Tools Demonstration

Overview

- **Power Measurement Methods**
- System Configuration
- Running Benchmarks
- Core/Package Idleness

Power Measurement Methods

- RAPL counters
  - Interface reporting energy consumption of different power domains (i.e., package, core, dram, gpu) based on models
  - Granularity: core, package, memory, gpu
  - Tools: Performance Counters, Turbostat

- Server Integrated Power Meter
  - Built-in feature of a server that monitors and reports real-time power consumption using hardware sensors and models
  - Granularity: cpu (core + package), memory, gpu, platform (e.g., chipset, disk)
  - Tools: Remote Console (e.g., HPE iLo)

- PowerWall meter
  - Power consumption of a server as measured by the wall sockets
  - Granularity: server
  - Tools: pdu, smart plug

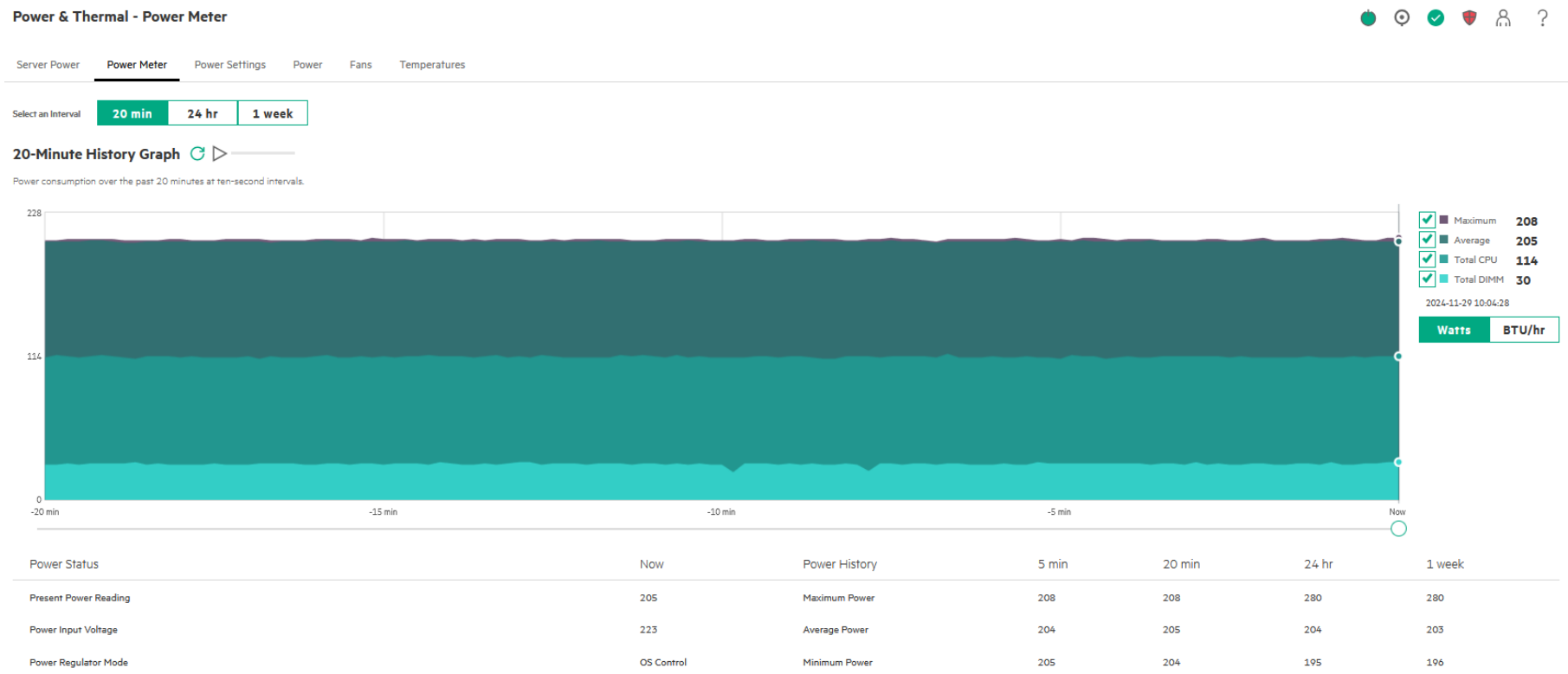
Power Measurement Methods-RAPL

- Turbostat

Package	Core	CPU	Avg_MHz	Busy%	Bzy_MHz	TSC_MHz	IRQ	SMI	C1	C1E	C6	C1%	C1E%	C6%	CPU%c1	CPU%c6	CoreTmp	PkgTmp	PkgWatt	RAMWatt	PKG_%	RAM_%
-	-	-	2	0.26	884	2201	2558	0	0	50	3013	0.00	0.02	99.79	1.22	98.52	44	45	43.51	44.14	0.00	0.00
0	0	0	3	0.29	854	2203	75	0	0	2	84	0.00	0.02	99.82	1.71	98.00	43	45	20.95	22.40	0.00	0.00
0	0	20	11	0.85	1318	2203	82	0	0	1	72	0.00	0.01	99.27	1.15							
0	1	1	2	0.25	800	2203	89	0	0	2	92	0.00	0.01	99.87	1.02	98.72	43					
0	1	21	1	0.11	800	2203	49	0	0	1	54	0.00	0.00	100.02	1.17							
0	2	2	1	0.11	800	2203	42	0	0	0	43	0.00	0.00	100.02	1.43	98.46	42					
0	2	22	3	0.40	800	2203	121	0	0	4	128	0.00	0.04	99.69	1.13							
0	3	3	2	0.21	800	2202	68	0	0	1	73	0.00	0.01	99.91	1.24	98.55	42					
0	3	23	2	0.31	800	2202	65	0	0	0	80	0.00	0.00	99.82	1.13							
0	4	4	2	0.20	800	2202	64	0	0	1	68	0.00	0.02	99.91	0.72	99.08	44					
0	4	24	1	0.11	800	2202	34	0	0	0	39	0.00	0.00	100.01	0.80							
0	8	5	1	0.08	800	2202	26	0	0	0	28	0.00	0.00	100.03	0.39	99.53	43					
0	8	25	1	0.07	800	2202	19	0	0	1	24	0.00	0.00	100.04	0.40							
