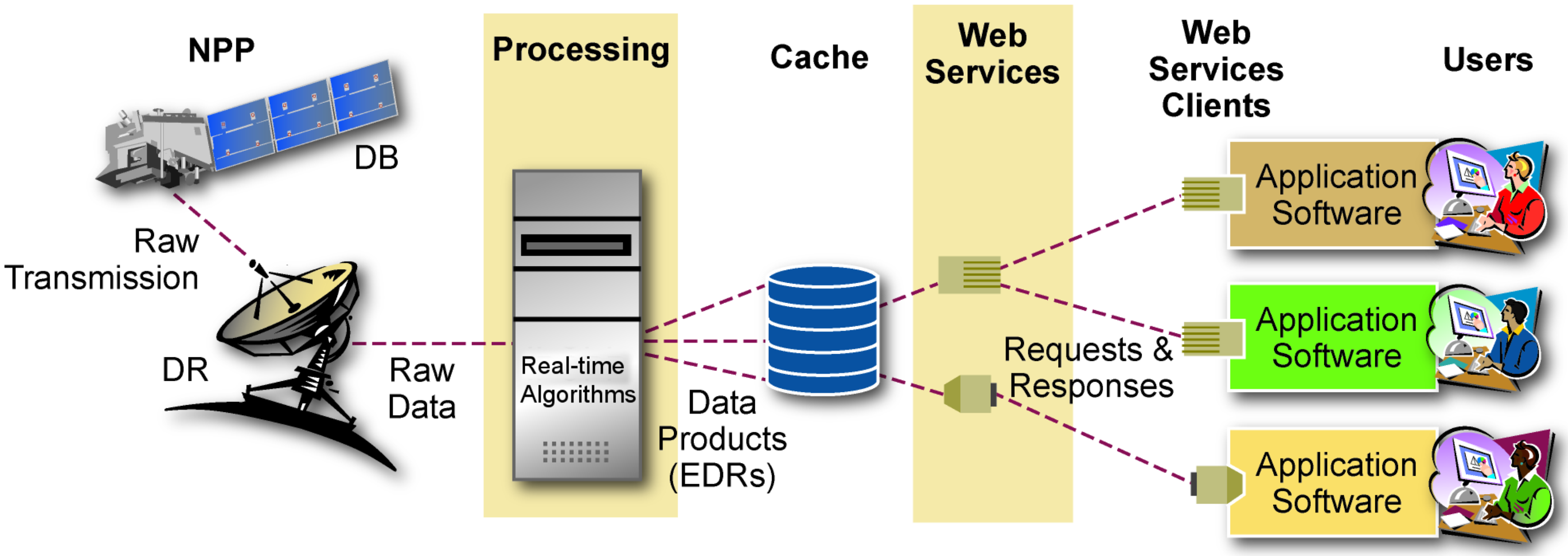
Systems and Services For Near-Real-Time Web Access to NPP data

John Evans
Eduardo Valente
Samir Chettri

Global Science & Technology, Inc. (GST)
Greenbelt, MD

Case study: Overview of near real-time workflow



Phase I: Data streams

- *Gearing up:* Aqua MODIS Direct Broadcast
  - Using data and software from NASA's Direct Readout Laboratory (DRL)

  http://directreadout.sci.gsfc.nasa.gov♪

  - *In particular:* International Polar Orbiter Processing Package (IPOPP)

- *Target:* NPP (precursor to JPSS) Direct Broadcast
  - NPP scheduled for Fall 2011 launch
  - NASA/NOAA ground system (SafetyNet TM) still in progress
    - This makes Direct Broadcast a vital alternative

Phase I: Use Case

- Working with NASA's Short-term Prediction Research and Transition program

  http://weather.msfc.nasa.gov/sport♪

Phase I

- Investigated information and processing technologies to provide near-real-time Web-based access to NPP/JPSS satellite data.

- Focus on computational h/w and s/w for serving Direct Broadcast data (and other near-real-time products) to modelers, forecasters and decision makers.

- For time sensitive applications, Direct Broadcast/Direct Readout will be the only way for access to NPP or JPSS-1 data (until JPSS SafetyNet ™ is completed.)

Phase I: Experiments with cloud computing

- In-house proof-of-concept quickly showed that
  - We would need a lot more bandwidth
  - We *might* need a full-blown data center for peak loads
  - Ensuring real-time throughput cost-effectively is hard
- We found that cloud computing provided:
  - Increased bandwidth and computing capacity
  - Configuration flexibility
  - Easy transition (Infrastructure as a Service (IaaS): Amazon EC2, NASA Nebula)

