Application-Server Matching (ASM)

William Strickland

4/8/2010
COT6410

Outline

- Abstraction
  - Key Terms
  - Assumptions
  - Informal Definition
  - Formal Definition
- Problem Difficulty
  - Proof of NP
  - Proof of NP-C

Outline cont.

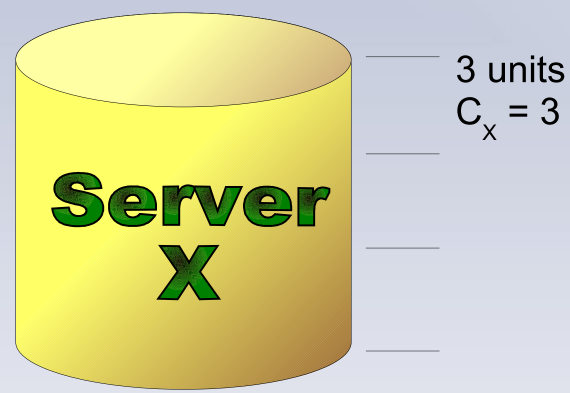
- Restricted Instance Analysis
    - Aspects of Problem
    - Restricted Set 1
    - Restricted Set 2
    - Restricted Set 3
- Conclusions

Key Terms

- Servers
  - Physical machines that have some capacity for computational work.
  - Will support some operating systems but not others.
- Applications
  - Each placed and run on some server.
  - Requires some amount of regular computation (workload).
  - Requires a operating system be installed on the server that this application execute under.

Key Terms cont.

- Operating System
  - A distinct collection of base system software.
  - Can be installed on a server (given it is compatible).
  - Enables sets of applications to execute.



Assumptions

- Server performance can be quantified in such that relations hold for all applications.
- Applications require a constant amount of computation per unit of time.
- All applications run (truly) concurrently.
- No usage of dynamic virtual server usage.
- Operating Systems are licensed on a per server basis (not processor/core/user/transactions)

Informal Definition

- Given:
  - Sets of OS, servers, applications, pairs of servers and OS, pairs of applications and OS, license counts, workloads and capacities
- Question: Can the applications be placed on servers such that:
  - All applications are placed.
  - Each application is placed on exactly one server.
  - No server capacity is exceeded by the workload of applications.
  - No application is running on a server with a OS not supported.
  - No server is running a OS it cannot support.
  - No server has been run more than one OS.
  - License count for each OS has not been exceeded.

Formal Definition

- Given
  - Set of operating systems, O
  - Set of servers, S
  - Set of applications, A
  - Set of server-os pairs, U
  - Set of app-os pairs, V
  - Set of license counts, L
  - Set of server capacities, C
  - Set of application workloads, W

- Question:
  - Does there exist a set of triples, K, such that:
    - $(o, s, a) \in K \; where \; o \in O, s \in S, a \in A$
    - $\forall K_i: \; \exists \; (s_i, o_i) \in U, \; (a_i, o_i) \in V$
    - $\forall a \in A \; \exists \; (o, s, a) \in K$
    - $\forall K_1 = (o_1, s_1, a_1), K_2 = (o_2, s_2, a_2) \in K$ $\quad if \; a_1 = a_2 \; then \; o_1 = o_2, s_1 = s_2$
    - $\forall K_1 = (o_1, s_1, a_1), K_2 = (o_2, s_2, a_2) \in K$ $\quad if \; s_1 = s_2 \; then \; o_1 = o_2$
    - $\forall s \in S, \sum (W_s) \leq C_s \; where \; W_s = W_a \; iff \; \exists (o, s, a) \in K$
    - $\forall o \in O, \sum (L_s) \leq L_o \; where \; L_s = 1 \; iff \; \exists (o, s, a) \in K$

Proof of NP class

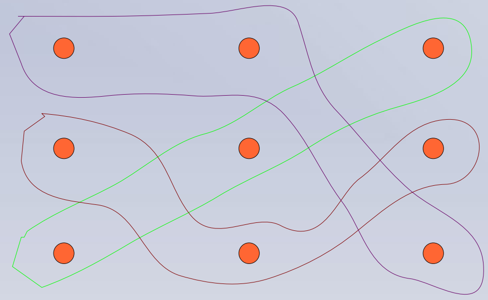
- ASM is a decision problem
- If given set of triples, K, as witness
- Verify yes instance in polynomial time
    - Loop through K
        - Verify S-O and A-O pairs exist
        - Verify all Applications placed exactly once
        - Verify OS licenses not exceeded
        - Verify capacities not exceeded
        - Verify no server assigned two OS

Proof NP Complete

- ASM shown NP
- Use known NP-C, 3-Dimensional Matching
- Show transformation from 3-DM to ASM
- Prove correctness of transformation

3-DM defined

- Given:
  - Set X, Set Y, Set Z
  - Set of triples T, $where\,(x, y, z) \in T, x \in X, y \in Y, z \in Z$
- Question:
  - Does there exist $M \subseteq T$
  - $\forall x \in X, \exists (x, y, z) \in M$
  - $\forall y \in Y, \exists (x, y, z) \in M$
  - $\forall z \in Z, \exists (x, y, z) \in M$
  - $\forall M_1, M_2 \in M, x_1 \neq x_2, y_1 \neq y_2, z_1 \neq z_2$



- Transformation $3\mathrm{DM} \underset{P}{\Rightarrow} ASM$

  1) Accept 3-DM instance; (X, Y, Z, T)

