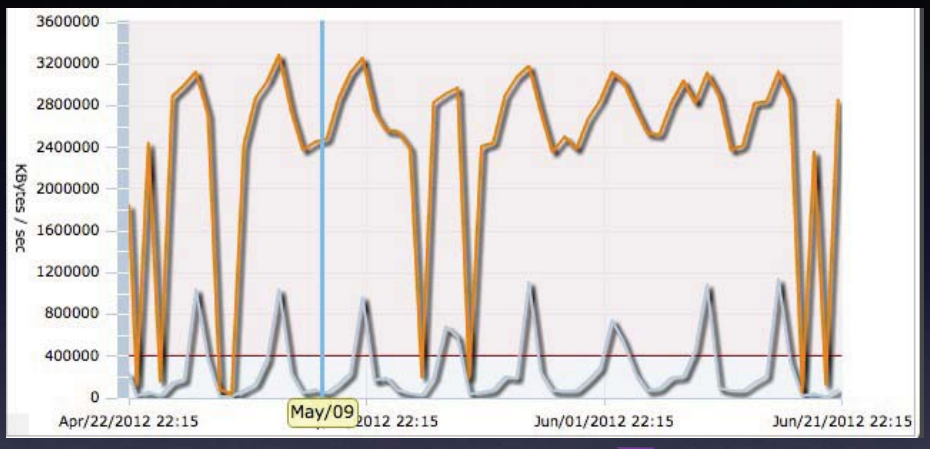
Genome-wide-association-study
in flight, manipulating 1.1TB of data in RAM.

The numbers.

The numbers.



The numbers.



60000 gene data sets,
running a Monte Carlo transform at > 3.25GB/sec

- Me: *"So, we know what's wrong!"*.

- Me: *"So, we know what's wrong!".*
- Researcher: *"CAN YOU FIX IT?!"*

- Me: *"So, we know what's wrong!".*
- Researcher: *"CAN YOU FIX IT?!"*
- Me: *"Remember when you asked for those 11 other people you loosely collaborate with on a far flung shore to have the same ability on the same 400TB filesystem?"*

- Researcher: "*Yeah! It's great!*".

- Researcher: *"Yeah! It's great!"*.
- Me: *"This is your problem. You've allowed too many people to do too much, all at the same time. There is no array here that can sustain that without some performance degradation"*.

- Researcher: *"Yeah! It's great!"*.
- Me: *"This is your problem. You've allowed too many people to do too much, all at the same time. There is no array here that can sustain that without some performance degradation"*.

- Researcher: *"So do we need a bigger one?..."*

- Researcher: *"Yeah! It's great!"*.
- Me: *"This is your problem. You've allowed too many people to do too much, all at the same time. There is no array here that can sustain that without some performance degradation"*.
- Researcher: *"So do we need a bigger one?..."*
- Me: *"You're missing the point, spectacularly"*.

What does it all mean?

What does it all mean?

- It doesn't mean we should all down tools and not try to do what we can.

What does it all mean?

What does it all mean?

- It doesn't mean we should all down tools and not try to do what we can.

What does it all mean?

- It doesn't mean we should all down tools and not try to do what we can.
- It does mean that efforts to 'build it bigger' are misplaced however, and we *really really* seem to be in the habit of building it bigger at the moment, as a state and nation.

What does it all mean?

- It doesn't mean we should all down tools and not try to do what we can.
- It does mean that efforts to 'build it bigger' are misplaced however, and we *really really* seem to be in the habit of building it bigger at the moment, as a state and nation.
- You can think all you want about the 'economy of scale', but it doesn't mean much apart from the bottom line.

Problems.

- Well actually, a really ***wicked*** problem.

What is that?

What is that?

- The problem isn't understood until after the formulation of a solution.

What is that?

- The problem isn't understood until after the formulation of a solution.
- It's got no stopping rule.

What is that?

- The problem isn't understood until after the formulation of a solution.
- It's got no stopping rule.
- Solutions to wicked problems are not right or wrong, but "better" or "worse".

What is that?

- The problem isn't understood until after the formulation of a solution.
- It's got no stopping rule.
- Solutions to wicked problems are not right or wrong, but "better" or "worse".
- Every problem is essentially novel or unique.

What is that?

- The problem isn't understood until after the formulation of a solution.
- It's got no stopping rule.
- Solutions to wicked problems are not right or wrong, but "better" or "worse".
- Every problem is essentially novel or unique.
- Every solution is a "one shot" operation.

What is that?

- The problem isn't understood until after the formulation of a solution.
- It's got no stopping rule.
- Solutions to wicked problems are not right or wrong, but "better" or "worse".
- Every problem is essentially novel or unique.
- Every solution is a "one shot" operation.
- These problems have no given alternative solution.

...for those who think it can be "fixed".

...for those who think it can be “fixed”.

- It can’t. It doesn’t have a solving rule.

- Technology to enable this kind of thing doesn't always have a solution, when it comes under fire.

- Technology to enable this kind of thing doesn't always have a solution, when it comes under fire.
- Vendors can't fix it.

- Technology to enable this kind of thing doesn't always have a solution, when it comes under fire.
- Vendors can't fix it.
- Good management can't fix it. Can mitigate it, but not fix it. Can help people over the line, but not sustainably bring it through orbit.

- Technology to enable this kind of thing doesn't always have a solution, when it comes under fire.
- Vendors can't fix it.
- Good management can't fix it. Can mitigate it, but not fix it. Can help people over the line, but not sustainably bring it through orbit.

But, but, but!

But, but, but!

- Someone is bound to say this:

But, but, but!

- Someone is bound to say this:
- *"Fine then. Just quota them, put a clamp on them, limit them and they will get used to it and as a consequence schedule appropriately!"*

But, but, but!

- Someone is bound to say this:
- *"Fine then. Just quota them, put a clamp on them, limit them and they will get used to it and as a consequence schedule appropriately!"*
- Not so fast, hot shot.


UN

Why?

Why?

- Horrible (and current) as it sounds, you'll make yourself as useless as the NATO/UN guys walking around the streets in Syria.

Why?

- Horrible (and current) as it sounds, you'll make yourself as useless as the NATO/UN guys walking around the streets in Syria.
- To be relevant, we need to enable.

