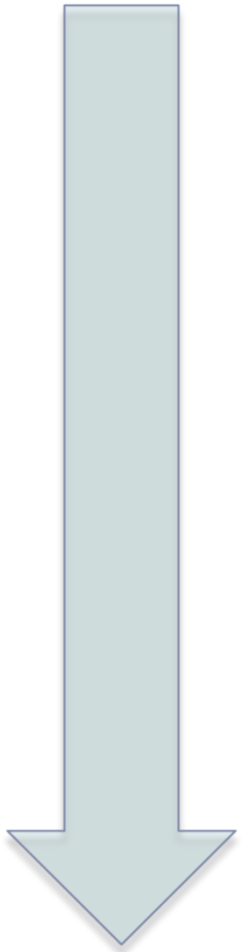
GreenMap: MapReduce with Ultra High Efficiency Power Delivery

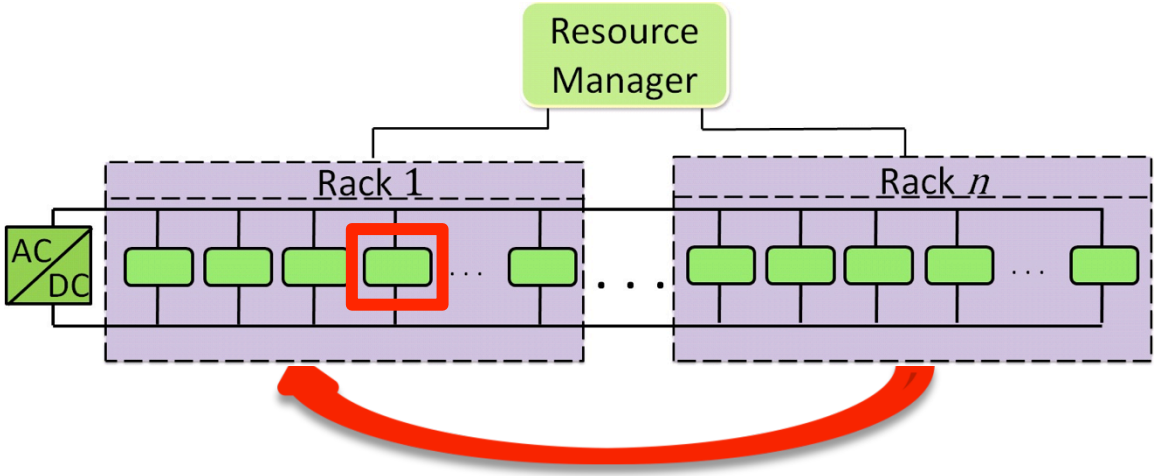
Du Su
University of Illinois at Urbana-Champaign
Advisor: Prof. Yi Lu

Power Consumption in Datacenters

- Online Services continue to grow
- Datacenters grow in size: Scale out (more servers) & Scale up (hardware upgrade)
- Large energy consumption of data centers
  - A significant fraction of the total cost of ownership (TCO).
  - Carbon emissions in 2007: 2-3% of the global carbon emissions
  - Electricity bills in 2009: $1,000,000 /month
  - Energy usage in 2013: 91 billion kwh
- Environmentally friendly data center