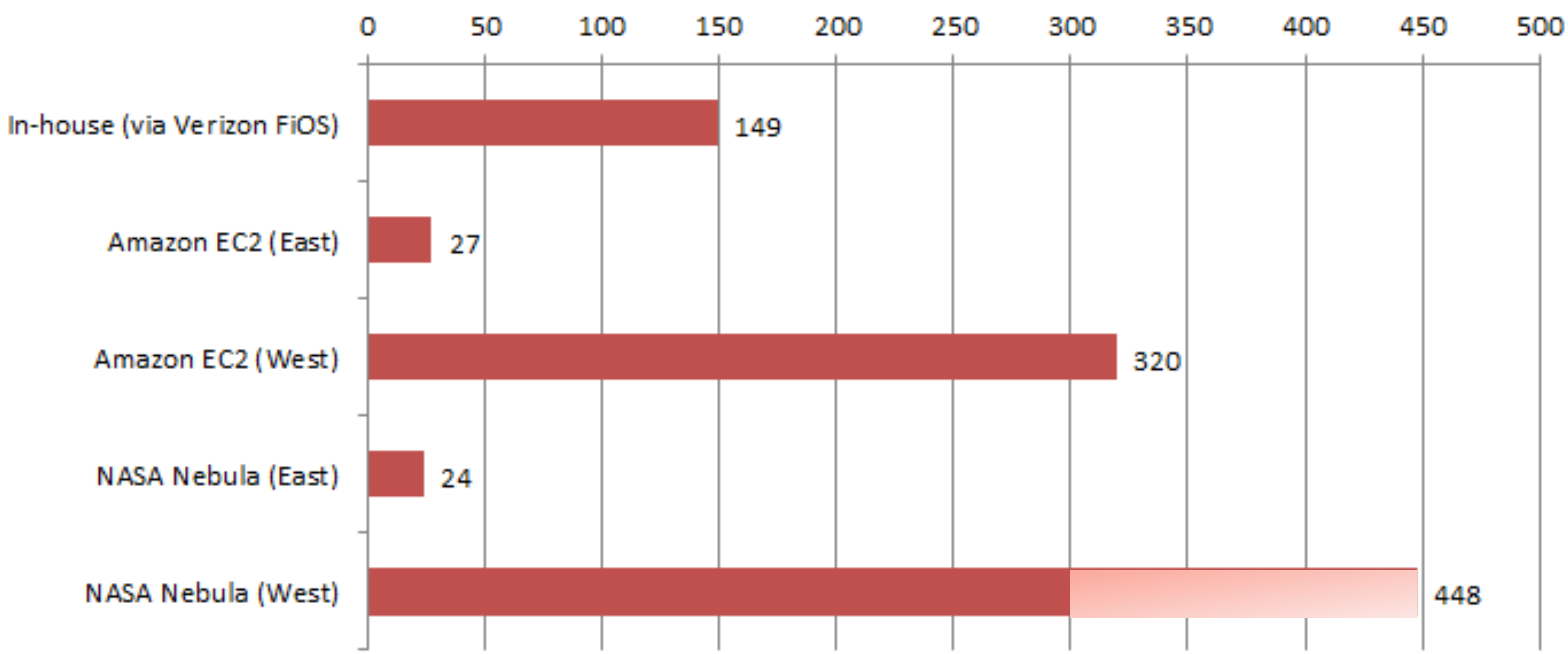
Phase I: Questions raised

- Could we sustain near-real-time processing of NPP data?
- Would it be worth (**$$**) doing this in the cloud?

We obtained partial answers to these questions within Phase I and investigated them further after Phase I as an internal R&D project.

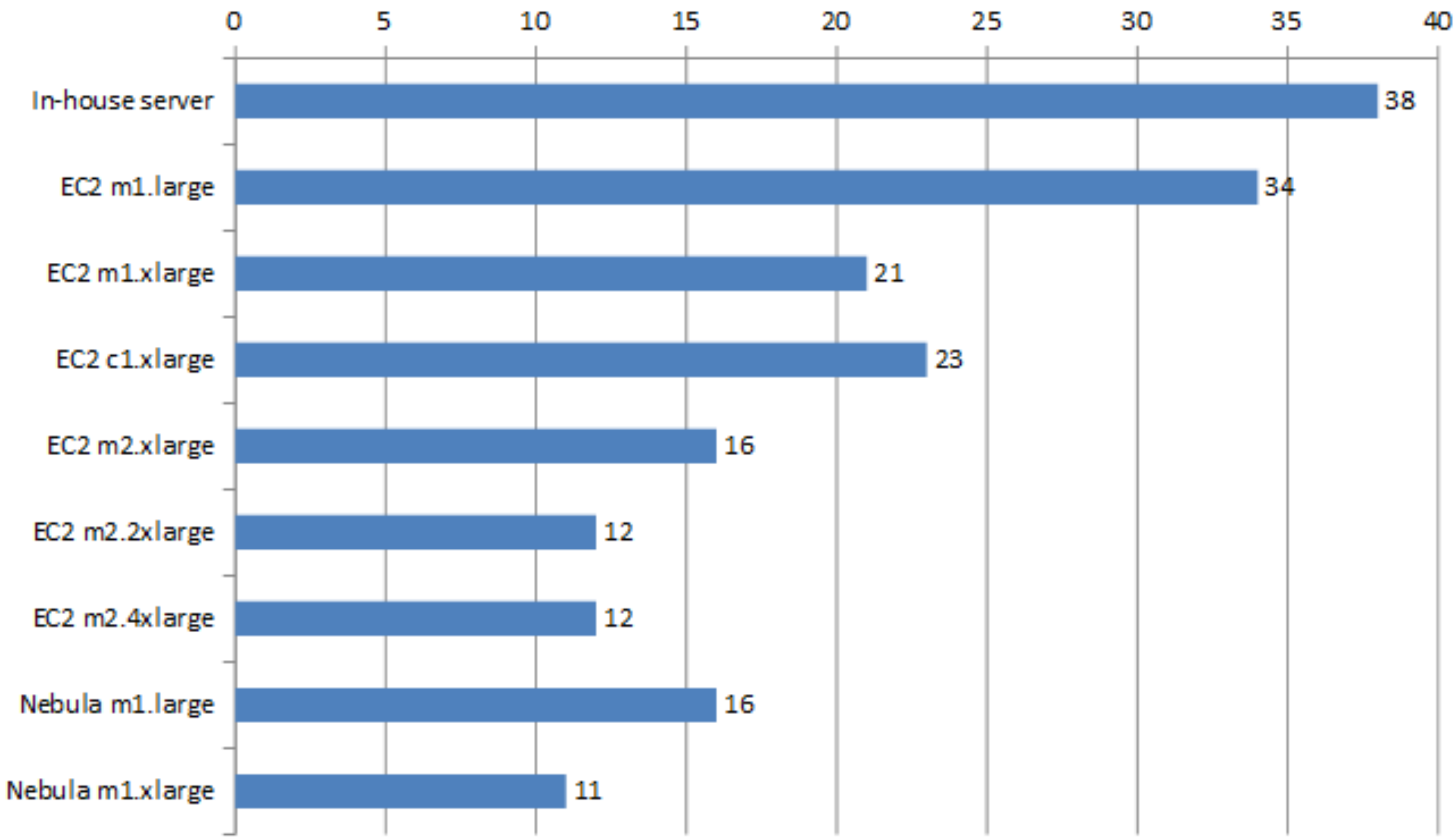
Post-Phase I – Pre-Phase II findings: Bandwidth

