LA-UR-16-25919
Approved for public release; distribution is unlimited.

Title: Analysis of Trinity Power Metrics for Automated Monitoring

Author(s): Michalenko, Ashley Christine

Intended for: Share with my university

Issued: 2016-08-02

Analysis of Trinity Power Metrics for Automated Monitoring

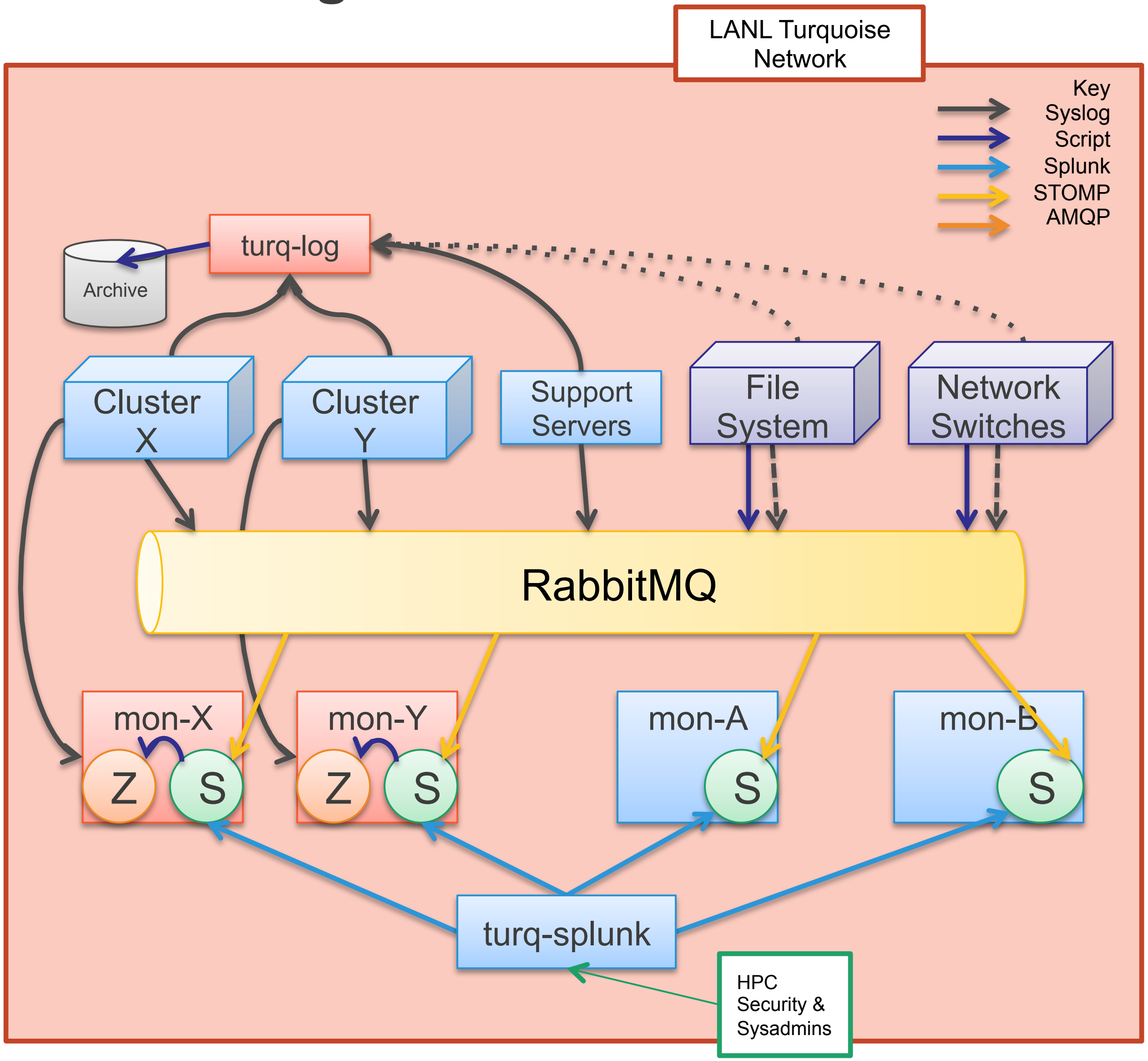
Ashley Michalenko

July 28, 2016

Outline

- Current monitoring efforts
- Motivation for analysis
- Tools used
- Methodology
- Summer work
- Future work