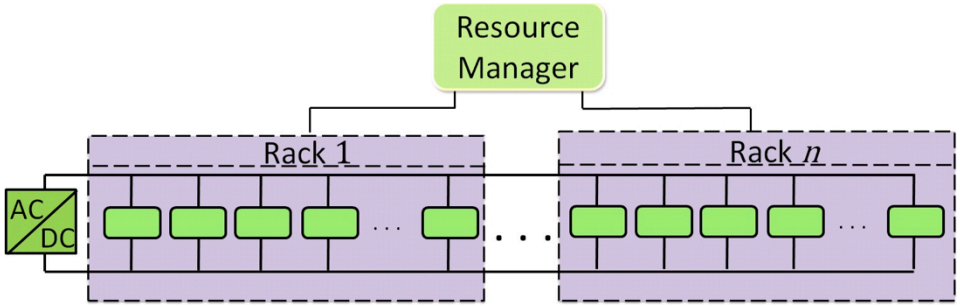


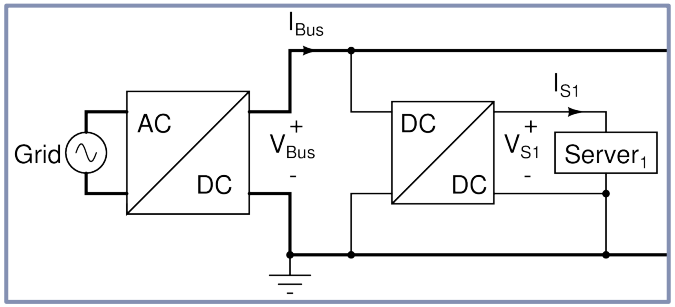
Previous Work



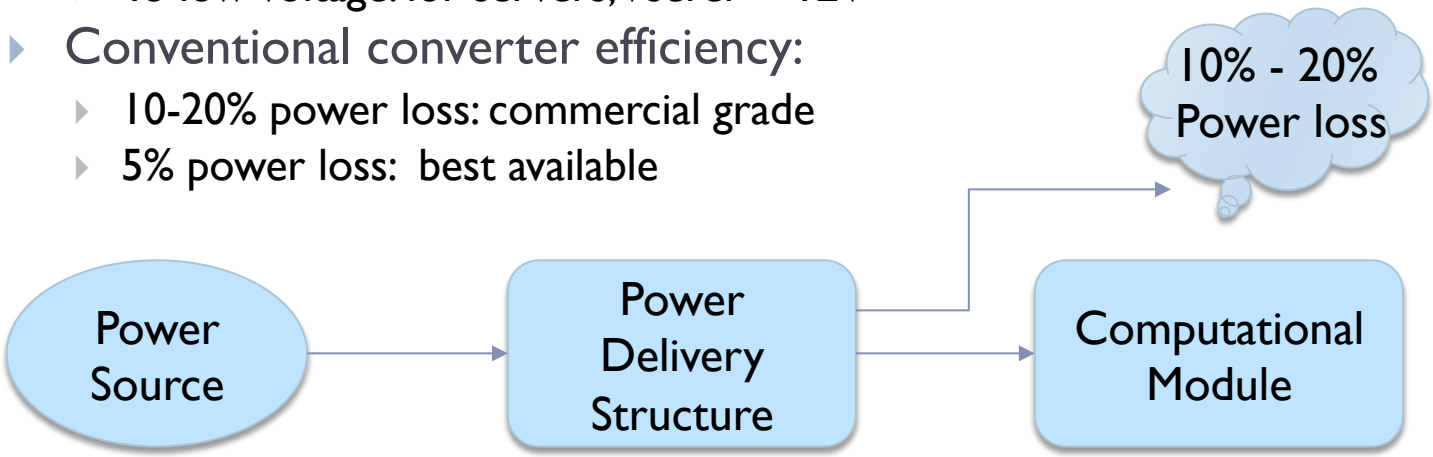
- Consolidate demand onto a small number of servers
  - Gandhi, A., et al '11, Krioukov, A., et al '10, Lin, M., et al '11
  - Method: request redirection or virtual machine migration
- Optimize individual power usage
  - Andrew, L., et al '10
  - Method: speed scaling each server