0	9	6	0	0.06	800	2202	20	0	0	1	21	0.00	0.01	100.04	0.33	99.60	42					
0	9	26	1	0.08	800	2202	25	0	0	2	28	0.00	0.01	100.02	0.31							
0	10	7	1	0.10	800	2202	24	0	0	0	29	0.00	0.00	100.01	0.25	99.65	43					
0	10	27	0	0.02	800	2202	12	0	0	0	14	0.00	0.00	100.09	0.33							
0	11	8	2	0.27	853	2201	50	0	0	2	48	0.00	0.02	99.78	3.30	96.43	42					
0	11	28	2	0.27	807	2201	439	0	0	2	427	0.00	0.40	99.43	3.30							
0	12	9	2	0.22	800	2201	120	0	0	0	136	0.00	0.00	99.83	1.23	98.55	43					
0	12	29	1	0.10	800	2201	29	0	0	1	33	0.00	0.00	99.95	1.35							
1	0	10	3	0.37	801	2200	47	0	0	0	148	0.00	0.00	99.67	1.59	98.04	38	41	22.56	21.74	0.00	0.00
1	0	30	2	0.30	801	2200	60	0	0	1	61	0.00	0.00	99.72	1.66							
1	1	11	10	1.25	800	2200	119	0	0	2	174	0.00	0.01	98.77	1.64	97.10	37					
1	1	31	1	0.15	800	2200	32	0	0	1	31	0.00	0.01	99.87	2.75							
1	2	12	4	0.45	800	2200	111	0	0	7	186	0.00	0.04	99.54	1.79	97.76	37					
1	2	32	1	0.16	800	2200	40	0	0	0	38	0.00	0.00	99.86	2.07							
1	3	13	1	0.16	800	2200	75	0	0	0	101	0.00	0.00	99.86	0.95	98.88	37					
1	3	33	0	0.05	800	2200	33	0	0	0	32	0.00	0.00	99.97	1.06							
1	4	14	1	0.07	800	2200	25	0	0	0	38	0.00	0.00	99.95	0.73	99.20	39					
1	4	34	1	0.15	800	2200	49	0	0	0	47	0.00	0.00	99.87	0.65							
1	8	15	12	1.03	1172	2200	54	0	0	0	55	0.00	0.00	98.99	1.44	97.53	38					
1	8	35	2	0.26	800	2200	89	0	0	3	108	0.00	0.02	99.75	2.22							
1	9	16	1	0.14	800	2200	46	0	0	1	50	0.00	0.00	99.88	1.16	98.70	37					
1	9	36	2	0.29	800	2200	65	0	0	1	96	0.00	0.01	99.72	1.01							
1	10	17	0	0.03	800	2200	16	0	0	1	17	0.00	0.01	99.97	0.65	99.32	38					
1	10	37	2	0.19	800	2200	52	0	0	6	53	0.00	0.02	99.80	0.49							
1	11	18	1	0.08	800	2200	26	0	0	0	27	0.00	0.00	99.93	0.97	98.95	36					
1	11	38	3	0.38	800	2200	47	0	0	1	76	0.00	0.01	99.63	0.67							
1	12	19	1	0.16	801	2200	50	0	0	2	56	0.00	0.02	99.83	1.44	98.40	38					
1	12	39	3	0.41	800	2200	69	0	0	3	98	0.00	0.02	99.59	1.18							

