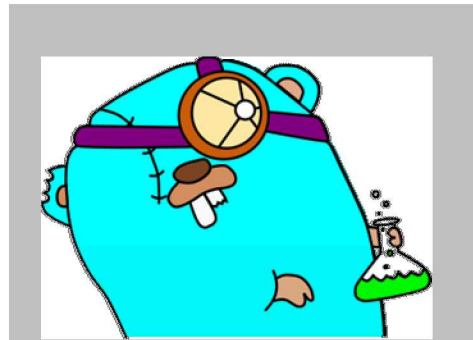


Igor Web

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192



Walk This Way...

A New Approach to Compute Cluster Scheduling and Provisioning using igor

March, 2020

Traditional Compute Cluster Needs

- Traditional HPC workflow is hands-off
 - Schedule Time
 - Hand over your code
 - Wait for results

- HPC Workflow management tools exist
 - Provisioning
 - Privileged
 - inflexible
 - Resource Management
 - No Scheduling (usually)
 - Licensing

Along Came Emulytics

- Sandia developed Emulytics for cyber security research, testing, and training
- Emulytics = emulation + analytics
- Run virtual networks of hundreds of thousands of virtual machines
- Set up realistic environments that mimic real-world corporate and internet systems
- Integrate Windows, Linux, and Android VMs into a single virtual environment
- minimega is one of several Sandia Emulytics tools

Evolving Compute Cluster Needs

- New Approaches to computation
 - Emulytics program support
 - Assessment of cyber systems at scale – 2 phases
 - Building/iterating cyber testbed
 - running cyber experimentation

- New Requirements: Hands-on
 - Researchers need access to bare metal
 - Customized configurations (OS, BIOS, VLAN, etc.)
 - Flexible provisioning
 - Resource scheduling!
 - Work with existing systems
 - Avoid Licensing (if possible)

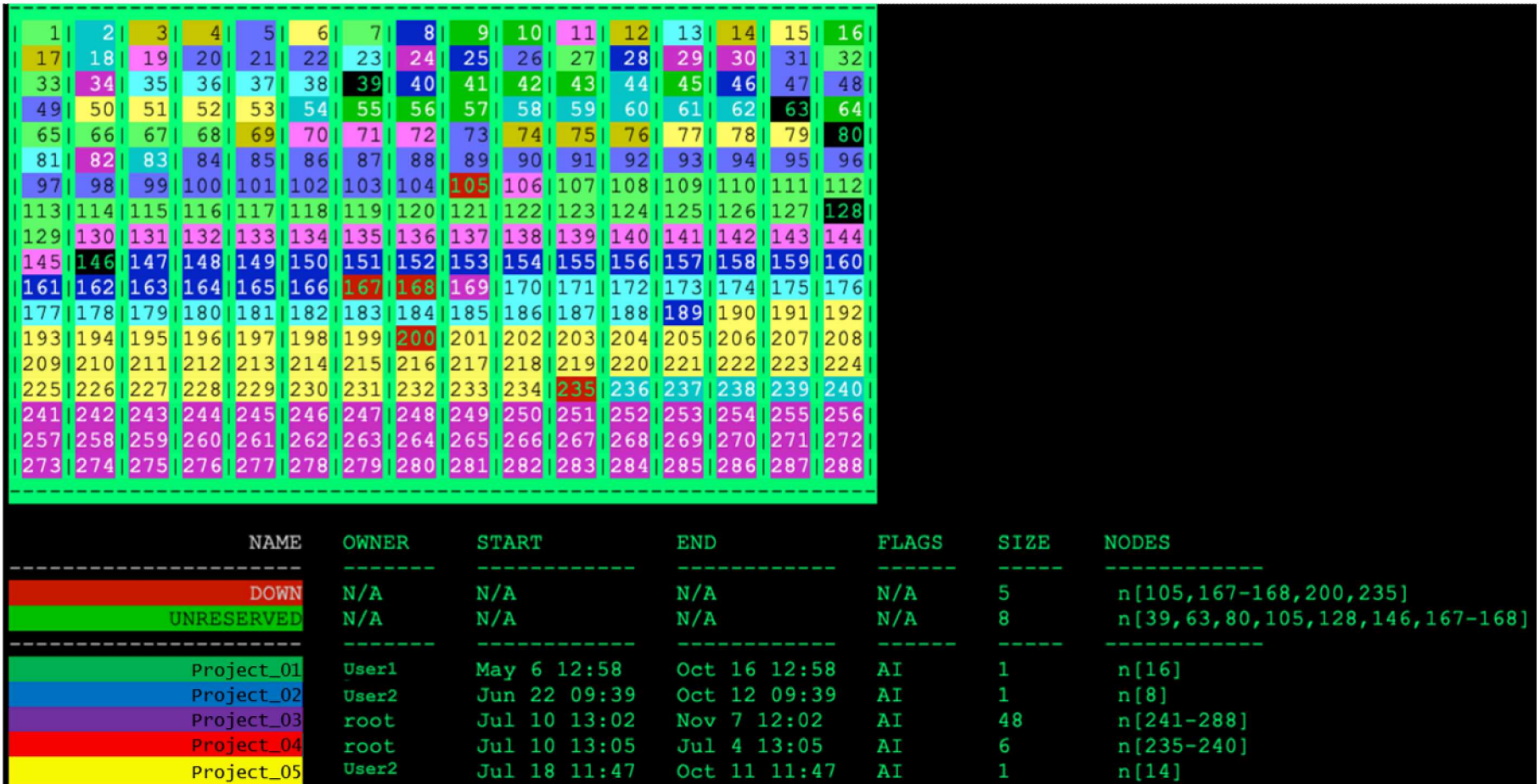
- Provisioning
 - TFTP/PXE
 - Support for Kernel/Initrd pairs
 - Integration with existing systems (Cobbler)
- Scheduling
 - CLI scheduler for sharing compute nodes
 - Web UI with full feature parity
- Open-Source
 - minimega tool suite
 - minimega.org