Time (in seconds) for transferring largest (840MB) input images from receiver (DRL)



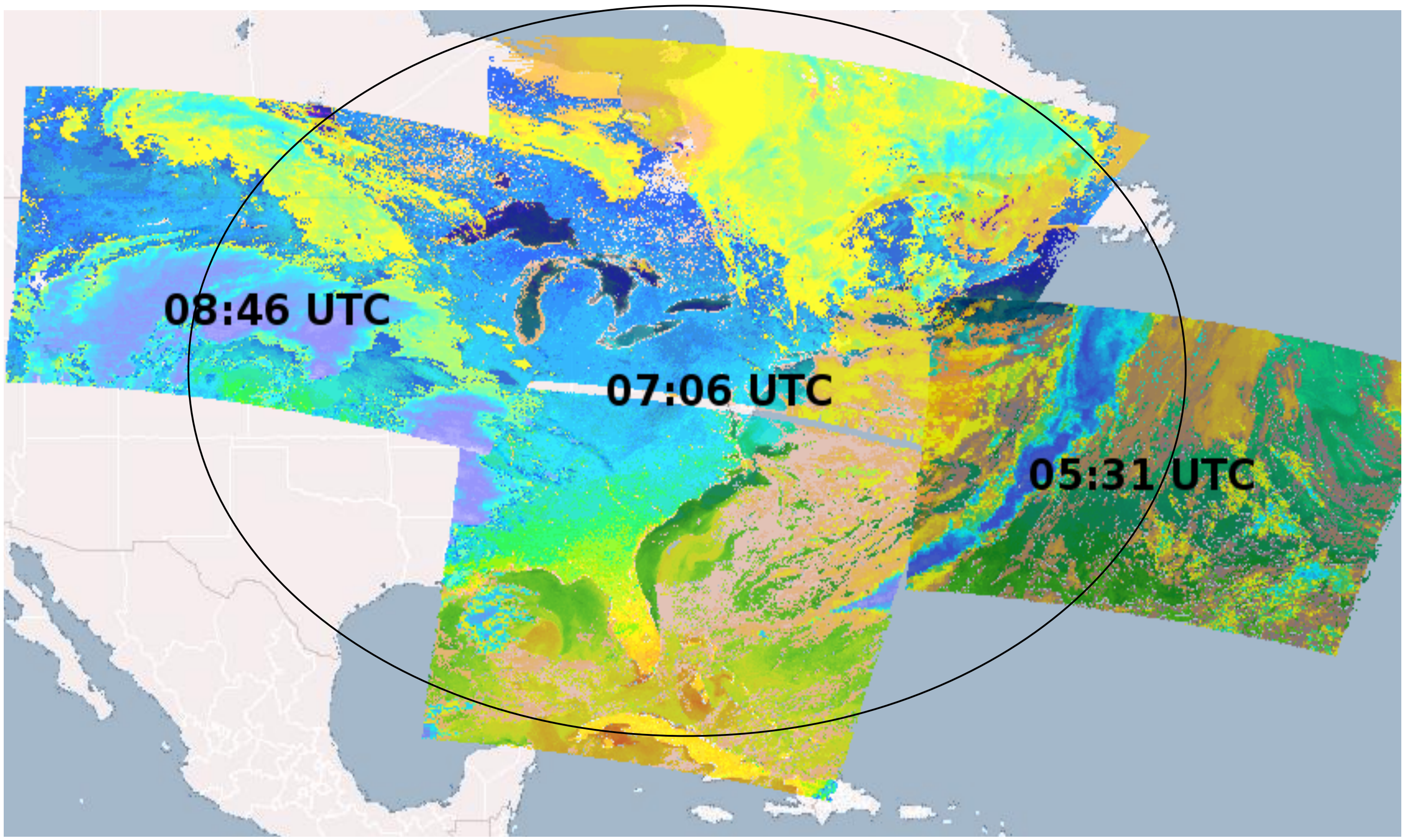
Post-Phase I – Pre-Phase II findings: Processing latency