Power Measurement Methods-Remote Console

- iLO Remote Console - Interface



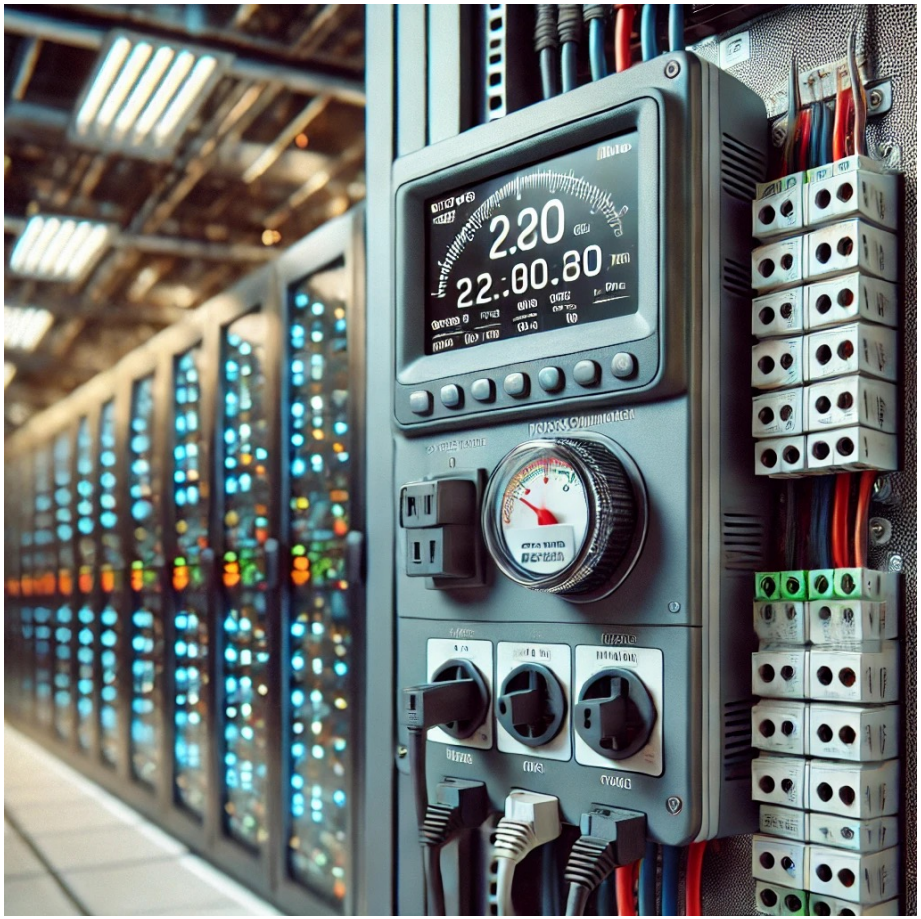
Power Measurement Methods-Server Integ. Power Meter