Current Monitoring Infrastructure



Motivation for enhanced monitoring

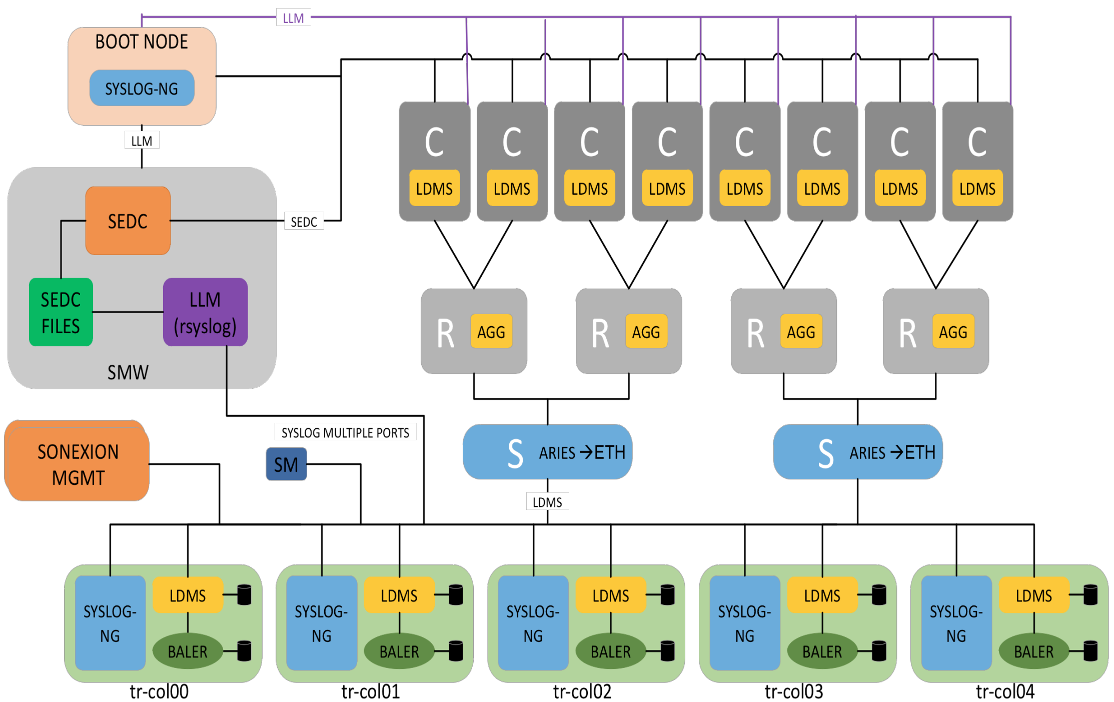
- Current monitoring efforts will not be sufficient for future machines
- These applications will not be able to handle the amount of data that Trinity and future systems will produce
- Admin’s do not have a full system view of statistics

Enhanced monitoring: LDMS

- Run LDMS on every compute node
- Continuous monitoring
- LDMS pulls detailed metrics about the systems state

LDMS

- Application out of SNL
- Collects metrics for memory, CPU, power, Cray Aries Network, etc.
- Set to collect once every 10 seconds
- Collected data on Trinity during open science period



Why Machine Learning?