Why?

- Horrible (and current) as it sounds, you'll make yourself as useless as the NATO/UN guys walking around the streets in Syria.
- To be relevant, we need to enable.
- Not walk around looking like we're protecting people, when the reality is, we're powerless to stop/start something.

A sobering example.

A sobering example.

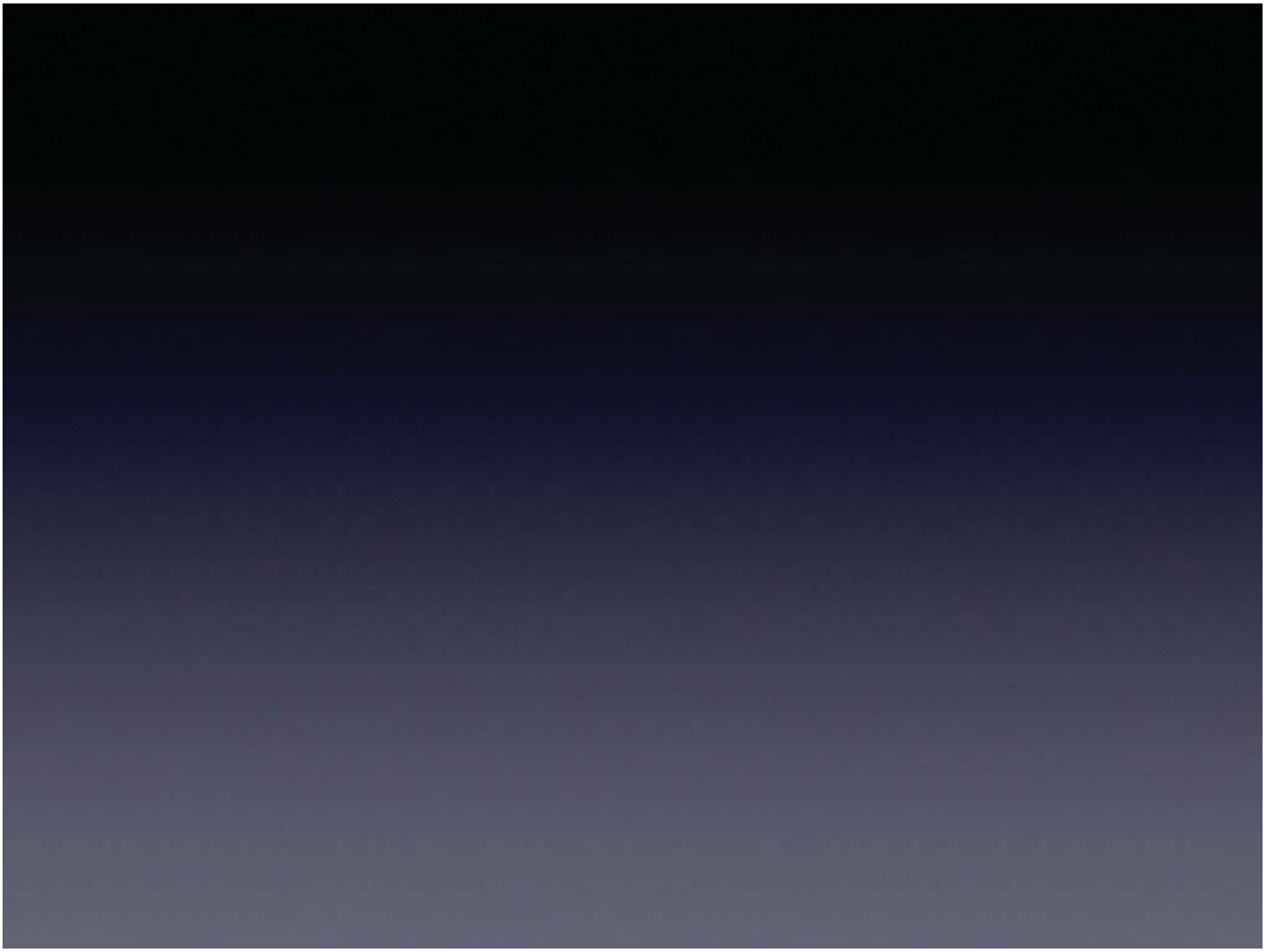
- About to show you a problem for us.

A sobering example.

- About to show you a problem for us.
- If scared of maths, close eyes **now.**

A sobering example.

- About to show you a problem for us.
- If scared of maths, close eyes **now.**
- We're going to jump into the frightening murky world of Quantitative Genomics using Markov chain Monte Carlo for Chromosome Chunking.



```
for (int i = 0; i < input.Length(); i++)
        {
        tokens.ReplaceTokens(input[i]);
        if (tokens.Length() != 2 || tokens[0].SlowCompare("M") != 0)
          {
          ifprintf(output, "%s\n", (const char *) input[i]);
          continue;
          {
          if (marker < oStart || marker > oStop)
          ifprintf(output, "S2 %s\n", (const char *) tokens[1]);
            else
            {
              ifprintf(output, "%s\n", (const char *) input[i]);
              ifprintf(snps, "%s\n", (const char *) tokens[1]);
            }
           {
              if (marker == oStart)
              mStart = tokens[1];
             if (marker == start)
            mFirst = tokens[1];
             if (marker == stop)
            mLast = tokens[1];
           if (marker == oStop && oStop < markers)
            mStop = tokens[1];
        }
       }

        marker++;
        }
```

Why is it nasty?

Why is it nasty?

- In terms of computational complexity, it is a mess and does horrible things to networking and storage.

Why is it nasty?

- In terms of computational complexity, it is a mess and does horrible things to networking and storage.
- $O(\log \log n)$, $O(n)$, $O(n^2)$ *[quadratic]*

Algorithms like this kill things.

Algorithms like this kill things.

- We gave it a 10GbE pipe. It hit 1.25GB/sec.

Algorithms like this kill things.

- We gave it a 10GbE pipe. It hit 1.25GB/sec.
- We gave it a 40GbE pipe. It hit 5GB/sec.

Algorithms like this kill things.