Conventional Stack: Problems

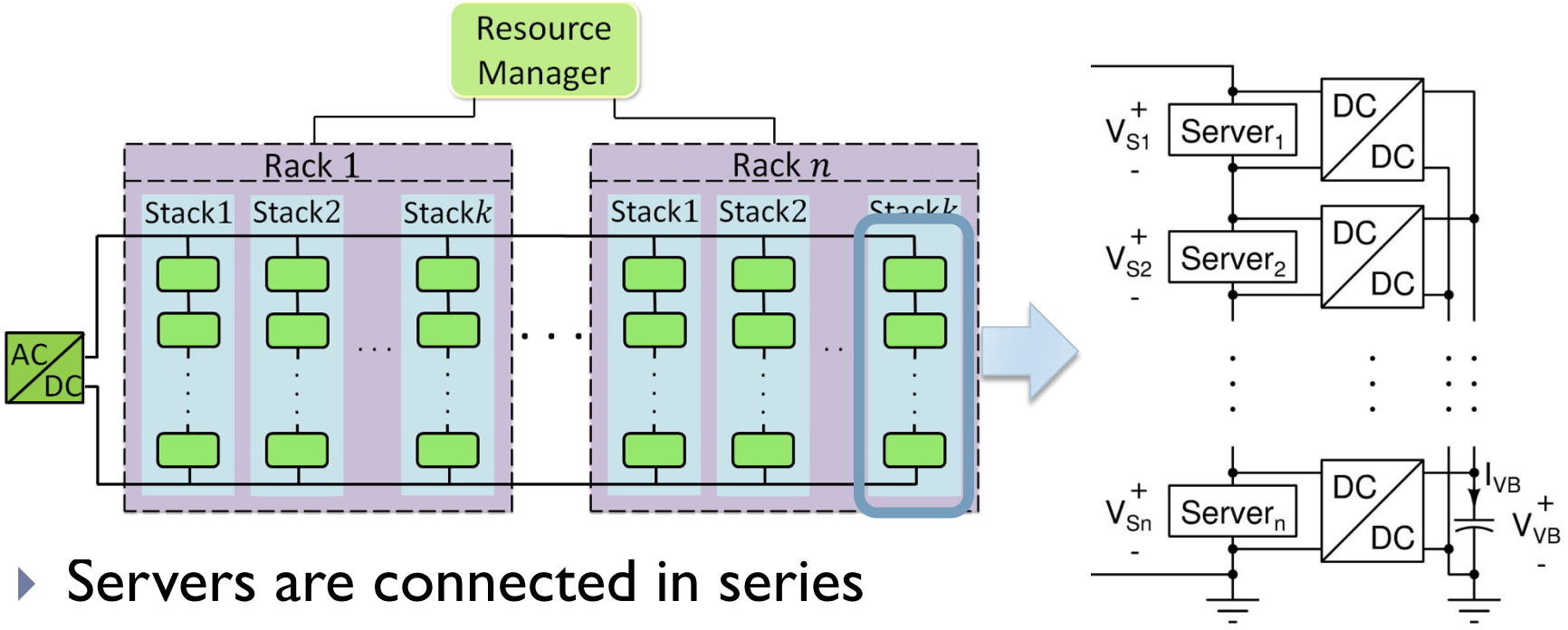




- Conventional stack's problem
  - Large step down:
    - From high voltage: distributed to racks, Vbus = 208 or 120V
    - To low voltage: for servers, Vserer = 12V
  - Conventional converter efficiency:
    - 10-20% power loss: commercial grade
    - 5% power loss: best available