- **Can handle large amounts of data**
  - Estimated to produce 4 TB/day of data
    - Compared to 40 GB/day of syslog
- **Automated anomaly detection**
- **Saves time**
- **There is no need to save all data, just data for events of interest**

What does LDMS output look like?

```
0,365253551,365253591,365253631,365253280,365253314,365253353,365253393,365253433,365253472,365253512,397,398,422,388,401,420,377,385,394,384
0,365672847,365672885,365672885,365672962,365673000,365672655,365672693,365672731,365672770,365672808,368,388,388,397,369,393,396,367,385,379
0,352091595,352091216,352091291,352091329,352091368,352091406,352091406,352091482,352091520,352091558,387,363,391,372,363,388,388,368,381,395
0,348161260,348160935,348160971,348161008,348161043,348161079,348161116,348161152,348161188,348161224,358,358,357,356,370,363,357,358,358,358
0,92949308,92949316,92949323,92949330,92949336,92949342,92949272,92949282,92949292,92949300,58,68,61,75,69,65,66,63,68,61
0,85345050,85345060,85345068,85345076,85345000,85345008,85345017,85345025,85345033,85345042,90,89,94,81,64,70,78,104,103,91
0,99310993,99311009,99311027,99311041,99311057,99310916,99310932,99310949,99310962,99310978,157,159,165,146,171,165,152,153,166,163
0,89773490,89773499,89773506,89773513,89773521,89773529,89773538,89773545,89773553,89773483,73,75,89,81,75,66,84,85,95,83
0,329840144,329840178,329840213,329840247,329839938,329839972,329840006,329840040,329840074,329840109,341,341,340,343,341,341,340,358,340,341
0,347877154,347876830,347876865,347876901,347876936,347876973,347877012,347877045,347877081,347877118,362,359,360,360,368,377,361,362,360,361
0,347874495,347874170,347874206,347874242,347874278,347874314,347874350,347874387,347874423,347874459,361,361,361,359,355,365,373,358,359,360
0,358905004,358905040,358905076,358905113,358905149,358905185,358904859,358904895,358904931,358904967,361,362,363,362,361,362,363,362,363,373
0,352587462,352587498,352587534,352587570,352587607,352587643,352587317,352587353,352587389,352587425,376,363,361,363,362,361,361,363,361,359
0,347277306,347277342,347277377,347277413,347277448,347277129,347277165,347277200,347277235,347277271,354,353,353,354,354,353,355,354,354,366
0,363576066,363576094,363576131,363575804,363575842,363575879,363575916,363575954,363575991,363576029,373,372,373,371,371,369,380,387,372,373
0,358824832,358824868,358824905,358824942,358824979,358825015,358825052,358824723,358824759,358824796,364,378,381,364,365,365,363,364,365,366
0,358163704,358163743,358163777,358163814,358163851,358163888,358163928,358163594,358163630,358163667,366,375,371,368,366,368,366,368,366,367
0,353252898,353252935,353252975,353252645,353252681,353252718,353252754,353252792,353252826,353252862,362,362,361,361,361,357,374,364,360,362
0,356026690,356026728,356026762,356026798,356026834,356026871,356026910,356026943,356026618,356026654,359,365,367,362,362,361,361,360,360,360
0,347532920,347532588,347532624,347532660,347532697,347532735,347532770,347532806,347532843,347532880,365,365,365,363,372,381,362,364,363,365
0,345997337,345997372,345997408,345997444,345997121,345997157,345997193,345997229,345997264,345997300,359,357,359,358,356,356,356,372,359,358
0,56701460,56701460,56701460,56701460,56701460,56701460,56701482,56701482,56701482,56701460,44,44,44,44,44,44,44,44,44,44
0,55194081,55194081,55194081,55194102,55194102,55194102,55194102,55194059,55194059,55194081,44,44,44,44,44,44,44,44,44,44
0,354353429,354353100,354353137,354353178,354353209,354353246,354353282,354353319,354353356,354353392,361,360,363,362,364,379,363,362,364,363
0,353747701,353747738,353747775,353747811,353747481,353747518,353747554,353747591,353747627,353747664,367,368,367,367,366,365,365,368,376,366
0,342111659,342111694,342111730,342111765,342111447,342111482,342111517,342111553,342111588,342111623,349,349,349,350,350,350,347,360,353,349
0,340514105,340514141,340514176,340514211,340514247,340514283,340514319,340514354,340514035,340514070,349,367,359,349,350,350,349,351,348,350
0,341686981,341687017,341687054,341687089,341687125,341687162,341687198,341687234,341686910,341686945,357,360,367,358,360,361,359,356,357,356
0,338511940,338511975,338512011,338512045,338512081,338511765,338511800,338511835,338511870,338511905,354,346,347,346,346,347,347,347,343,361
0,349021720,349021756,349021792,349021828,349021542,349021542,349021583,349021613,349021648,349021684,355,355,357,355,353,355,354,363,357,357
0,344437570,344437606,344437642,344437678,344437353,344437389,344437425,344437461,344437497,344437534,356,356,357,357,358,358,357,353,375,376
                                                                                                                          39,1          0%
```