- We gave it a 10GbE pipe. It hit 1.25GB/sec.
- We gave it a 40GbE pipe. It hit 5GB/sec.
- We gave it three aggregate 40GbE pipes and it hit 15GB/sec.

Algorithms like this kill things.

- We gave it a 10GbE pipe. It hit 1.25GB/sec.
- We gave it a 40GbE pipe. It hit 5GB/sec.
- We gave it three aggregate 40GbE pipes and it hit 15GB/sec.
- If we had the resources to give it 10 * 40GbE pipes, it would hit 50GB/sec.

No easy answers.

No easy answers.

- There are no easy answers, nor are there things to “solve” here.

No easy answers.

- There are no easy answers, nor are there things to "solve" here.
- There is one concept that might *actually* help users get done what they need to, but it's far from perfect.

Autonomic computing.

Autonomic computing.

- The idea is designed to address rapidly growing complexity.

Autonomic computing.

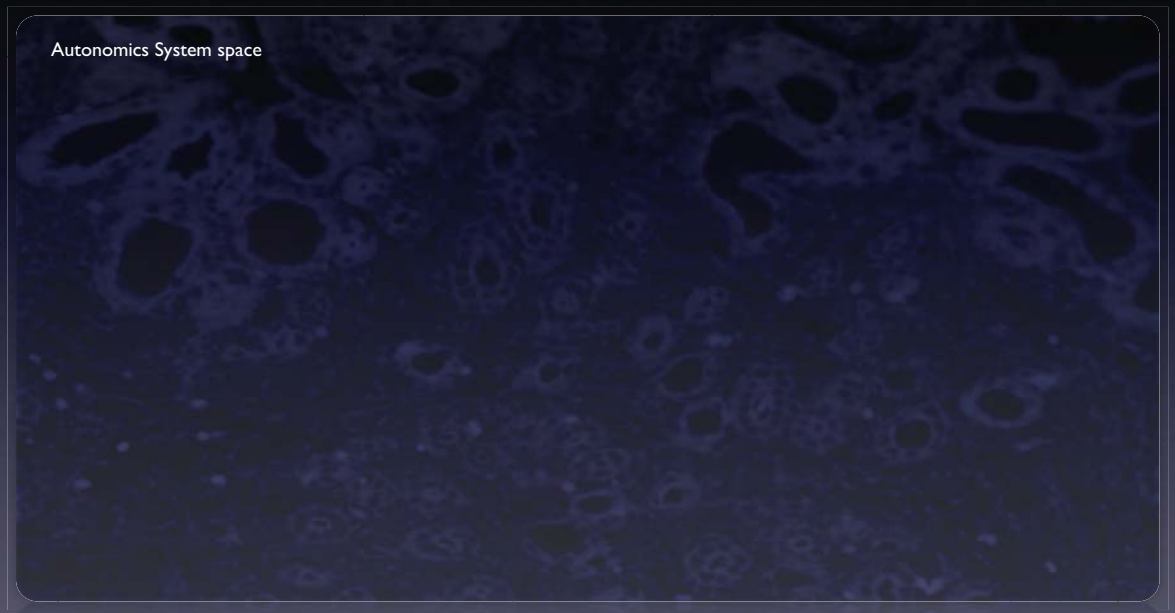
- The idea is designed to address rapidly growing complexity.
- Self-managing characteristics of distributed computing resources, networks, communications channels to deal with unpredictable change, chaos and load whilst hiding complexity from end users.

Autonomic computing.

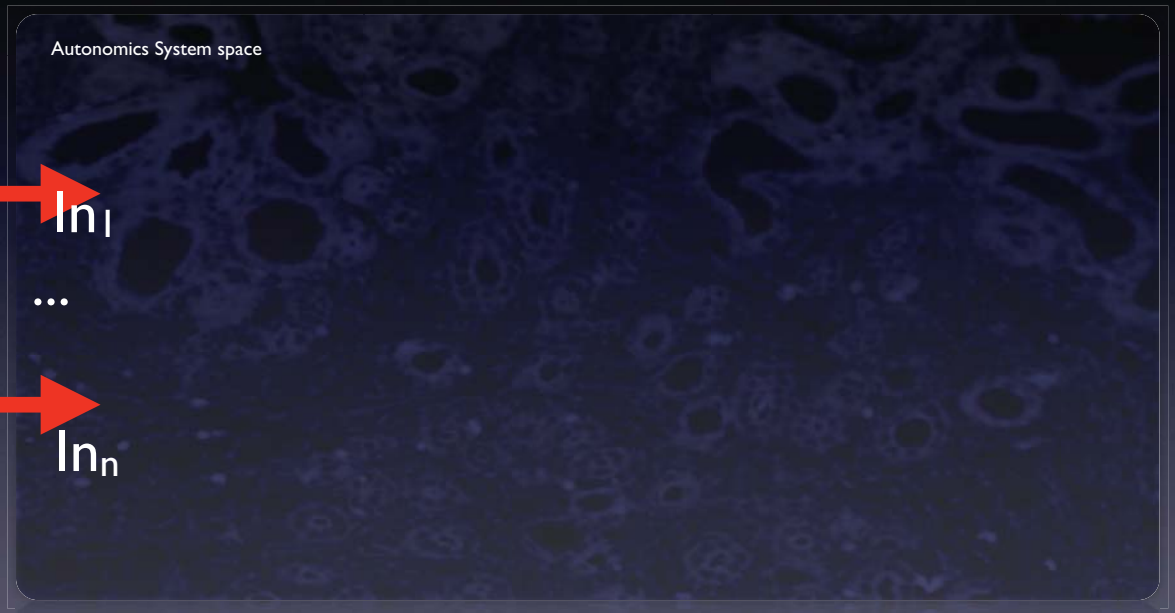
- The idea is designed to address rapidly growing complexity.
- Self-managing characteristics of distributed computing resources, networks, communications channels to deal with unpredictable change, chaos and load whilst hiding complexity from end users.
- The idea came from IBM's labs in 2001.