Processing latency (in minutes) for last product derived from largest (840MB) input images



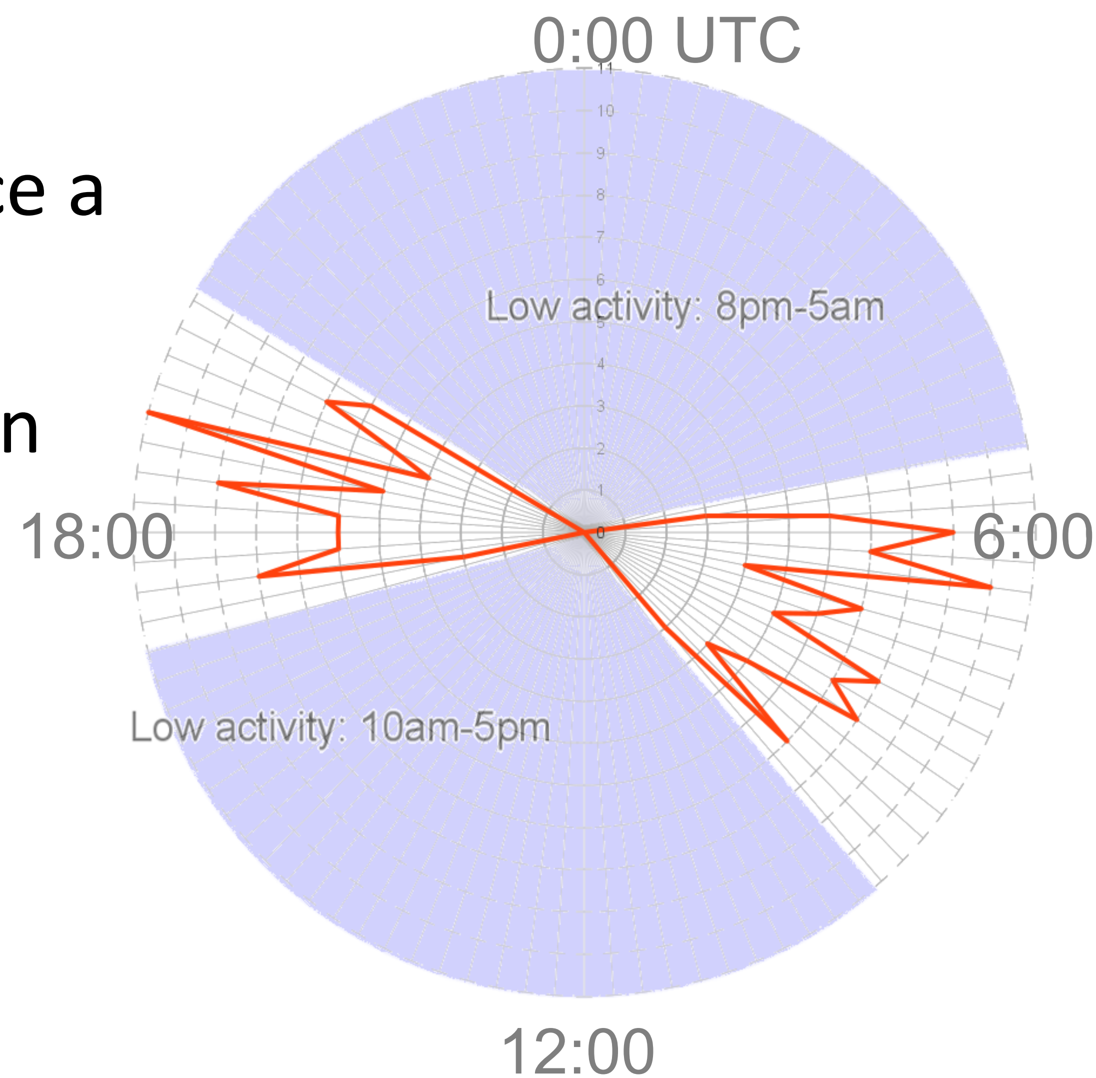
Post-Phase I – Pre-Phase II findings: Costs

One morning's Direct Broadcast from NASA's Aqua satellite to a receiving station in Greenbelt, MD.