- iLO Remote Console - API


```
"@odata.context": "/redfish/v1/$metadata#HpePowerMeter.HpePowerMeter",
"@odata.etag": "W/\"C1120F83\"",
"@odata.id": "/redfish/v1/Chassis/1/Power/FastPowerMeter",
"@odata.type": "#HpePowerMeter.v2_0_1.HpePowerMeter",
"Average": 157,
"Id": "FastPowerMeter",
"Maximum": 307,
"Minimum": 138,
"Name": "PowerMeter",
"PowerDetail": [
    {
        "AmbTemp": 27,
        "Average": 147,
        "Cap": 0,
        "CpuAvgFreq": 0,
        "CpuCapLim": 100,
        "CpuMax": 0,
        "CpuPwrSavLim": 100,
        "CpuUtil": 0,
        "CpuWatts": 63,
        "DimmWatts": 28,
        "GpuWatts": 0,
        "Minimum": 147,
        "Peak": 148,
        "PrMode": "osc",
        "PunCap": false,
        "Time": "2023-10-16T22:33:23Z",
        "UnachCap": false
    },
    {
        "AmbTemp": 27,
        "Average": 147,
```

Power Measurement Methods-PowerWall Meter

- PowerWall meter - PDU

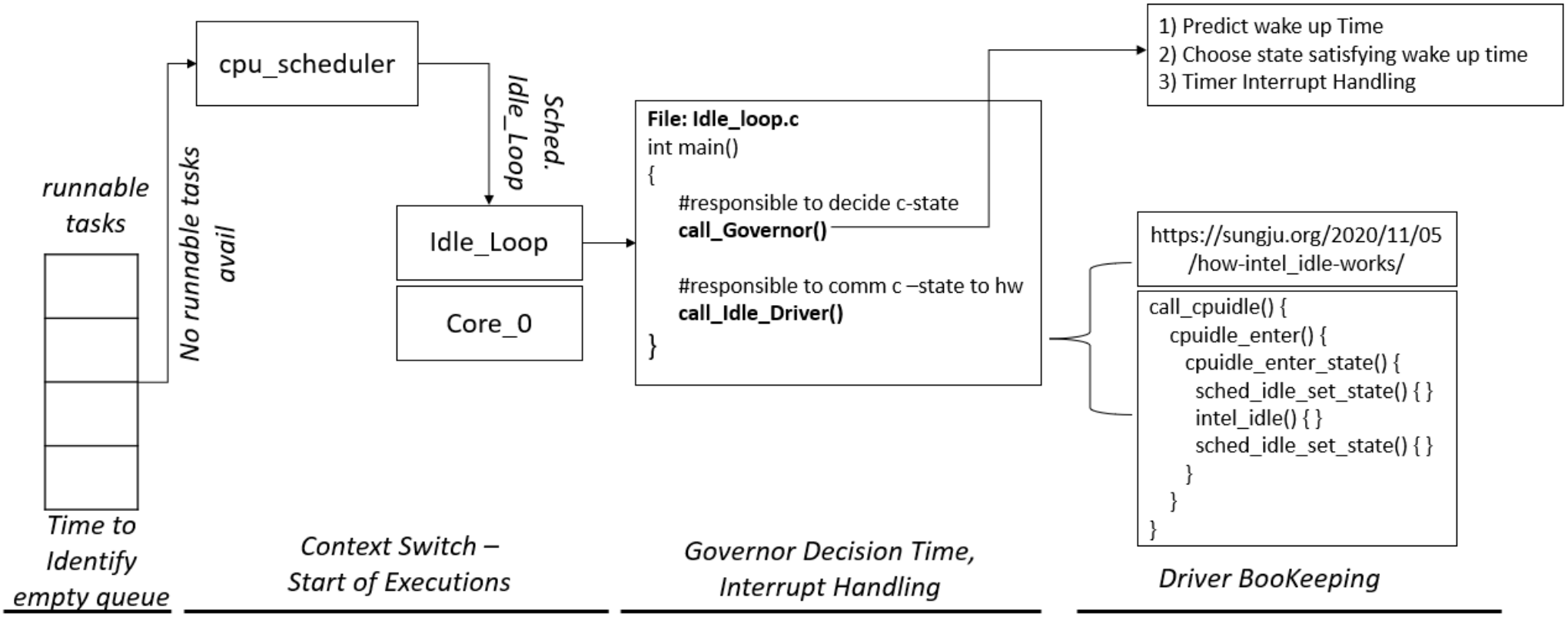


```
username = argv[1]
password = argv[2]
hostname = argv[3]
experiment_duration = int(argv[4])
start_time = round(time.time())
end_time = start_time + experiment_duration

pdu1 = ssh_connection_to_pdu("[redacted], username, password)
pdu2 = ssh_connection_to_pdu("[redacted], username, password)

host_to_rcps = {}
host_to_rcps[[redacted]] = [(pdu2, [redacted]]
host_to_rcps[[redacted]] = [(pdu1, [redacted]]

rcps = host_to_rcps[hostname]
while round(time.time()) < end_time:
  power = 0
  for rcp in rcps:
    pdu_ssh = rcp[0]
    pdu_command = rcp_to_command(rcp[1])
    stdin, stdout, stderr = pdu_ssh.exec_command(pdu_command)
    lines = stdout.readlines()
    m = re.match("\[.*\]-+\(\d*\).([0-9]+.?[0-9]*).W", lines[5])
    if m:
      power += float(m.group(1))
  print(str(round(time.time() - start_time))+","+str(power))
```

