Urgent Computing, Sharing Grid Resources, and Elastic Computing

Pete Beckman
Argonne National Laboratory
University of Chicago

http://www.mcs.anl.gov/~beckman
Urgent Computing: I Need it Now!

• Applications with dynamic data and **result deadlines** are being deployed
• Late results are useless
  - Wildfire path prediction
  - Storm/Flood prediction
  - Influenza modeling
• Some jobs need priority access
  “Right-of-Way Token”
How can we get cycles?

- Build supercomputers for the app
  - **Pros**: Resource is ALWAYS available
  - **Cons**: Incredibly costly (99% idle)
  - **Example**: Coast Guard rescue boats
- Share public infrastructure
  - **Pros**: low cost
  - **Cons**: Requires complex system for authorization, resource mgmt, and control
  - **Examples**: school buses for evacuation, cruise ships for temporary housing
Introducing SPRUCE

• The Vision:
  - Build cohesive infrastructure that can provide urgent computing cycles

• Technical Challenges:
  - Provide high degree of reliability
  - Elevated priority mechanisms
  - Resource selection, data movement

• Social Challenges:
  - Who? When? What?
  - How will emergency use impact regular use?
  - Decision-making, workflow, and interpretation
Existing “Digital Right-of-Way” Emergency Phone System

GETS is a "ubiquitous" service in the Public Switched Telephone Network…if you can get a DIAL TONE, you can make a GETS call

Calling cards are in widespread use and easily understood by the NS/EP User, simplifying GETS usage

GETS priority is invoked “call-by-call”

Dial 1-710-NCS-GETS (627-4387)
At the tone, enter your PIN.
When prompted, dial your destination number (area code + number).
If you cannot complete a call, use a different long distance carrier:

- AT&T: 1010 + 288 -or- 1-888-288-4387
- MCI: 1010 + 222 +1-710-627-4387 -or- 1-800-900-4387
- Sprint: 1010 + 333 -or- 1-800-257-8373

WPS
From a Wireless Priority Service enabled device:
Dial *272 before any call, including a GETS call.

Assistance: For help or to report trouble, dial 1-800-818-GETS (4387) or 1-703-818-GETS (4387).
Test Calls: Make periodic GETS calls to 703-818-3924.

US GOVERNMENT PROPERTY. If found, return to:
NCS (N3), PO Box 4502, Arlington, VA 22204-4502
WARNING: For Official Use Only by Authorized Personnel.

http://www.mcs.anl.gov/~beckman
SPRUCE Architecture Overview (1/3)
Right-of-Way Tokens

Event

1. Automated Trigger

2. Human Trigger

First Responder

SPRUCE Science Gateway

Human Trigger

SPRUCE Urgent Computing Right-of-Way Token
spruce.teragrid.org

TeraGrid

5D7E-XUU9-237Q-7123
SPRUCE Architecture Overview (2/3)

Submitting Urgent Jobs

1. User Team
2. Urgent Computing Job Submission
3. Choose a Resource
4. Authentication
5. Priority Job Queue
6. Supercomputer Resource

Choose a Resource

SPRUCE Job Manager

Local Site Policies

Urgent Computing Parameters

Conventional Job Submission

SPRUCE Urgent Computing

Flat Rock, North Carolina, 2006

http://www.mcs.anl.gov/~beckman
SPRUCE Architecture Overview (3/3)  
Analyzing Urgent Jobs

Supercomputer Resource → Results → Domain Specialist Interpreter

6

7

Decision Maker
Student fun with AJAX...
SPRUCE: Special PRiority and Urgent Computing Environment

Token: K8VT-F5FY-NSQD-D6ZS
Fetching User information..

Status: Activated
Lifetime: 72:00:00
Maximum Urgency: red
Creation date: 2006-09-08 10:57:33.0
Expiration date: 2006-12-12 12:00:00.0
Activation date: 2006-09-08 11:05:03.0
Deactivation date: 2006-09-11 11:05:03.0

Resources on TG:
- ia32 @ ANL
- ia64 @ ANL
- lear @ PUR
- Fast-CPU @ NCSA
- Mer-ia64 @ NCSA
- Fast-I0 @ NCSA

Users:
- Demo User remove
  (/C=US/O=SDSC/OU=SDSC/CN=Demo User/UID=duser)

(Fresh info as of Fri Sep 08 2006 11:15:24 GMT-0500 (CDT))

If you need to go back to the menu, press the User Portal link from the menu. Browser BACK button has no functionality.
Site–Local Response Policies: How will Urgent Computing be treated?