LDMS Headers

```
#Time, Time_usec, ProducerName,component_id,job_id,energy0,energy1,energy2,energy3,energy4,energy5,energy6,energy7,energy8,energy9,energy_timeval0,energ
y_timeval1,energy_timeval2,energy_timeval3,energy_timeval4,energy_timeval5,energy_timeval6,energy_timeval7,energy_timeval8,energy_timeval9,energy_timeva
l10,energy_timeval11,energy_timeval12,energy_timeval13,energy_timeval14,energy_timeval15,energy_timeval16,energy_timeval17,energy_timeval18,energy_timev
al19,power0,power1,power2,power3,power4,power5,power6,power7,power8,power9,power_timeval0,power_timeval1,power_timeval2,power_timeval3,power_timeval4,po
wer_timeval5,power_timeval6,power_timeval7,power_timeval8,power_timeval9,power_timeval10,power_timeval11,power_timeval12,power_timeval13,power_timeval14
,power_timeval15,power_timeval16,power_timeval17,power_timeval18,power_timeval19
~
~
~
```

```
#Time, Time_usec, ProducerName,component_id,job_id,aries_rtr_id,AR_NIC_NETMON_ORB_EVENT_CNTR_REQ_PKTS,AR_NIC_NETMON_ORB_EVENT_CNTR_REQ_FLITS,AR_NIC_NETM
ON_ORB_EVENT_CNTR_REQ_STALLED,AR_NIC_RSPMON_PARB_EVENT_CNTR_PI_PKTS,AR_NIC_RSPMON_PARB_EVENT_CNTR_PI_FLITS,AR_NIC_RSPMON_PARB_EVENT_CNTR_PI_STALLED,AR_N
IC_RSPMON_PARB_EVENT_CNTR_AMO_PKTS,AR_NIC_RSPMON_PARB_EVENT_CNTR_AMO_FLITS,AR_NIC_RSPMON_PARB_EVENT_CNTR_AMO_BLOCKED,AR_NIC_RSPMON_PARB_EVENT_CNTR_WC_PK
TS,AR_NIC_RSPMON_PARB_EVENT_CNTR_WC_FLITS,AR_NIC_RSPMON_PARB_EVENT_CNTR_WC_BLOCKED,AR_NIC_RSPMON_PARB_EVENT_CNTR_BTE_RD_PKTS,AR_NIC_RSPMON_PARB_EVENT_CN
TR_BTE_RD_FLITS,AR_NIC_RSPMON_PARB_EVENT_CNTR_BTE_RD_BLOCKED,AR_NIC_RSPMON_PARB_EVENT_CNTR_IOMMU_PKTS,AR_NIC_RSPMON_PARB_EVENT_CNTR_IOMMU_FLITS,AR_NIC_R
SPMON_PARB_EVENT_CNTR_IOMMU_BLOCKED
~
~
```