Post-Phase I – Pre-Phase II findings: Costs

- 2-3 overpasses, twice a day
- Scale down between overpasses
- Scale up just in time for next overpass



Phase II Technical Objectives

- Engineer Phase I working proof-of-concept into a **robust, adaptable, secure** set of components.
- Equip Web services with industry standard interfaces that permit **access by a variety of end-user tools**.
- **Generalize** from Aqua/Terra + NPP to other data streams
- **Streamline performance** to ensure real-time throughput to many users
- **Outreach** to users and developers in target markets (weather forecasting and others)

Objectives: 1

Engineer Phase I working proof-of-concept into a robust, adaptable and secure set of components.

Moved from an investigation of the use of the cloud to making cloud computing a key element in our investigations.

Adaptable:

- Data input - easily add new data streams.
- Data output – easily accommodate new users and applications.

Objectives: 2

Equip Web services with industry standard interfaces that permit access by a variety of end-user tools.

- OGC Web Services suite (WMS) + KML
- OGC Web Feature Service *(points, lines, polygons)*
- OGC Web Coverage Service *for grid data and imagery*
- OPeNDAP *for scientific data*
- OGC Sensor Observation Service (SOS) for *in situ* or orbital sensor data

First use case scenario: Provide data in a form amenable to AWIPS II for WFO at MSFC

Objectives 3

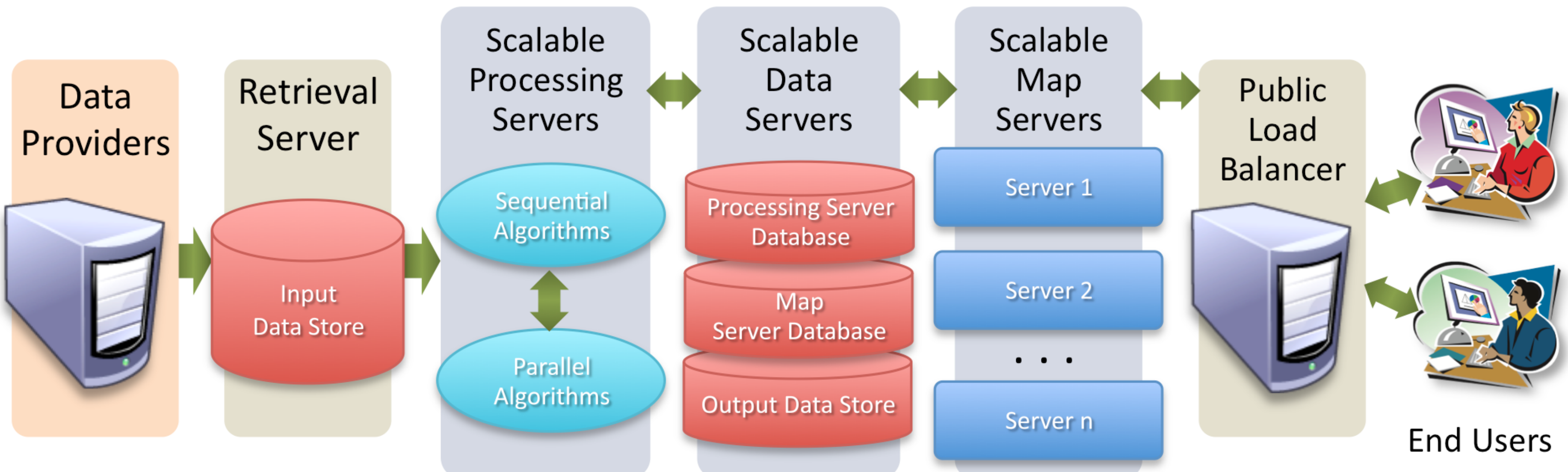
Generalize from Aqua/Terra + NPP to other data streams

- GOES
- *In situ* weather stations (National Mesonet)
- AERONET
- NEXRAD
- Simulation outputs
- Others such as LIDAR, GOES-R

Objectives 4

Streamline performance to ensure real-time throughput to many users

Scale components up and down as needed



Objectives: 5

Outreach to users and developers in target markets (weather forecasting and others)

- Gather alpha and beta users of our Phase 2 s/w and conduct a survey (ease of use, appropriateness)
- Investigate opportunities to partner with SPoRT.

Commercialization opportunities

- Software as a Service:
  - Build and host Web-based applications for specific user communities
- Platform as a Service:
  - Host data services, relying on real-time data providers
- Infrastructure as a Service:
  - Maintain / distribute machine image(s) for hosting real-time data services

"Elevator speech"

- The project has immediate relevance because of the delay in the deployment of SafetyNet.
- Collaboration with SPoRT links us directly to a customer for our products.
- Cloud Computing puts all of these opportunities within reach of a small business

Thank you.