- “Next-to-run” status for priority queue; wait for running jobs to complete
- Force checkpoint of existing jobs; run urgent job
- Suspend current job in memory (kill –STOP); run urgent job
- Kill all jobs immediately; run urgent job

- Provide differentiated CPU accounting
  - “jobs that can be killed because they maintain their own checkpoints will be charged 20% less”
- Other incentives
Emergency Preparedness Testing: “Warm Standby”

• In urgent computing situation, there is no time to port applications
  ❖ Applications must be in “warm standby”
  ❖ Verification and validation runs test readiness periodically (Inca)
  ❖ Only verified apps participate in urgent computing

• Grid–wide Information Catalog
  ❖ Application was last tested & validated on <date>
  ❖ Also provides key success/failure history logs
Choosing a Resource
An Advisor

Urgent Computation Request

Deadline
Urgency Level

Live Job/Queue Data

<table>
<thead>
<tr>
<th>Platform</th>
<th>Next Available Job (Policy Based)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCSA::Cobalt</td>
<td>Immediate</td>
<td></td>
</tr>
<tr>
<td>SDSC::Datastar</td>
<td>(5.3 hrs, 1024 nodes)</td>
<td>...</td>
</tr>
<tr>
<td>PSC::Rachel</td>
<td>Immediate</td>
<td></td>
</tr>
</tbody>
</table>

Site Policies

<table>
<thead>
<tr>
<th>Platform</th>
<th>Policy</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCSA::Cobalt</td>
<td>Human-in-the-loop, immediate access, kill existing jobs, 15 min. turnaround</td>
<td>...</td>
</tr>
<tr>
<td>SDSC::Elimidata</td>
<td>Automated, next job</td>
<td>...</td>
</tr>
<tr>
<td>SDSC::Datastar</td>
<td>Normal priority, no SPRUCE support</td>
<td></td>
</tr>
<tr>
<td>PSC::Rachel</td>
<td>Automated, immediate access, kill existing jobs, 10 min turnaround</td>
<td>...</td>
</tr>
</tbody>
</table>

Warm Standby Validation History

<table>
<thead>
<tr>
<th>Platform</th>
<th>App.</th>
<th>Validated</th>
<th>Reliability</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCSA::Cobalt</td>
<td>Tornado</td>
<td>8 days ago</td>
<td>95%</td>
<td></td>
</tr>
<tr>
<td>NCSA::Cobalt</td>
<td>City Airflow</td>
<td>14 days ago</td>
<td>98%</td>
<td></td>
</tr>
<tr>
<td>SDSC::Elimidata</td>
<td>City Airflow</td>
<td>45 days ago</td>
<td>78%</td>
<td></td>
</tr>
<tr>
<td>SDSC::Elimidata</td>
<td>Influenza</td>
<td>30 days ago</td>
<td>59%</td>
<td></td>
</tr>
</tbody>
</table>

Best HPC Resource
Deployment Status

- Deployed and available:
  - UC/ANL
  - Purdue
  - TACC
  - SDSC
- Very close:
  - Indiana
  - LSU
- Ready to integrate LEAD into SPRUCE
  - First user–customer
  - Warm standby apps
What About “Capacity” Computing?

- SPRUCE works well with “capability” computing:
  - Interface to small set of large resources
- Imagine a larger set of smaller resources?
  - Condor management?
  - Real on-demand servers?
- Amazon S3 & EC2
Amazon S3 & EC2
It’s a Web Services World

- S3: Simple Storage Service
  - Cost: $0.20/GB transfer, $.15/GB-month
- EC2: Elastic Compute Cloud
  - Cost: $0.10/cpu-hr, $0.20/GB transfer
  - No cost for internal bandwidth
- Cost is extraordinarily good
- Commoditization is good!!
- The real keys are reliability and dynamic behavior
## Account Activity

### Account Number: 871212-12314513127

#### Summary of This Month's Activity as of August 30, 2006

**Billing Cycle for this Report:** August 1 - August 31, 2006

<table>
<thead>
<tr>
<th>Usage Charges</th>
<th>Rate</th>
<th>Usage</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amazon Simple Storage Service</strong></td>
<td>$0.20 per GB of data transferred</td>
<td>0.007 GB</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$0.15 per GB-Month of storage used</td>
<td>0.000 GB-Mo</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>0.02</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Amazon Elastic Compute Cloud</strong></td>
<td>$0.10 per instance hour consumed</td>
<td>1 BoxUsage</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>(or part of an hour consumed)</td>
<td>8.6121741196425e-05</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$0.20 per GB of data transferred outside of Amazon (i.e., Internet traffic)</td>
<td>Usage Report</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>0.11</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Subtotal** $0.13

**Taxes** $0.00

**Charges due on September 1, 2006** $0.13

---

*All charges for this billing cycle will be charged to your credit card on your next billing date, September 1, 2006. These charges include 1) next billing cycle's subscription charges due on the next billing date and 2) usage charges from the current billing cycle. Not included in the charges displayed here are any additional usage charges you will accrue this billing cycle. The current billing cycle ends August 31, 2006 GMT. Visit the Amazon Web Services FAQs to learn more about web services pricing models and billing.

All web services are sold by Amazon Digital Services, Inc.*
Imagine…

- Other companies catching up...
- Commoditization (like web email)
- A standardized interface to web-service “request vm”
- Dynamic capacity provides availability of 250K “node instances”
- urgent computing resources available immediately
- Missing bisection bandwidth, but great for capacity computing
The Future

- Web services interfaces to all the portal functions
- Extended submission schema
- Flexible tokens – aggregation, extension
- Encode local site policies
- Warm standby integration
- Automated ‘advisor’
- Data movement
- Redundancy to avoid downtime of portal
Questions?
Ready to Join?

spruce@ci.uchicago.edu
beckman@mcs.anl.gov

http://spruce.teragrid.org