Data Analytics Accelerated Library - DAAL

- **Library created by Intel**
- **Designed and optimized for Intel processors (such as Trinity)**
- **Contains a set of basic machine learning algorithms**
- **MPI Based**
  - Preferred over other distributed configurations

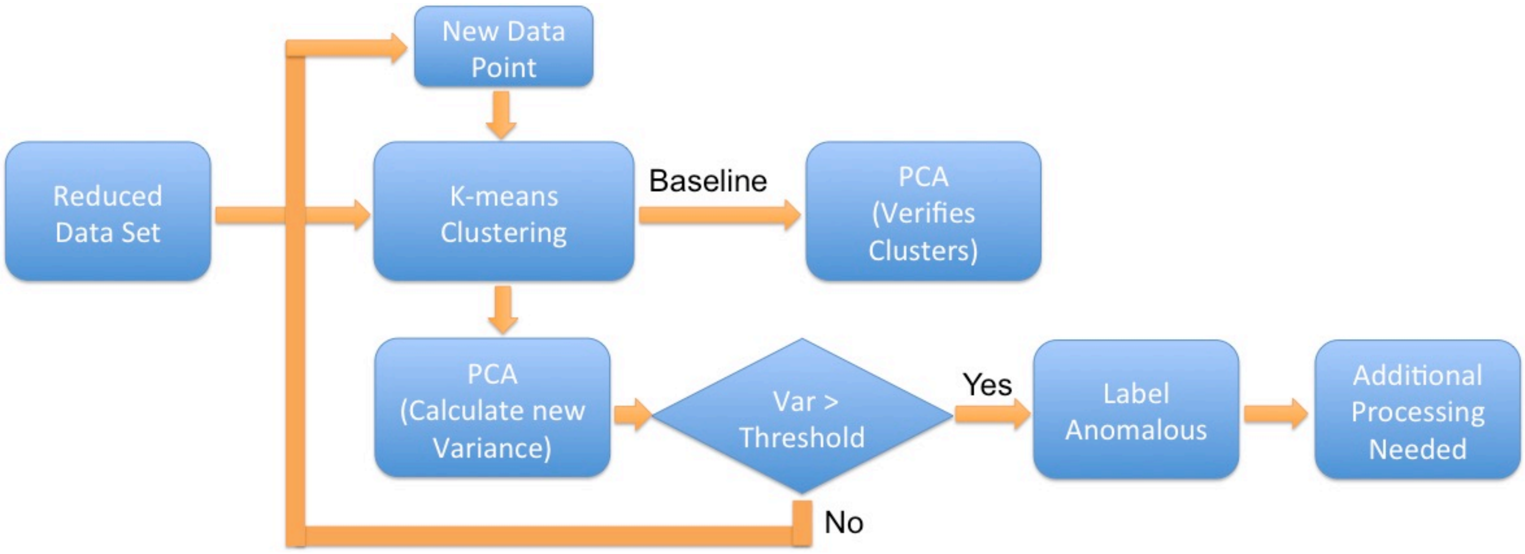
DAAL Cont.

- **Excerpt from a k-means example file**

```
        /* Retrieve the data from the input file */
        dataSource.loadDataBlock();

        /* Create an algorithm object for the K-Means algorithm */
        kmeans::Distributed<step1Local> localAlgorithm(nClusters, it == nIterations);

        /* Set the input data to the algorithm */
        localAlgorithm.input.set(kmeans::data,           dataSource.getNumericTable());
        localAlgorithm.input.set(kmeans::inputCentroids, centroids);

        localAlgorithm.compute();

        if( it == nIterations )
        {
            localAlgorithm.finalizeCompute();
            assignments[i] = localAlgorithm.getResult()->get(kmeans::assignments);
        }
        else
        {
            masterAlgorithm.input.add(kmeans::partialResults, localAlgorithm.getPartialResult());
        }
    }

    if( it == nIterations ) break;

    masterAlgorithm.compute();
    masterAlgorithm.finalizeCompute();
```