Overview

- Power Measurement Methods
- **System Configuration**
- Running Benchmarks
- Core/Package Idleness

SW C-state Entry Flow



PM Configuration

- Idle Governor (menu, ladder)
  - Software part of the CPUIdle Subsystem responsible to decide which C-state a core should enter based on some heuristics
  - Tools: Grub parameter, cpupower
- Scaling Driver/Governor
  - Software part of the CPUFreq Subsystem responsible to scale the frequency/voltage (DVFS) of the system
  - Tools: Grub parameter, cpupower
- Core C-states
  - Power saving states a core enters when idle to reduce its power consumption
  - Tools: Grub parameter, cpupower, turbostat

System Configuration – Idle Governor

- Available Idle Governors
  - Command: **zgrep CONFIG_CPU_IDLE /boot/config-$(uname -r)**

```
CONFIG_CPU_IDLE=y
CONFIG_CPU_IDLE_GOV_LADDER=y
CONFIG_CPU_IDLE_GOV_MENU=y
```

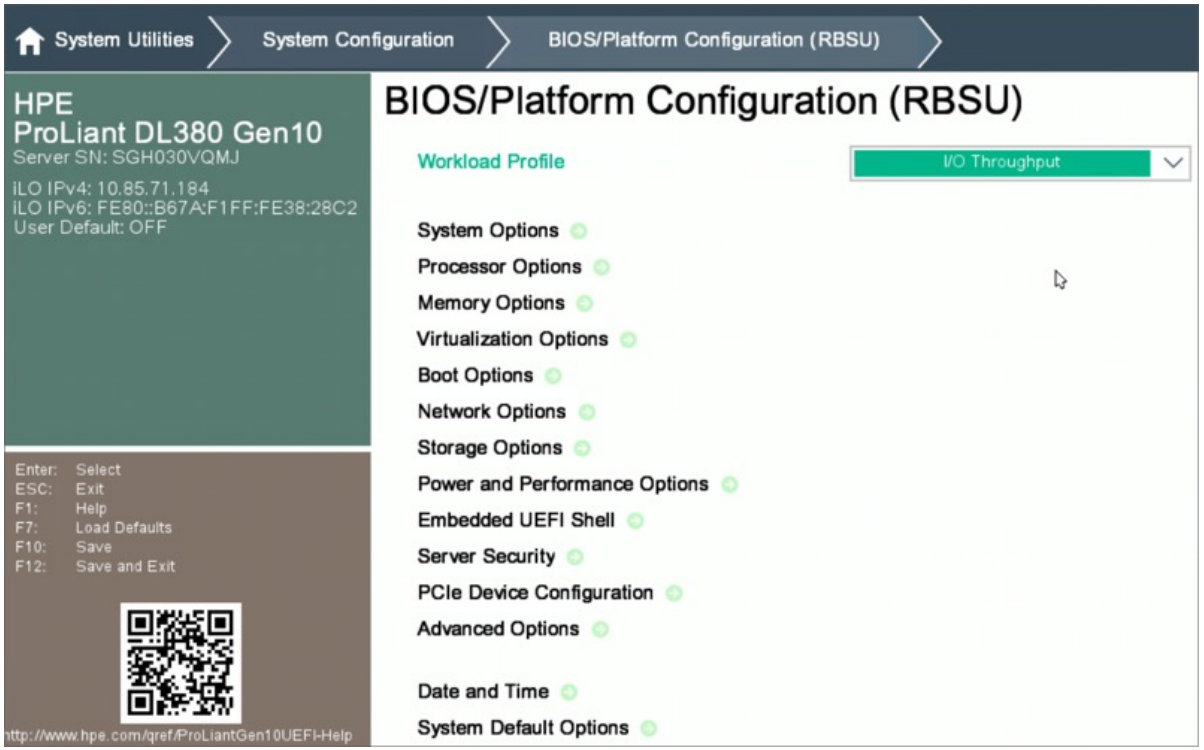
- Current Idle Governor
  - Command: **sudo cpupower idle-info**

```
CPUidle driver: intel_idle
CPUidle governor: menu
```