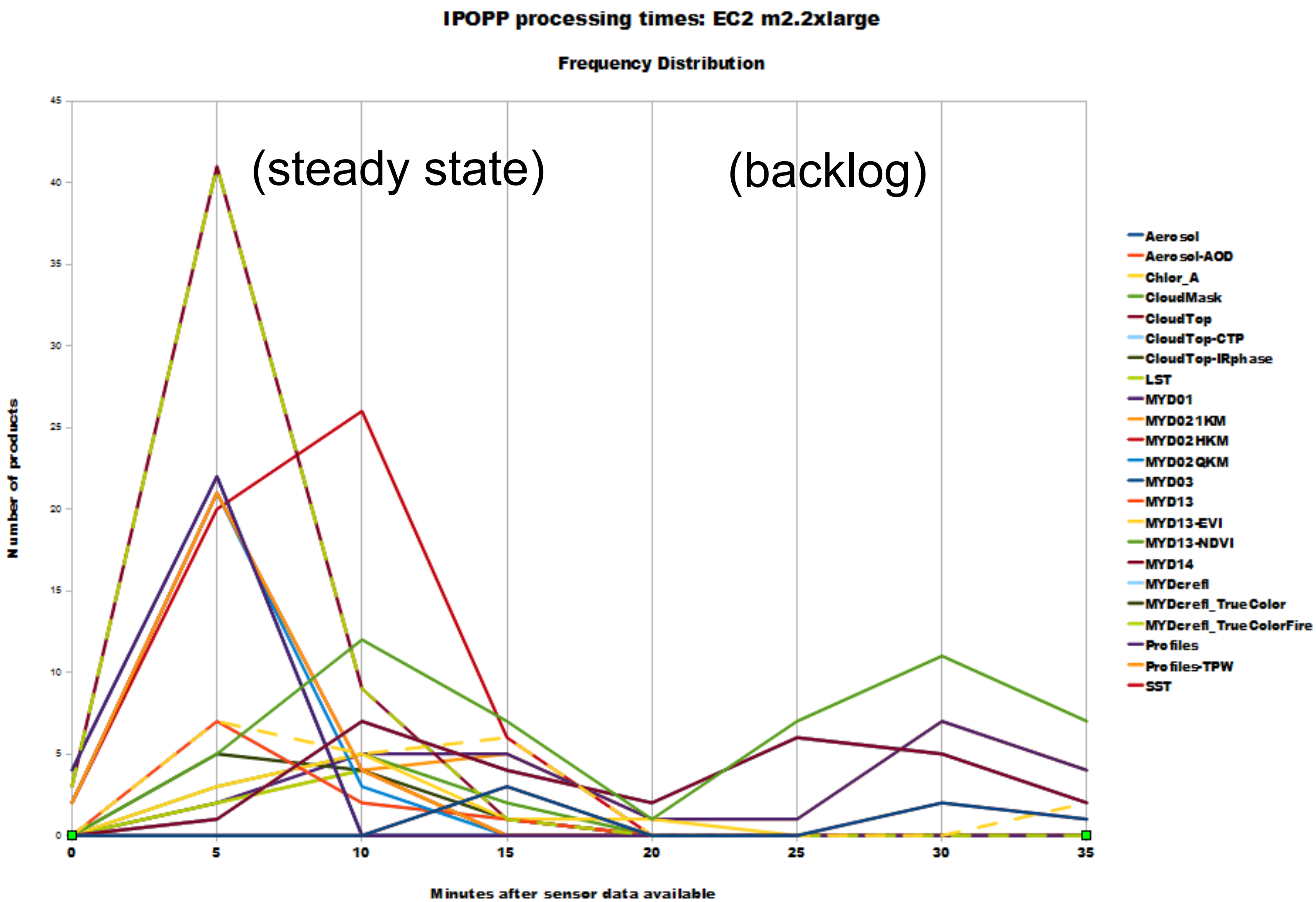
Any questions?.