Methodology

- **First step is to reduce the data set**
  - Remove information that is not necessary for baseline analysis
- **Use two algorithms**
  - K-means clustering
    - Cluster based on node activity
  - Principal Component Analysis
    - Use variance to detect anomalies and check that clustering is working correctly

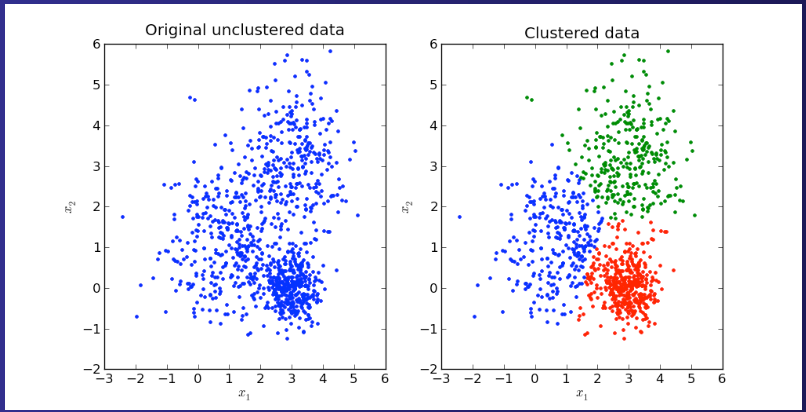


Reducing Dimensions

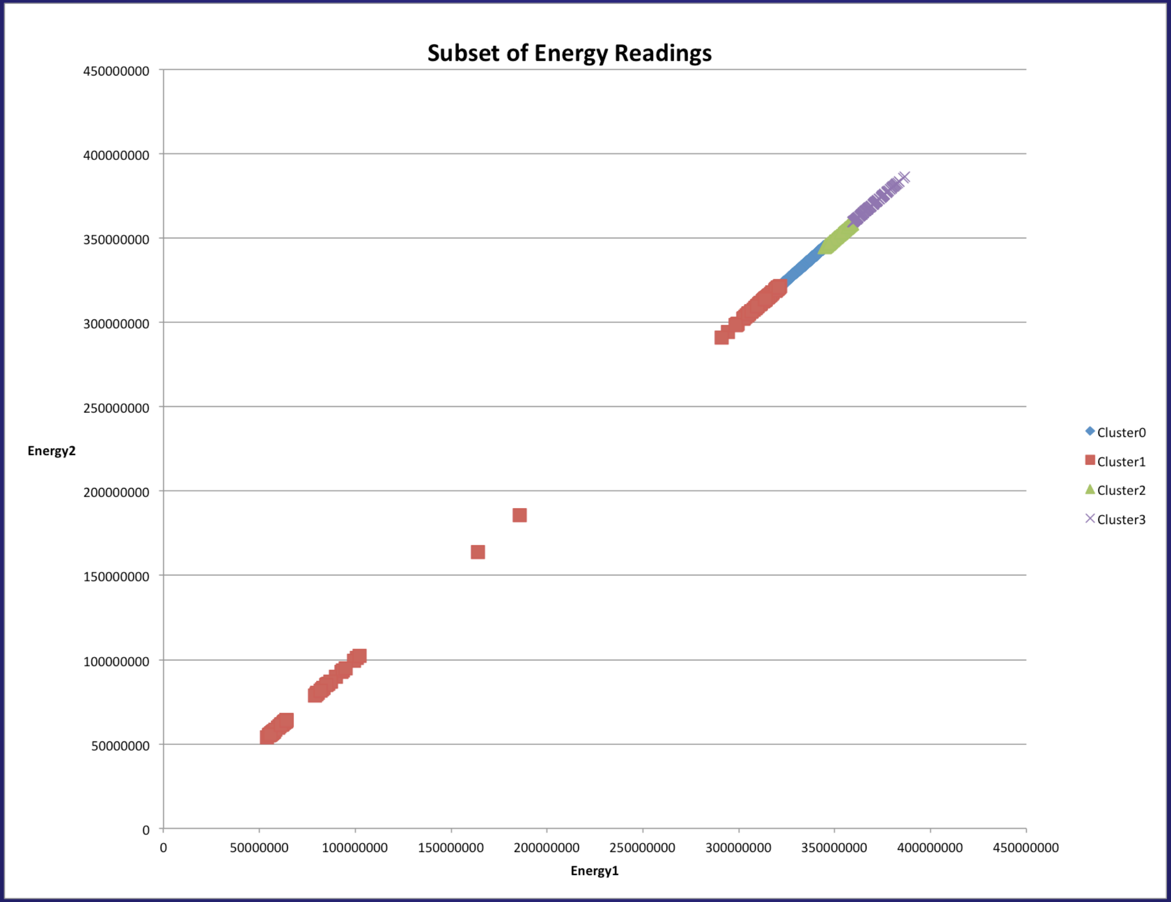
- **Remove unnecessary features**
  - Timestamps and node descriptors
- **When looking at Cray power sampler we reduce the dimensions of the data set from 65 to 20**
- **The reduced file size allows for faster analysis**