- Enable Idle Governor
  - Insert the parameter **cpuidle.governor=menu** in the grub file

System Configuration – Core C-states

- Available Core C-states
  - Command: **BIOS, Intel Documentation**



System Configuration – Core C-states

- Current Enabled Core C-states
  - Command: **sudo turbostat,**
  - Command: **sudo cpupower idle-info**
  - Command: **ls /sys/devices/system/cpu0/cpuidle/state*/name**

```
C1%     C1E%    CPU%c1  CPU%c6
0.00    99.80   99.79   0.00
0.00    99.76   99.73   0.00
0.00    99.83   99.80
0.00    99.90   99.88   0.00
0.00    99.87   99.84
0.00    99.91   99.89   0.00
0.00    99.93   99.91
0.00    99.98   99.96   0.00
0.00    99.94   99.92
0.00    99.98   99.96   0.00
0.00    99.75   99.74
0.00    99.94   99.93   0.00
0.00    99.81   99.79
0.00    99.77   99.75   0.00
```

```
Number of idle states: 3
Available idle states: POLL C1 C1E
POLL:
Flags/Description: CPUIDLE CORE POLL IDLE
Latency: 0
Usage: 2
Duration: 28816
C1:
Flags/Description: MWAIT 0x00
Latency: 2
Usage: 1617
Duration: 7649460
C1E:
Flags/Description: MWAIT 0x01
Latency: 10
Usage: 48089
Duration: 3131715931
```

```
ganton12@node0:~$ cat /sys/devices/system/cpu/cpu0/cpuidle/state*/name
POLL
C1
C1E
```

System Configuration – Core C-states

- Enable/Disable Core C-states
  - Insert the parameter intel_idle.max_cstate=n in the grub file where n is:
    - 3 - all C-states enabled
    - 2 - all except C6
    - 1 - all except C1E and C6
    - 0 - C1 and C0 enabled
  - To disable all C-states insert the parameters:
    - intel_idle.max_cstate=0
    - idle=poll

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=false
GRUB_TIMEOUT=4
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT=""
GRUB_CMDLINE_LINUX="console=ttyS1,115200 intel_idle.max_cstate=2"
```

Overview

- Power Measurement Methods
- System Configuration
- **Running Benchmarks**
- Core/Package Idleness

Deploying Memcached and Mutilate

- Memcached
  - Popular key-value store deployed as a low latency caching service
  - Deployment
    - Version: 1.6.12 (https://github.com/memcached/memcached.git)
    - Necessary dependencies: sudo apt install libevent-dev libzmq3-dev sysstat msr-tools linux-tools-common linux-tools-generic
    - Commands to install: git clone * cd ./memcached/; ./autogen.sh; ./configure; make -j4
- Mutilate
  - Memcached synthetic workload generator
  - Deployment
    - Version: 0.4 (https://github.com/shaygalon/memcache-perf)
    - Necessary dependencies: Same as Memcached
    - Commands to install: git clone* pushd memcache-perf; git checkout 0afbe9b; make -j4

Running Memcached

- Start Memcached process:
  - ./memcached -t 10 -m 16384 -c 32768 &
    - 10 worker threads
    - 16GB of memory
    - 32768 maximum simultaneous connections

- Pin worker threads to cores:
  - tids=$(ps -p `pgrep memcached` -o tid= -L | sort -n | tail -n +3 | head -10)s; cpu_id=0; for tid in $tids; do taskset -pc $cpu_id $tid; ((cpu_id=cpu_id+1)); done

Running Mutilate

- Load dataset to Memcached
  - ./mcperf -s localhost --loadonly -r 1000000 --iadist=fb_ia --keysize=fb_key --valuesize=fb_value
  - telnet localhost 11211 , stats