Conceptual model

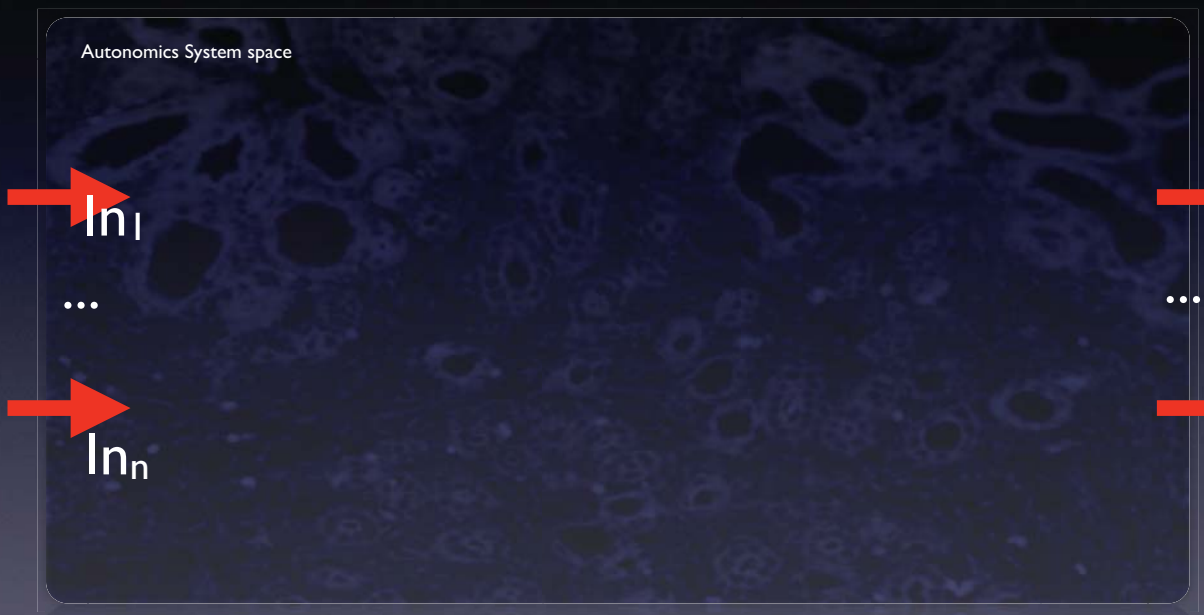
Conceptual model



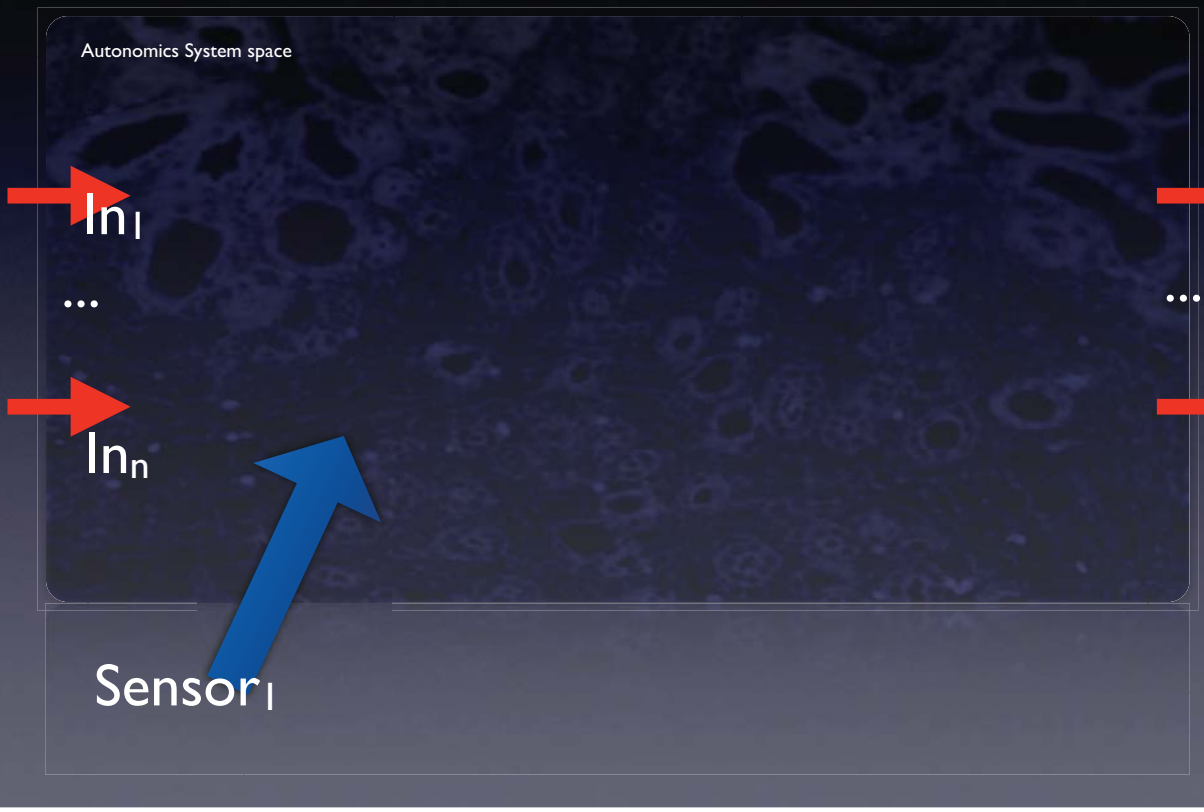
Conceptual model



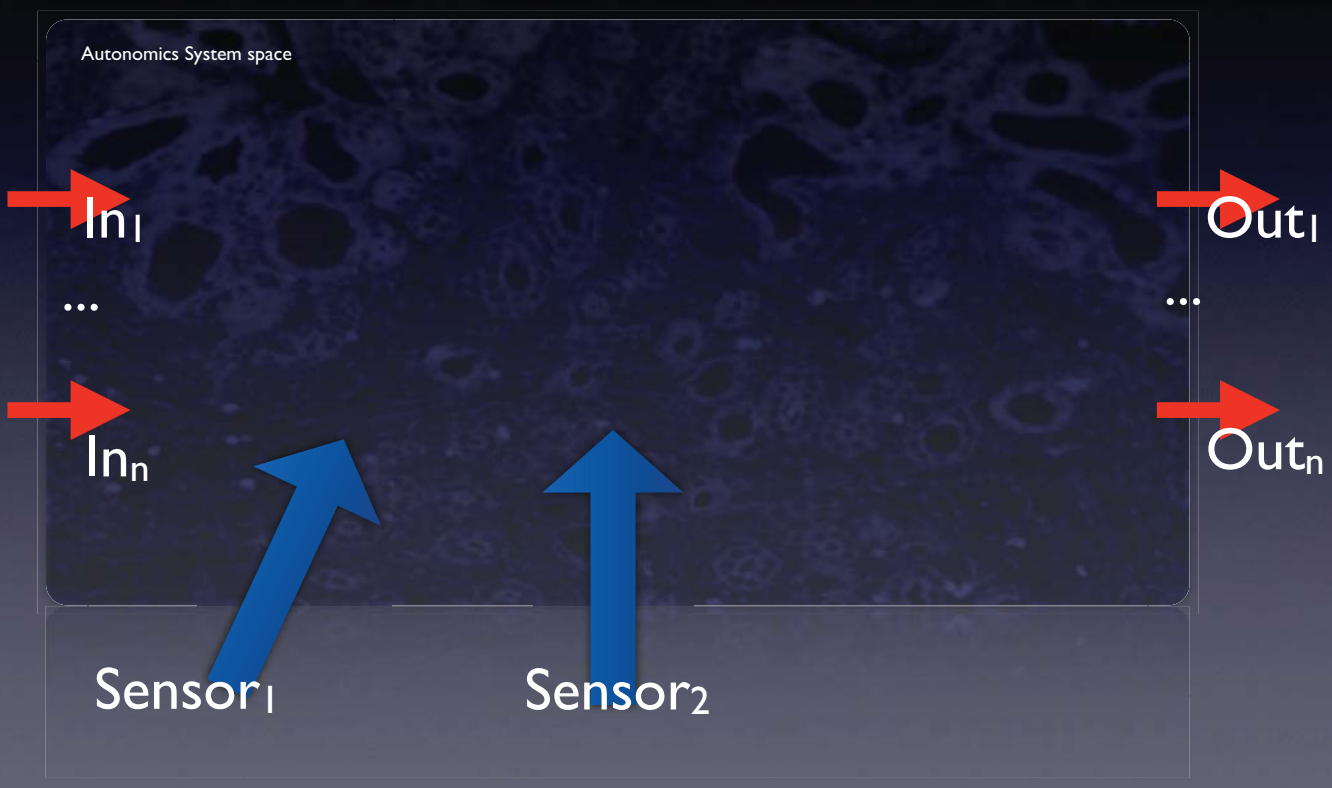
Conceptual model



Conceptual model

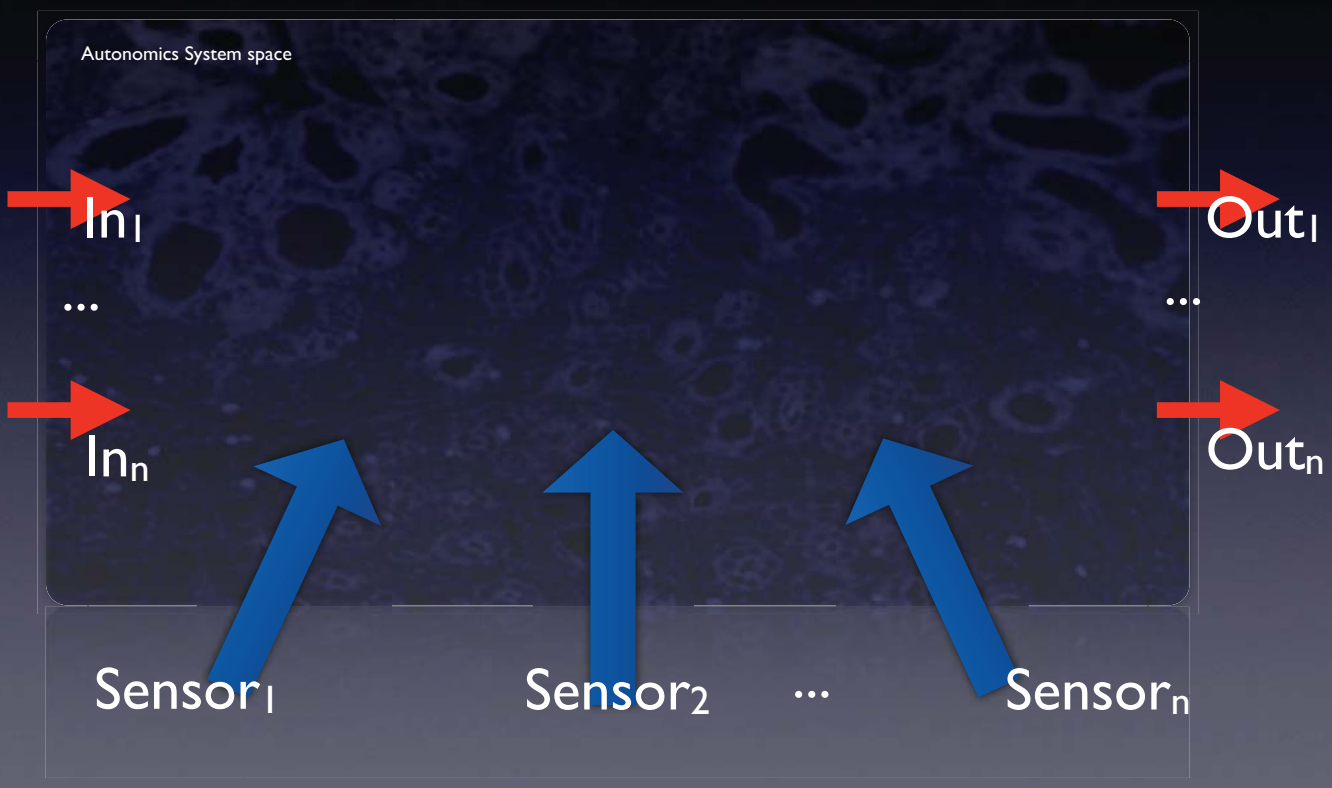


Conceptual model

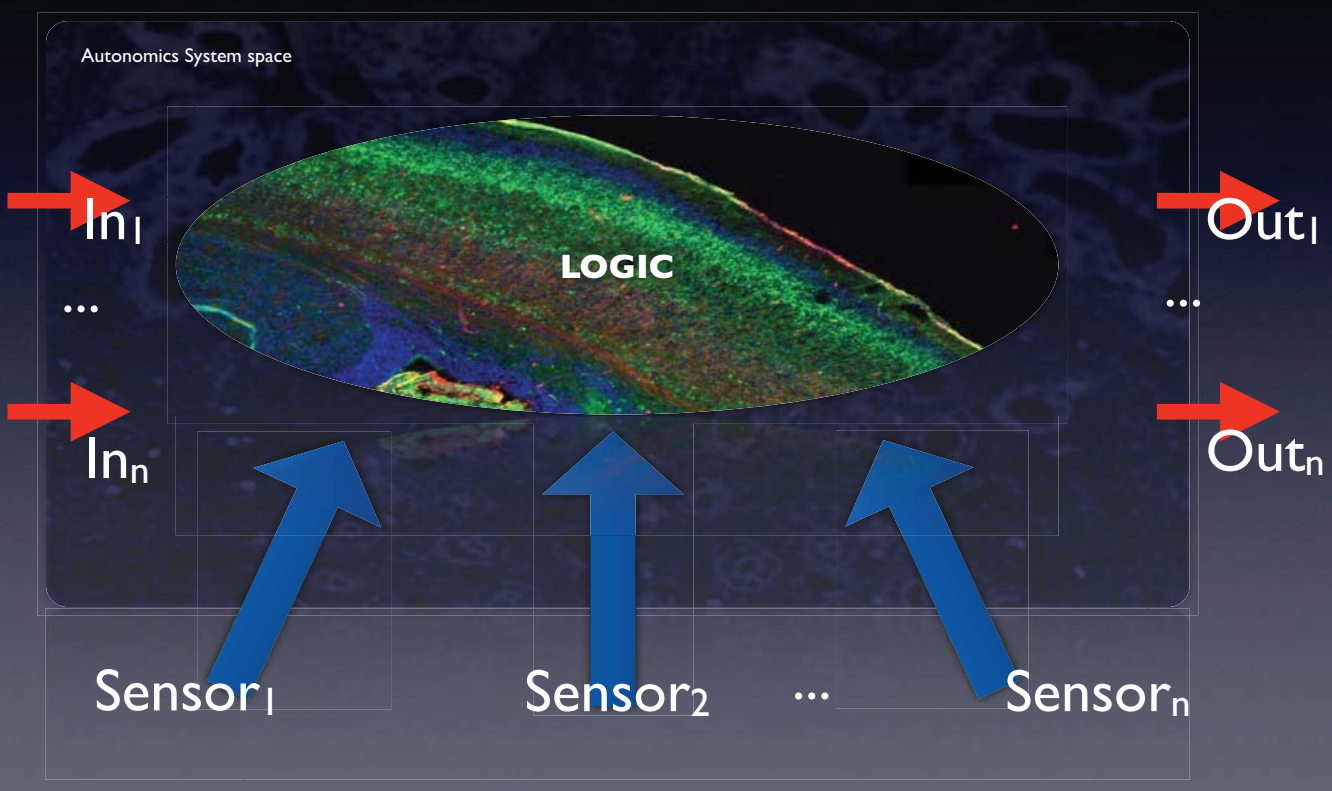


Conceptual model

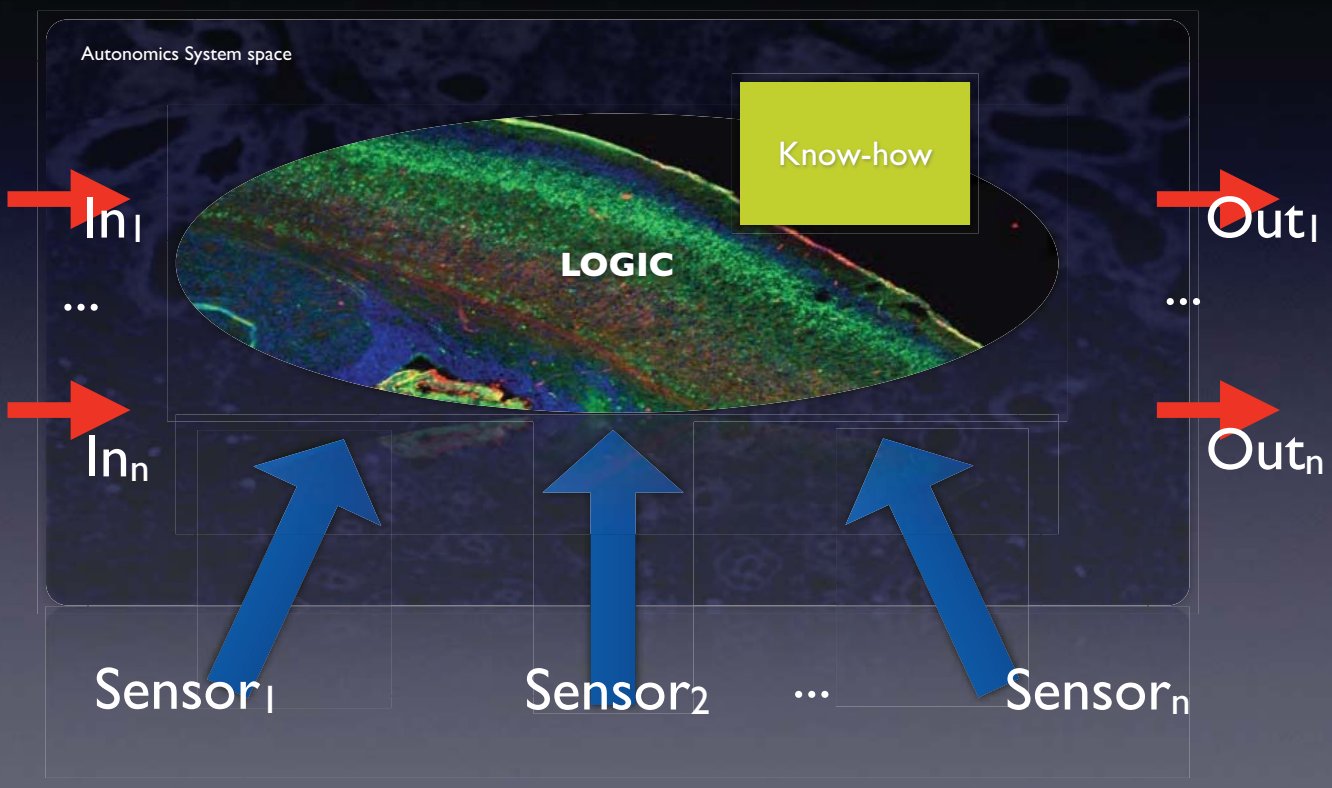
Conceptual model



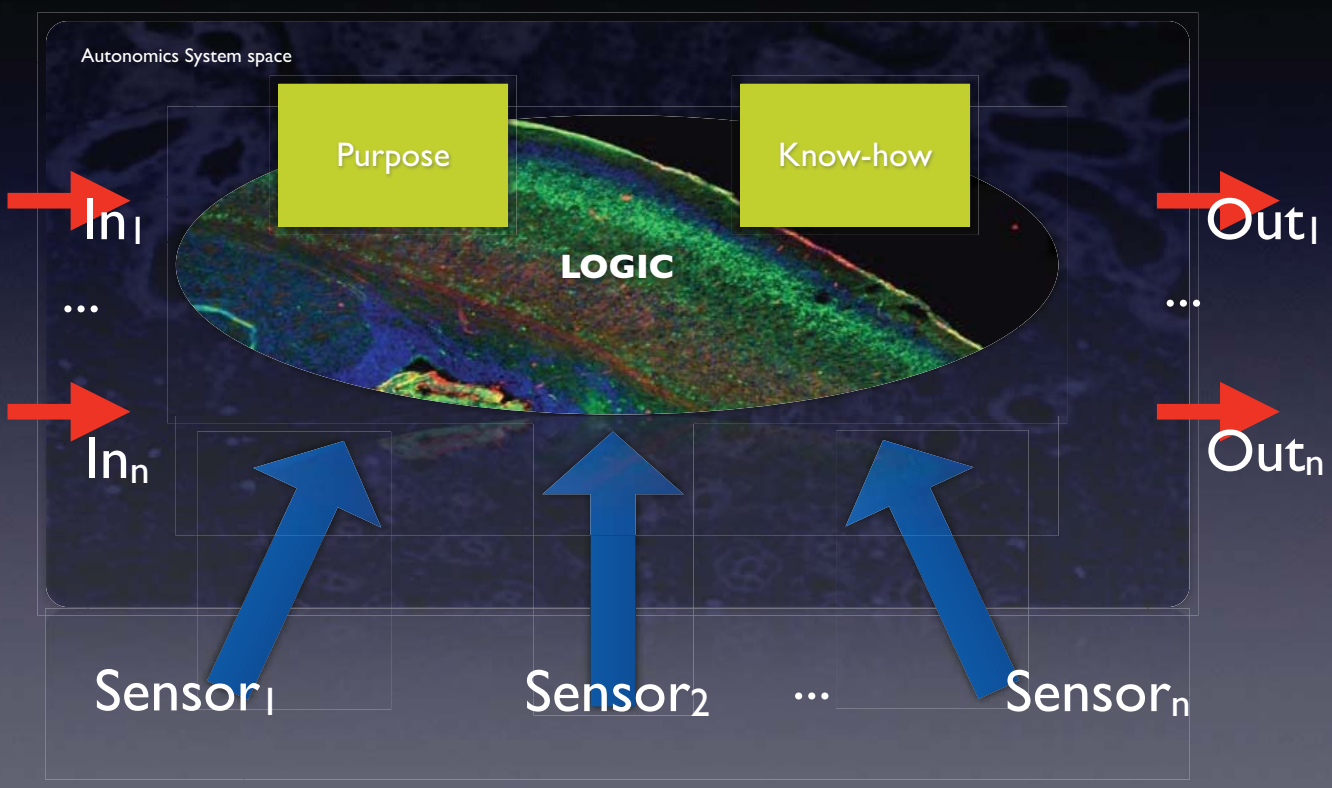
Conceptual model



Conceptual model



Conceptual model



How it works.

How it works.

- Imagine some world event occurs that smashes the social media server farms, or you've just discovered something massive in high-energy physics that needs big computation time.