flexible, feature-rich

- Written in Go
 - golang.org
- Configurable
 - Single .conf file
- Scriptable (cli)
- Support for VLAN segmentation (with Arista)
- Usage Statistics
- Logging

igor

Open-Source Compute cluster scheduling and provisioning



igor web

Web UI for a secure, pleasant experience

Igor Web

+

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240

Available

Reserved

Powered On 5 278

Powered Off 3 2

Filter Reservations

☒ Owner
 ☐ Group
 ☐ Start Time
 ☒ End Time
 ☐ Nodes
 ☒ Range

Name ↑	Owner	End Time	Range
adminRes	root	Dec 18 10:36	node [200]
myExperiment	user01	Nov 7 12:02	node [241-288]
cyberTestbed	user02	Oct 1 10:59	node [45]

igor – easy to use

- **Show** – Cluster/Reservations at a glance
- **Sub** – reserve specific/first-available node(s)
- **Extend** – when you need more time
- **Edit** – Needs change, so can your reservation
- **Power** – on/off/cycle (ipmi, or custom)
- **Notify** – email notifications for events
- **Stats** – log analysis for usage statistics
- Specify vlan tags – isolate your experiment (new)
- ...and more

igor – easy to configure

- "tftproot ",
- "usecobbler",
- "poweron/offcommand",
- "autoreboot",
- "vlan_min/max ",
- "network": "arista",
- "node_map ": {"pre1": "<port>"}

igor – not so easy to test

- Testing on the developer's desktop prohibitive
 - Igor expects to be able to communicate with resources it needs
- Testing production software in a live environment
 - Discouraged (traditionally)
- Solution: use minimega
 - Define notional (or specific) compute cluster environment
 - Include all resources igor expects
 - Break to your heart's content

- Active Development, Community
 - Dedicated team of developers
 - igor running on 2 clusters currently, a third outside Sandia planned
- FY20 Roadmap
 - Database integration
 - Track/Reserve by Node features/metadata
 - Richer usage statistics
 - Re-architect to run as a service
 - Support node state, callbacks, RESTful API
 - DNS, DHCP, TFTP
- Long Term
 - Distributed, Federated igors

Thank you! Questions?