Instance types tested

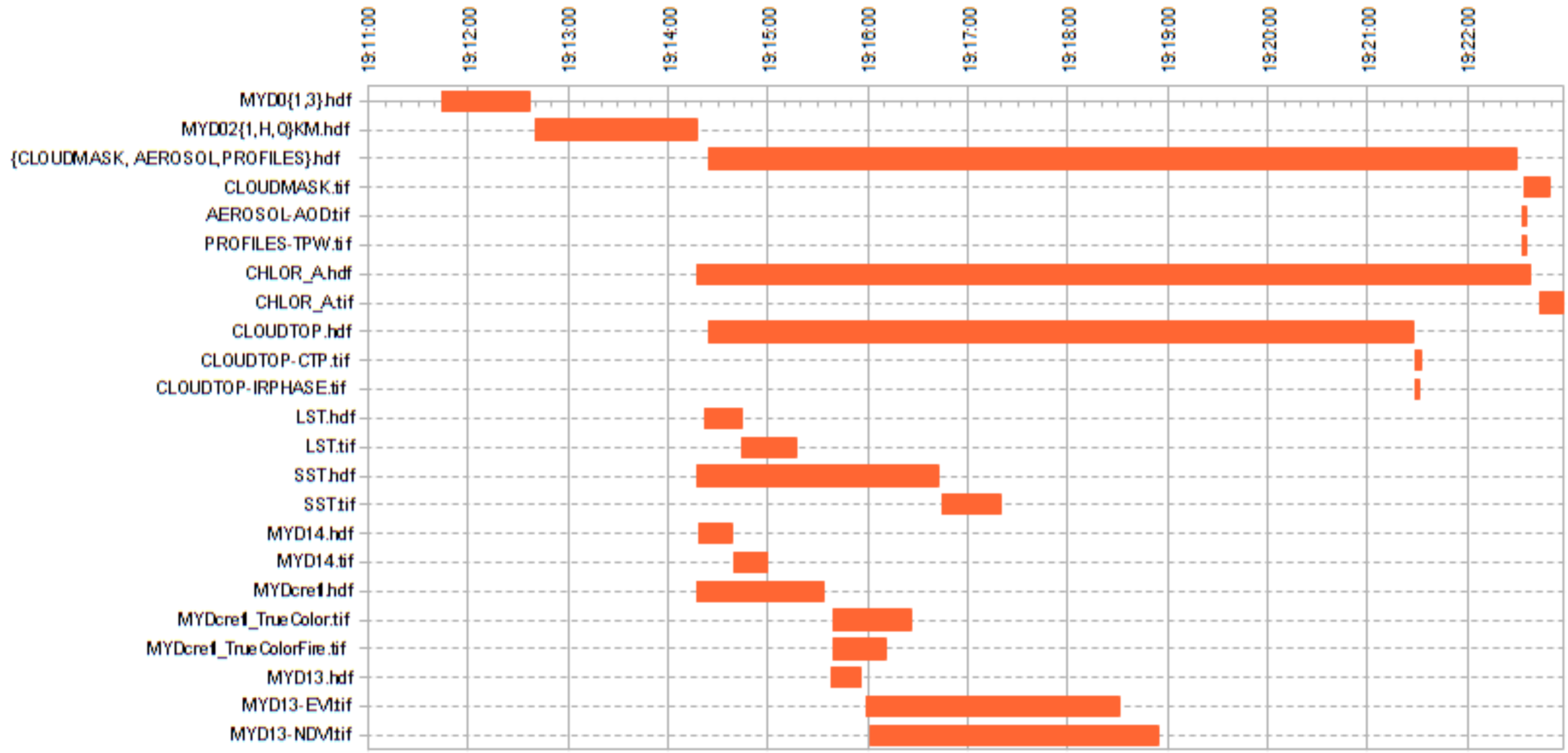
Instance Type	CPU type	PassMark Score	CPU cores	RAM (GB)	Cost ($/hr)
In-house server	Xeon X3220	3,114	4	8	
Amazon EC2 m1.large	Xeon E5430	4,158	2	7.5	$0.34
Amazon EC2 m1.xlarge			4	15	$0.68
Amazon EC2 c1.xlarge	Xeon E5506	3,374	8	7	$0.68
Amazon EC2 m2.xlarge	Xeon X5550	5,232	2	17.1	$0.50
Amazon EC2 m2.2xlarge			4	34.2	$1.00
Amazon EC2.4xlarge			8	68.4	$2.00
NASA Nebula m1.large			4	8	
NASA Nebula m1.xlarge			8	16	

(For detailed cloud-computing benchmarks see http://cloudharmony.com*)*

EC2 m2.2xlarge performance



Simultaneous execution: a single run



Cloud computing costs *(Amazon EC2)*

- CPU usage *(see table)*
  - Pennies per hour add up!
  - Note 100:1 cost ratio
- Data Storage
  - EBS: $100/TB/month
  - S3: $37-140/TB/month
- Data Transfer
  - In: $100/TB
  - Out: $80-150/TB
- Other
  - SQL queries; I/O requests
  - Snapshot GETs/PUTs

Instance type	cost/hr	/day	/mo	/yr
t1.micro	$0.02	$0.48	$15	$175
m1.small *(default)*	$0.085	$2	$62	$745
m1.large	$0.34	$8	$248	$2,978
m1.xlarge	$0.68	$16	$496	$5,957
c1.xlarge	$0.68	$16	$496	$5,957
m2.xlarge	$0.50	$12	$365	$4,380
m2.2xlarge	$1.00	$24	$730	$8,760
m2.4xlarge	$2.00	$48	$1,460	$17,520
cc1.4xlarge *(cluster compute)*	$1.60	$38	$1,169	$14,026
cg1.4xlarge *(GPU cluster)*	$2.10	$50	$1,534	$18,409

Ways to reduce cloud costs

- Reserved instances: fixed sum + lower hourly rate
  - Worthwhile if usage exceeds 11 hrs/day (1 year)
    … 8.5 hrs/day (2 years) … 5.5 hrs/day (3 years)
- Spot instances
  - Spot price: as little as 1/3 of regular hourly rate
  - Pick a rate ($/hr) ceiling; run until spot price exceeds ceiling
  - Useful when load-balancing across many identical machines
- **Scale up & down as needed**
  - A polar-orbiting satellite may let us save 60% or more by scaling up & down several times a day

Case study summary

- Proof of concept
  - Infrastructure as a Service made for a quick & easy transition
- Task effectiveness
  - Tried many different instance types @ low cost, no risk
  - Reduced latency from ~40 minutes to ~12
  - Significant add'l performance is clearly within reach
  - Gained significant bandwidth
- Cost effectiveness
  - Cloud computing costs are complex and significant

    (But so are data center costs)
  - Elastic provisioning and utility pricing worked well for us
    - Scaling up & down 2x/day will shrink our costs by 60+%