How it works.

- Imagine some world event occurs that smashes the social media server farms, or you've just discovered something massive in high-energy physics that needs big computation time.
- A significant impact occurs on the infrastructure that is going to "work" on it.

How it works.

- Imagine some world event occurs that smashes the social media server farms, or you've just discovered something massive in high-energy physics that needs big computation time.
- A significant impact occurs on the infrastructure that is going to "work" on it.
- Autonomics should take over here.

- The information from Sensors feed the system so that it understands the constraints and conditions “outside” that it has to deal with in context.

- The information from Sensors feed the system so that it understands the constraints and conditions "outside" that it has to deal with in context.
- In this respect, when certain conditions are taking place, the whole world doesn't melt when one person puts up their hand to do something "big".

Complex.

Complex.

- We can talk all we wish about ubiquitous communications and scale of technology/ clusters/compute etc, but to do what we've proposed back here takes things we don't have yet.

Complex.

- We can talk all we wish about ubiquitous communications and scale of technology/ clusters/compute etc, but to do what we've proposed back here takes things we don't have yet.
- It takes a *far* more clear picture and global overview than what we've currently got.

We've started. Long way to go, though!

We've started. Long way to go, though!

- Our internal HPC schedulers are now at least partially "external influence" aware. We created SGE complexes to take into account network conditions, external processing entities and interactions taking place between things *well outside* of the cluster and compute resource.

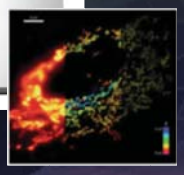
Real time computing.

Real time computing.

Real time computing.

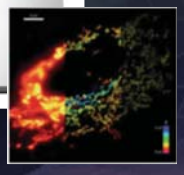
Real time computing.