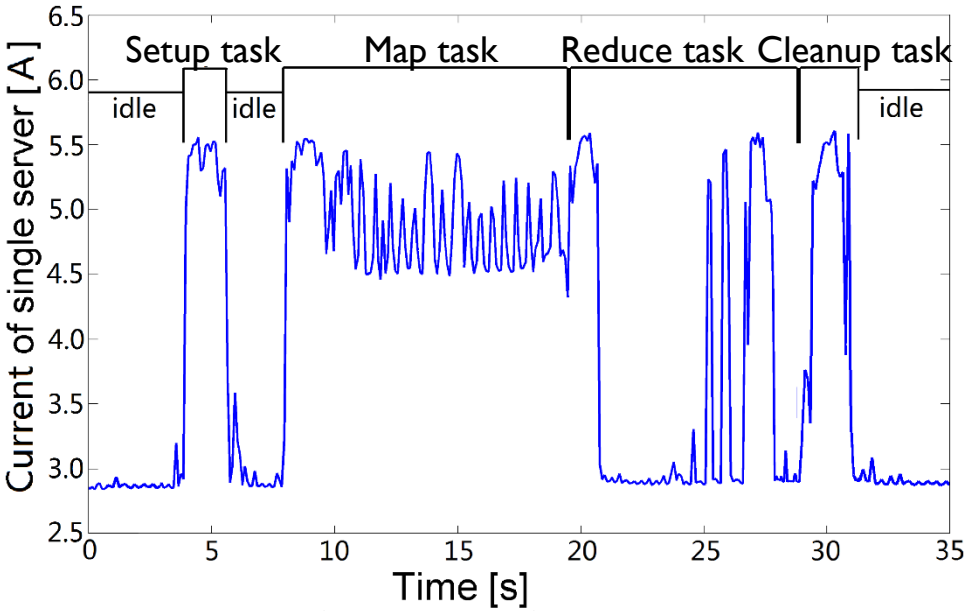
Series Stack: A New Power Delivery Architecture



- Servers are connected in series
  - Void large voltage step-down
- Differential power converters
  - Remove / compensate voltage
  - Power loss is proportional to imbalance of load
- Balance the load

Current Profiling: A Word-Count Job

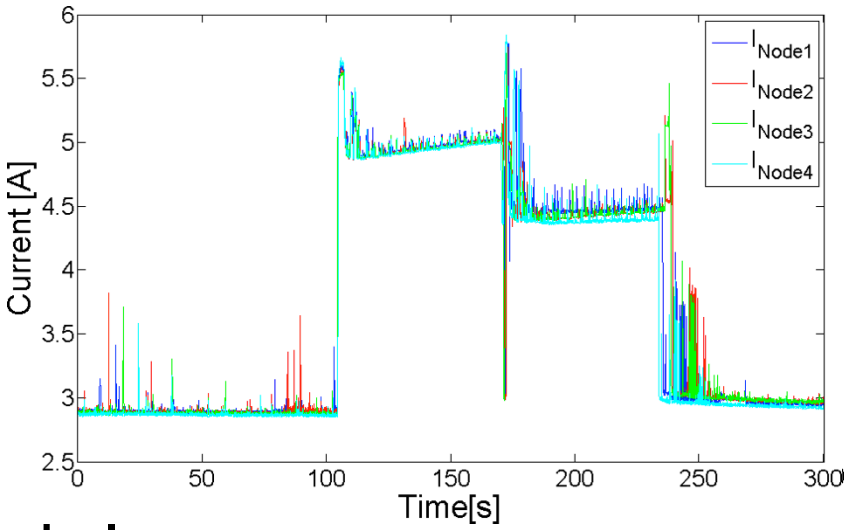
- Map task
  - Peak: initialization
  - Oscillation: generate <key, value>
- Reduce task
  - Peak: initialization
  - Low current: copy <key, value>
  - Oscillation: counting <key, value>
- Setup & Cleanup
  - Peak: initialization and clean up



- Map / Reduce task current consumption varies
  - Need synchronization
- One setup / cleanup task per job
  - Difficult to balance

GreenMap: Synchronized Task Assignment

- Map tasks: need synchronization
  - # tasks < # server w/ idle slots
    - Delay task assignment
  - # tasks ≥ # server w/ idle slots
    - Assign tasks in batch
    - Prefer tasks from the same job



- Setup and Cleanup tasks: hard to balance
  - Moved to RM, or
  - Moved to parallel connected servers
- Forever wait?
  - Set timeout
  - Refresh at new job arrival and synchronized assignment