Other use cases and data streams

- Many fields could benefit from easier near-real-time access to weather & environmental data – *e.g.*,
  - Transportation … Emergency management ... Agriculture ... Law Enforcement …
- We're aiming to build an infrastructure suitable for many data streams – *e.g.*,
  - GOES (geostationary imagery) … Radar … Simulations … Sensor networks …
- Upstream and downstream interoperability will result from industry-standard interfaces – *e.g.*,
  - OGC Web Services … OPeNDAP …

In summary

- Proof of concept
  - Infrastructure as a Service made for a quick & easy transition
- Task effectiveness
  - Reduced latency from ~40 minutes to ~12
  - Significant add'l performance is clearly within reach
  - Gained significant bandwidth
- Cost effectiveness
  - Cloud computing costs are complex and significant

    (But so are data center costs)
  - By scaling up & down frequently, we hope to reduce costs 60%

    (This would not be practical without cloud computing)
- Commercial potential; many people can play

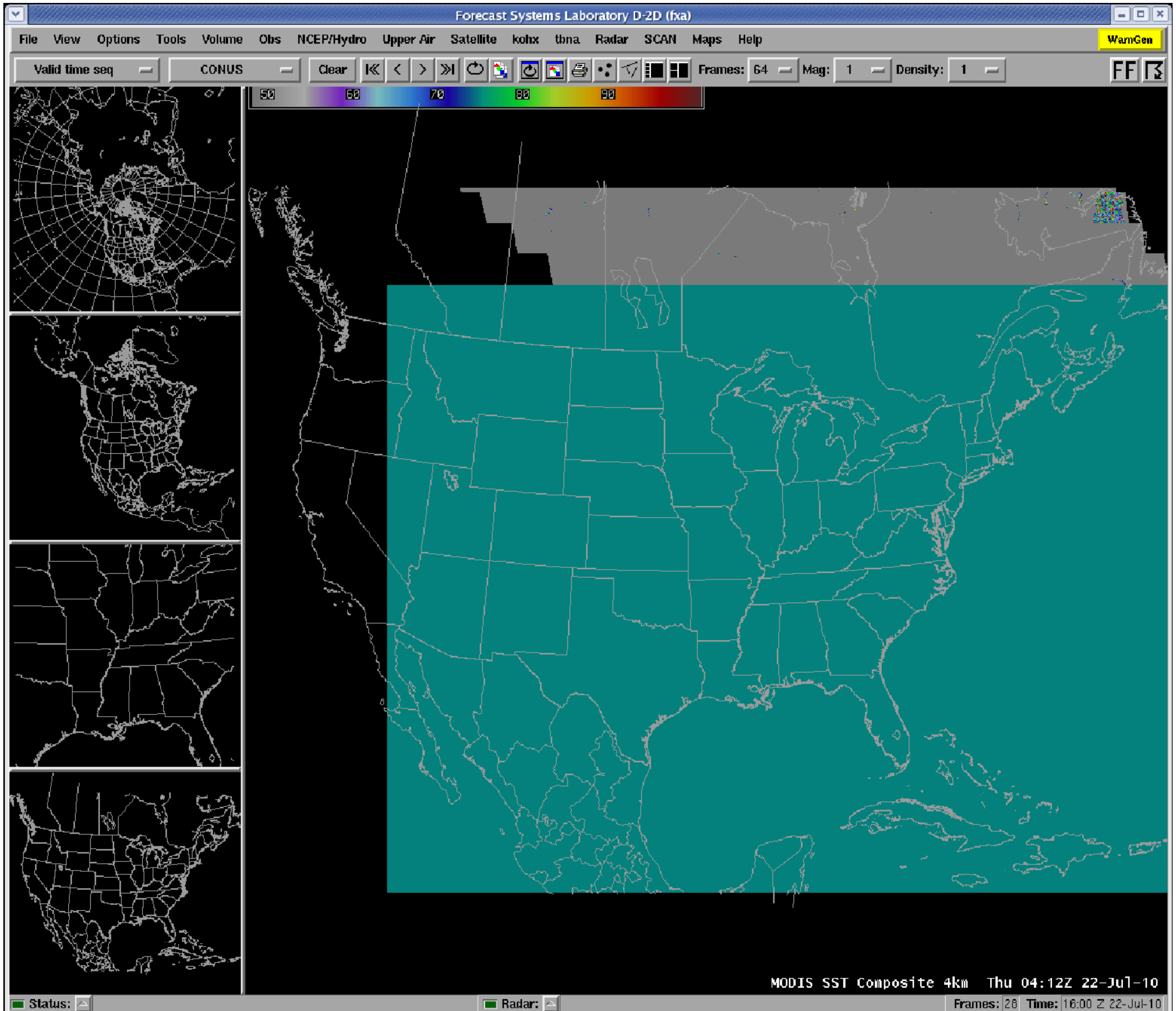


Figure 2. IPOPP-produced data conversion for AWIPS input

Objectives 4 (Cont'd)

Streamline performance to ensure real-time throughput to many users

- Optimize IPOPP algorithm scheduling
- Use lots of fast CPU cores to handle backlogs
- Distribute services & algorithms across multiple machines
- Parallelize algorithms
- Compile algorithms to run on GPUs
- Streamed data / streamed processing