Real time computing.



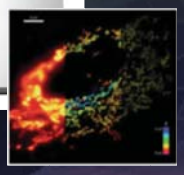
LIMS

Real time computing.



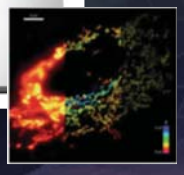
LIMS

Real time computing.



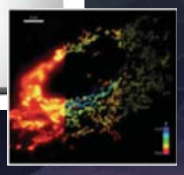
LIMS

Real time computing.

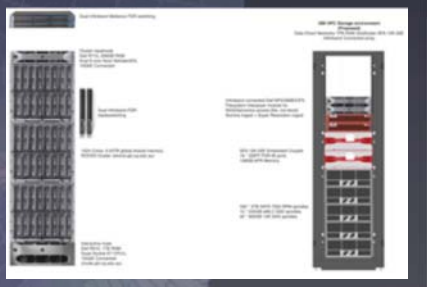


LIMS

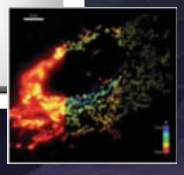
Real time computing.



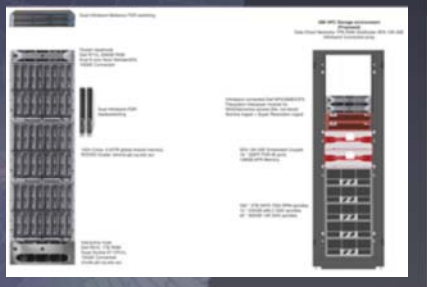
LIMS



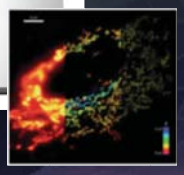
Real time computing.



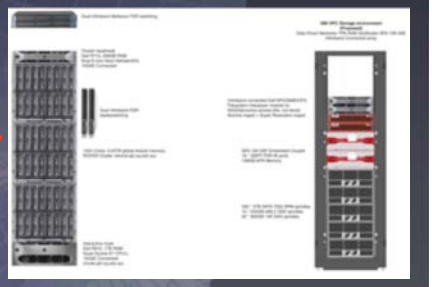
LIMS



Real time computing.

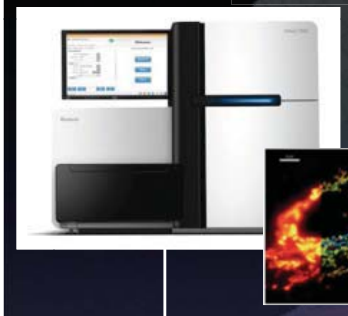


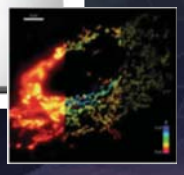
LIMS



SGE$_{queueComplex}$

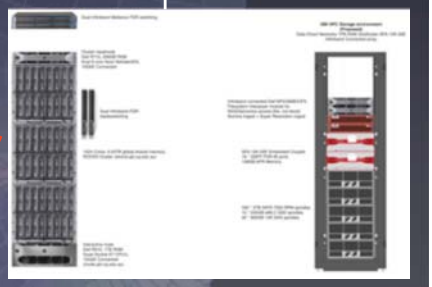
Real time computing.





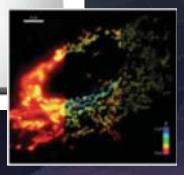
LIMS

Sensor Network



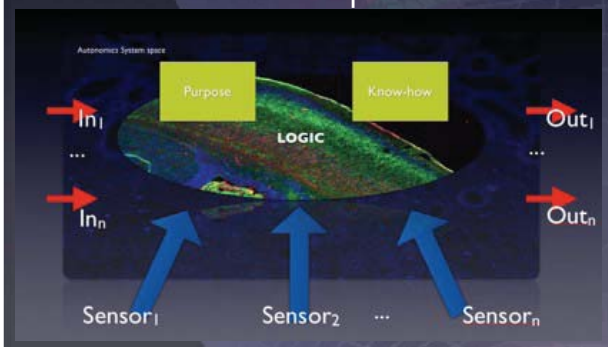
SGE$_{queueComplex}$

Real time computing.

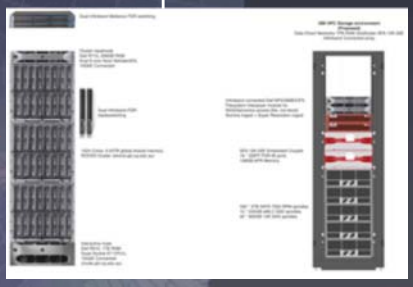


LIMS

Sensor Network



SGE queueComplex



...

...

- We found out, the hard way, that production bioinformatics and super-resolution is 'sensitive' if you've got serious multi-tenancy environments.

...

- We found out, the hard way, that production bioinformatics and super-resolution is 'sensitive' if you've got serious multi-tenancy environments.
- To that end, you need feed back loops and application awareness/interconnectedness to do it correctly at all with satisfactory outcomes.

...

- We found out, the hard way, that production bioinformatics and super-resolution is 'sensitive' if you've got serious multi-tenancy environments.
- To that end, you need feed back loops and application awareness/interconnectedness to do it correctly at all with satisfactory outcomes.
- It might not work *en-masse*.

...

...

- SGE needs Queue Complex that takes inputs from sensors sending SNMP data from network switching/routing hardware, such that SGE can make decisions about what I/O to put where and on what ports.

...

- SGE needs Queue Complex that takes inputs from sensors sending SNMP data from network switching/routing hardware, such that SGE can make decisions about what I/O to put where and on what ports.
- Sequencer box needs inputs from HPC/ Cluster and LIMS to determine what it needs 'next' to carry on a sequence.

Future.

Future.

- Complete application awareness and interconnectedness is *next*.

Future.

- Complete application awareness and interconnectedness is *next*.
- Part of it is here and now.

Future.

- Complete application awareness and interconnectedness is *next*.
- Part of it is here and now.
- Part of it, we can't quite fathom yet.

The world is not enough.

...unless you change the world.

Thank you all, sincerely.
You've been a wonderful audience.

See you again, one day...maybe.
--JC