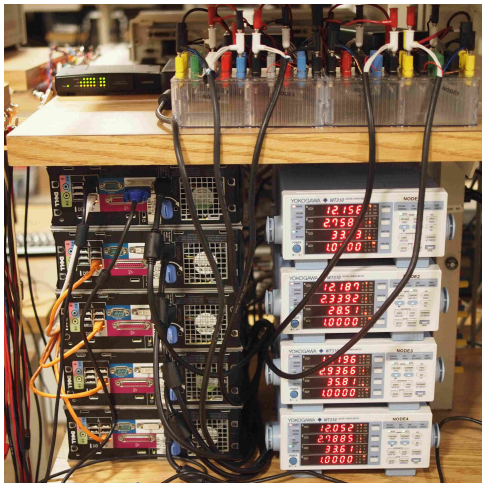
Evaluation: Setup

- System
  - Hadoop1
- Workstations
  - RM x1, outside series stack
  - series stack server x4, 48V series stack
  - Dell Optiplex SX775 Core 2 Duo workstations: 2 map slots / machine
- Trace
  - SWIM benchmark
    - Scale down
    - File block size: 32 MB
    - Identical map tasks ~70s
  - 50 jobs, 447 tasks
    - Poison arrival
    - Pareto size distribution

Bins	1	2	3	4	5	6
Job count	25	9	6	4	3	3
Map count per job	1	2	4	8	16	100

Evaluation: Measurement

- Equipment
  - Yokogawa wt310 digital power meter x4
  - 10 samples (I,V) / (second x server)
- Calculate power loss in converter
  - Conventional stack:



$$L_{\text{conv}} = (1-E)P = (1-E)V\sum_{i=1}^{n} I_i$$

$$\text{where } V = 12V,$$

$$E = \begin{cases} 0.8-0.9 & \text{for converters in data centers,} \\ 0.95 & \text{for best available converters.} \end{cases}$$

  - Series stack:

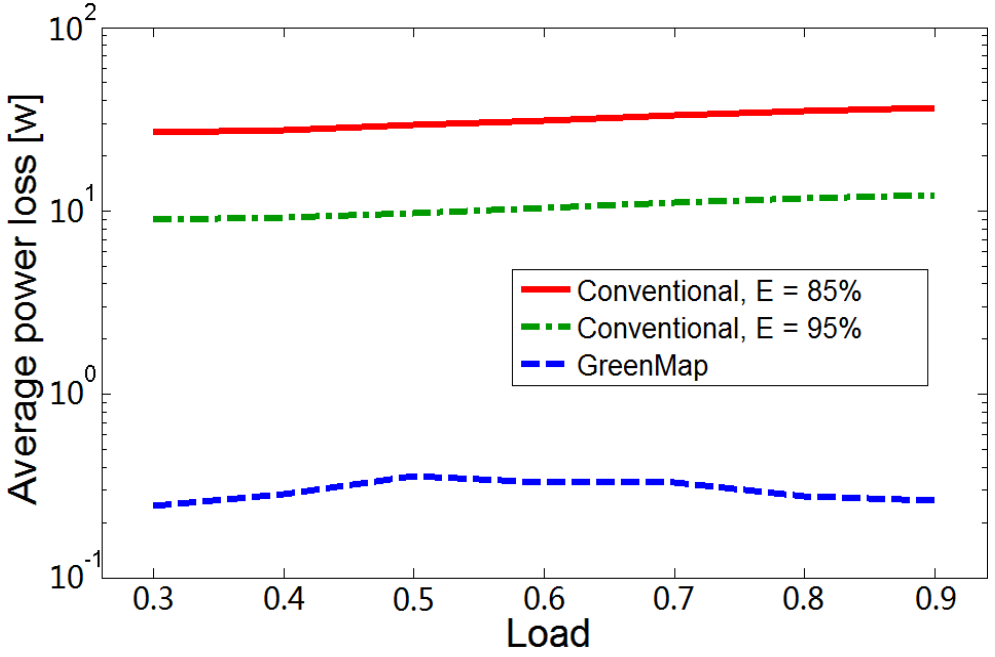
$$L_{\text{diff}} = 1.5(1-E)V\sum_{i=1}^{n}(I_i - I_{\text{avg}})$$