```
STAT curr_items 1000000
STAT total_items 1000000
```

- Run mutilate workload generator
  - ./mcperf -s localhost --noload -B -T 40 -c 1 -q 200 -t 120 -r 1000000 --iadist=fb_ia --keysize=fb_key --valuesize=fb_value

```
#type       avg     std     min      p5     p10     p50     p67     p75     p80     p85     p90     p95     p99    p999   p9999
read      102.0    10.1    74.4    88.7    90.8   100.6   105.0   107.1   108.4   110.4   114.1   117.9   132.6   162.6   230.7
update      0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
op_q        1.0     0.0     1.0     1.0     1.0     1.0     1.1     1.1     1.1     1.1     1.1     1.1     1.1     1.1     1.1

Total QPS = 192.3 (23101 / 120.2s)

Total connections = 40
Misses = 0 (0.0%)
Skipped TXs = 0 (0.0%)
```

Overview

- Power Measurement Methods
- System Configuration
- Running Benchmarks
- **Core/Package Idleness**

Core/Package Idleness

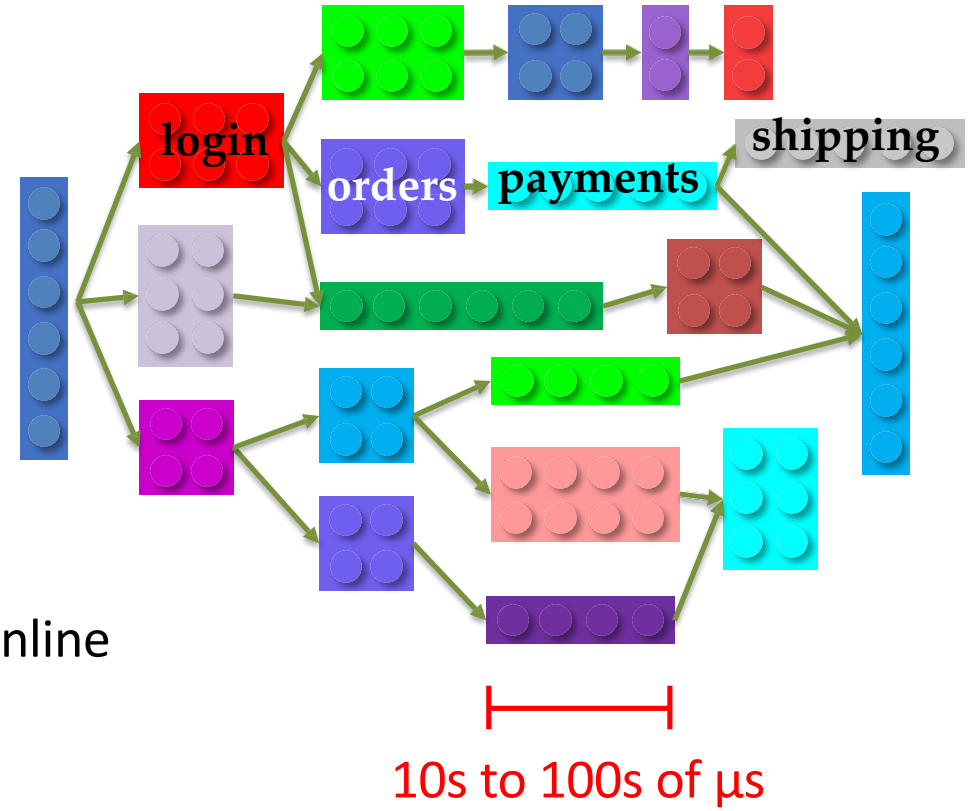
- Turbostat

Package	Core	CPU	Avg_MHz	Busy%	Bzy_MHz	TSC_MHz	IRQ	SMI	C1	C1E	C1%	C1E%	CPU%c1	CPU%c6	CoreTmp	PkgTmp	PkgWatt	RAMWatt	PKG_%	RAM_%
-	-	-	2	0.16	933	2200	2669	0	1	2974	0.00	99.84	99.84	0.00	49	49	51.29	44.21	0.00	0.00
0	0	0	2	0.29	800	2200	125	0	0	147	0.00	99.72	99.71	0.00	48	49	23.90	22.44	0.00	0.00
0	0	20	0	0.02	799	2200	21	0	0	22	0.00	99.99	99.98							
0	1	1	1	0.17	800	2200	42	0	0	82	0.00	99.84	99.83	0.00	48					
0	1	21	1	0.14	800	2200	27	0	0	29	0.00	99.86	99.86							
0	2	2	2	0.24	800	2200	70	0	0	93	0.00	99.77	99.76	0.00	47					
0	2	22	0	0.04	800	2200	30	0	0	30	0.00	99.96	99.96							
0	3	3	1	0.14	809	2200	42	0	0	70	0.00	99.86	99.86	0.00	47					
0	3	23	10	0.75	1359	2200	59	0	0	52	0.00	99.25	99.25							
0	4	4	1	0.19	800	2200	45	0	0	65	0.00	99.82	99.81	0.00	49					
0	4	24	1	0.07	800	2200	94	0	0	96	0.00	99.94	99.93							
0	8	5	2	0.21	800	2200	144	0	1	165	0.00	99.80	99.79	0.00	48					
0	8	25	1	0.11	800	2200	31	0	0	33	0.00	99.89	99.89							
0	9	6	1	0.08	800	2200	34	0	0	39	0.00	99.93	99.92	0.00	47					