K-means Clustering

- **Cluster together similar data points**
- **User can choose how many clusters to make**
- **We expect to form clusters based on node activity**
  - Active vs. idle node
  - Type of activity/process on the node
  - Power consumption should be different between these types of applications

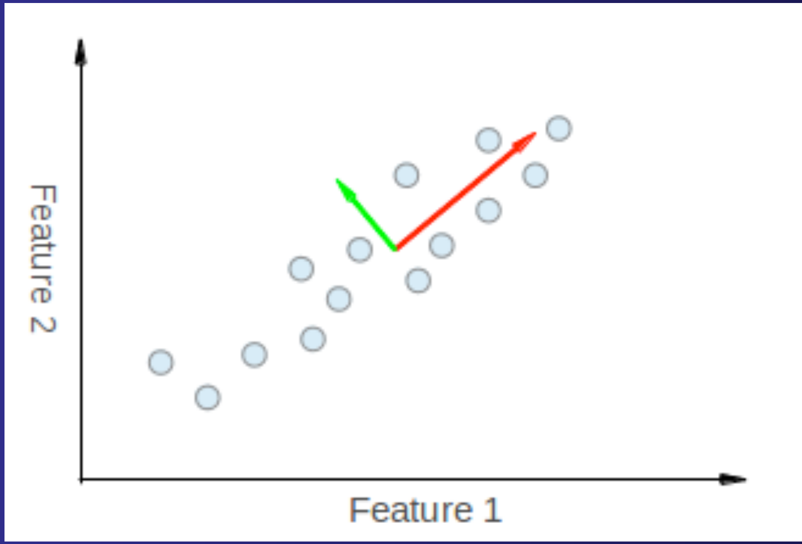


K-means applied to LDMS Power Data



Principal Component Analysis - PCA

- **Measures variance of dimensions within a data set**
- **Normally used to reduce dimensions**
- **Perform PCA on each of the clusters**
  - Use variance to determine if we have clustered correctly
  - In future work use variance to detect anomalies



Baseline

- **Goal is to use these algorithms to create a baseline for the system**
  - Model performance throughout the day
  - Baseline can be compared with future data to determine anomalies

Goal of Summer work

- **Become familiar with DAAL, machine learning algorithms**
- **Work with MPI, Woodchuck**
- **Compute Trinity power consumption baselines**

Future work

- **Continue developing baselines**
- **Expand analysis to other datasets**
  - Memory, CPU, power, Cray Aries Network, etc.
- **Analyze efficiency of DAAL for our work**
  - Compare with other distributed set ups
  - Compare with other ML algorithms

Questions?