$$\text{where } V = 12V,\ E = 0.95,$$

$$I_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} I_i.$$

Evaluation: Result

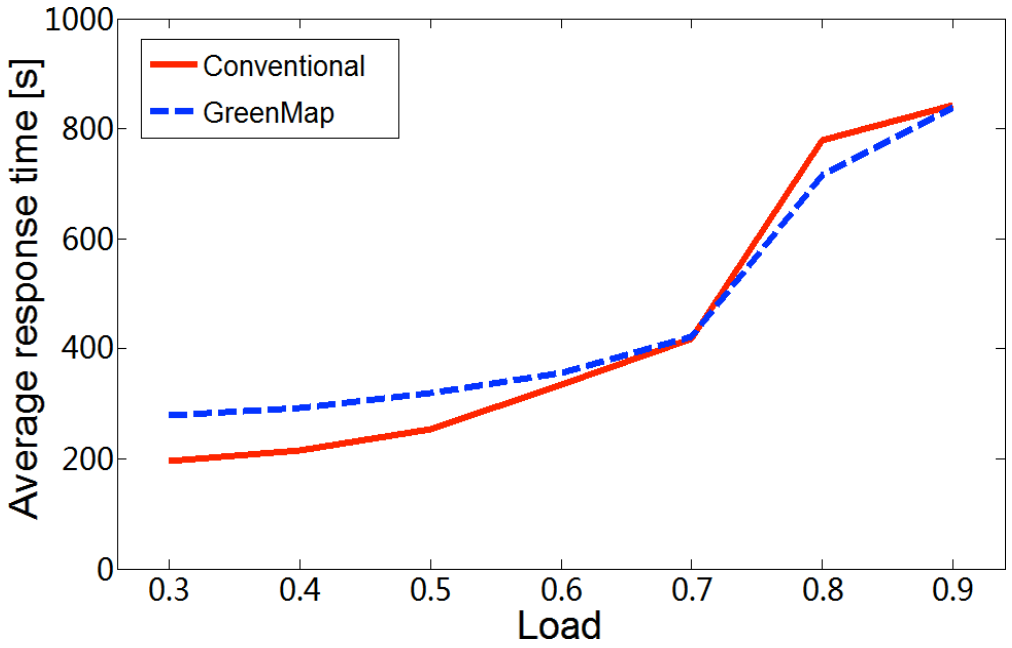
- Power loss in converter
  - Vs. commercial-grade conventional stack of 85% efficiency
    - 81x-138x reduction: two magnitude
    - Average Power loss from 31.4W to 0.3W
    - 14.999% reduction in total energy consumption.



  - Vs. best available conventional stack of 95% efficiency
    - 27x-46x reduction
    - From an average of 10.5W to 0.3W

Evaluation: Result (cont'd)

- Response time
  - Below 0.6 load
    - Increased by 26%-42%
  - Over 0.6 load
    - No obvious degeneration
    - Reason
      - Large number of tasks
      - Tasks are seldom delayed



- Deal with response time degeneration at low load?
  - Prefer to have system running at high load
  - Dynamic scaling

GreenMap with Dynamic Scaling

- Map tasks: need synchronization
- Setup and Cleanup tasks: hard to balance
- Forever wait?
- Dynamic scaling
  - Turn off a fraction of stacks, consolidate load
  - 10 stacks at 0.4 load
    - Total power: 1924.9W
  - 7 stacks at 0.57 load
    - Total power: 1294.4W: 32.8% redection
    - Response time: 15% increase

Conclusions

- Implementation
  - GreenMap: synchronized task assignment
- Performance
  - The conversion loss is reduced by two orders of magnitude
  - 15% of total energy consumption
- Future work
  - Implementing multiple series-stacks and heterogeneous jobs
  - Evaluating the system on actual series-connected stacks

Discussions

- Priority
  - Energy saving
  - Performance
- Granularity:
  - Large
    - Less power-loss; more imbalance
  - Small
    - Flexible constraints; potential power-loss
- Future work
  - Implementing multiple series-stacks and heterogeneous jobs
  - Evaluating the system on actual series-connected stacks