0	9	26	1	0.10	800	2200	34	0	0	39	0.00	99.90	99.90							
0	10	7	1	0.15	800	2200	156	0	0	163	0.00	99.86	99.85	0.00	48					
0	10	27	1	0.11	800	2200	45	0	0	51	0.00	99.89	99.89							
0	11	8	1	0.08	800	2200	37	0	0	41	0.00	99.92	99.92	0.00	47					
0	11	28	1	0.15	800	2200	64	0	0	71	0.00	99.85	99.85							
0	12	9	1	0.16	800	2200	47	0	0	53	0.00	99.85	99.84	0.00	48					
0	12	29	1	0.10	800	2200	34	0	0	44	0.00	99.90	99.90							
1	0	10	1	0.12	800	2200	51	0	0	63	0.00	99.88	99.88	0.00	44	45	27.39	21.77	0.00	0.00
1	0	30	2	0.21	800	2200	528	0	0	535	0.00	99.81	99.79							
1	1	11	2	0.23	800	2200	80	0	0	81	0.00	99.77	99.77	0.00	42					
1	1	31	0	0.06	800	2200	31	0	0	44	0.00	99.95	99.94							
1	2	12	2	0.20	800	2200	43	0	0	49	0.00	99.81	99.80	0.00	42					
1	2	32	1	0.09	800	2200	22	0	0	28	0.00	99.91	99.91							
1	3	13	11	0.82	1295	2200	59	0	0	51	0.00	99.18	99.18	0.00	41					
1	3	33	2	0.16	989	2200	46	0	0	51	0.00	99.84	99.84							
1	4	14	0	0.04	800	2200	23	0	0	26	0.00	99.96	99.96	0.00	44					
1	4	34	0	0.06	801	2200	55	0	0	59	0.00	99.94	99.94							
1	8	15	0	0.02	800	2200	20	0	0	21	0.00	99.98	99.98	0.00	43					
1	8	35	1	0.16	800	2200	56	0	0	57	0.00	99.85	99.84							
1	9	16	1	0.07	800	2200	31	0	0	33	0.00	99.93	99.93	0.00	42					
1	9	36	1	0.07	800	2200	34	0	0	41	0.00	99.93	99.93							
1	10	17	0	0.04	800	2200	27	0	0	33	0.00	99.96	99.96	0.00	43					
1	10	37	1	0.13	800	2200	47	0	0	52	0.00	99.87	99.87							
1	11	18	1	0.10	800	2200	43	0	0	52	0.00	99.90	99.90	0.00	41					
1	11	38	1	0.18	800	2200	213	0	0	219	0.00	99.83	99.82							
1	12	19	1	0.10	800	2200	47	0	0	53	0.00	99.90	99.90	0.00	43					
1	12	39	2	0.30	800	2200	32	0	0	41	0.00	99.70	99.70							

Backup Slides

Running Benchmarks

- Latency critical applications are implemented using a microservice-based software architecture.
- Microservices are characterized by:
  - Strict latency requirements
  - Unpredictable active/idle periods
  - Sensitivity to killer microsecond overheads



- This demo: Focuses on Memcached
  - MicroSuite containerized version available online