  2) Create new sets for ASM

     $O' = \emptyset,\ S' = \emptyset,\ A' = \emptyset,\ U' = \emptyset,$
     $V' = \emptyset,\ L' = \emptyset,\ C' = \emptyset,\ W' = \emptyset$

  3) $\forall x \in X:\ O' = O' \cup \{x\},\ L'_x = 1$

  4) $\forall y \in Y:\ S' = S' \cup \{y\},\ C'_y = 1$

  5) $\forall z \in Z:\ A' = A' \cup \{z\},\ W'_x = 1$

  6) $\forall t = (x, y, z) \in T:\ U' = U' \cup \{(y, x)\};\ V' = A' \cup \{(z, x)\}$

  7) answer ASM (O',S',A',U',V',L',C',W')

- Transformation proof key points
  - If 3-DM is yes
    - Exists M → K;
    - Guarantied for K_i, x, y, and z unique
    - Guarantied all Os, Server, and App exactly once
    - Guarantied OS pairs, T → U and T → V
    - ASM yes if 3-DM yes

- For constructed form ASM instance, if yes
    - Because $L_i=C_i=W_i=1$, must use every OS, Server and App exactly once
    - Must exist $T_i=(x,y,z)$ if $U_i=(y,x)$ and $V_i=(z,x)$
    -
    - If ASM yes then 3-DM is yes
- ASM is yes iff 3-DM is yes

- ASM is NP-complete

Aspects of Problem

- Two sources of difficulty in this problem:
  - Matching Elements on criteria (OS-Server-App).
  - Allocating units of work to units of computation.
- Whole problem shown to be NP-C.
- Instance set will be restricted to isolate sources of difficulty.
- For different instances, one or both parts may be trivial.

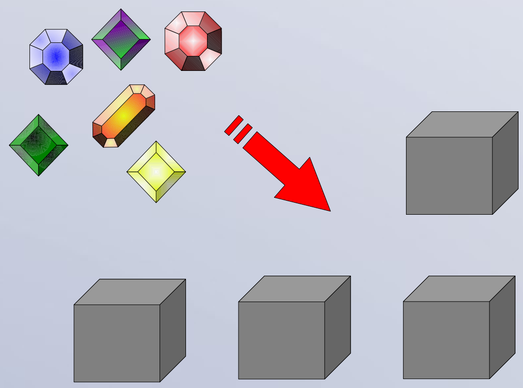
Restricted Set 1

- Removing the task allocation difficulty
  - Setting Server capacity to 1
  - Setting Application workload to 1
  - Setting OS license limit to 1
- This has already been done through the construction from 3-DM
- With trivial allocation, problem remains NP-C

Restricted Set 2

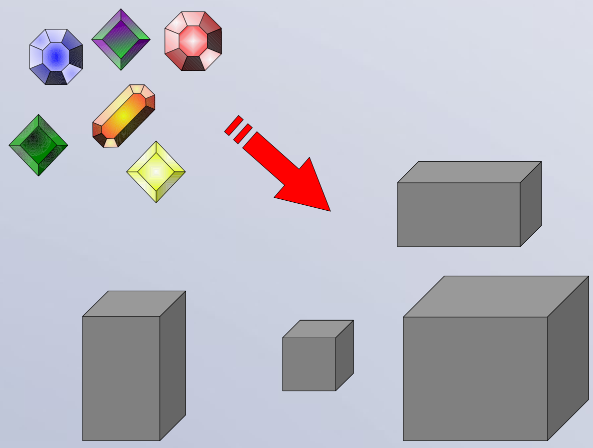
- Remove matching components
  - All applications support a single OS
  - All Servers support a single OS
  - License count for this OS = |S|
- Trivial to match Server and Application with OS
- Remaining problem similar to Bin Packing

Bin Packing Defined



- Given:
  - Set of item sizes, A
  - Bin size, V
  - Number of bins, B
- Question:
  - Can all items in A be placed in a bin such that for each bin S {1...B} $\sum A_S \leq V$ where is A_s is an item in bin S.
- Problem known to be NP-C.

Assorted Bin Packing Defined



- Given:
  - Set of item sizes, A
  - Set of bin sizes, V
  - Number of bins, B
- Question:
  - Can all items in A be placed in a bin such that for each bin S {1...B} $\sum A_S \leq V_S$ where is A_s is an item in bin S.
- Problem NP-C from Bin Packing. All Bin packing instances are instances of Assorted Bin Packing.

Redefine problem

- Problem redefined to match new set of instances
- Given:
  - Set of servers, S
  - Set of applications, A
  - Set of server capacities, C
  - Set of application workloads, W
- Question: Does there exist a set pair such that

$$(s,a)\in K \; where \; s\in S, a\in A$$
$$\forall a\in A \;\; \exists \;\; (s,a)\in K$$
$$\forall K_1=(s_1,a_1), K_2=(s_2,a_2)\in K$$
$$if \;\; a_1=a_2 \;\; then \;\; s_1=s_2$$
$$\forall s\in S, \sum (W_s)\leq C_s \;\; where \; W_s=W_a \;\; iff \;\; \exists (s,a)\in K$$

- Transformation $ABP \underset{P}{\Rightarrow} ASM_2$

    1) Accept ABP instance; (A, V, B)

    2) Create new sets for ASM

    $S' = \emptyset,\ A' = \emptyset,\ C' = \emptyset,\ W' = \emptyset$

    3) $\forall v \in V:\ S' = S' \cup \{v\},\ C'_v = v$

    4) $\forall a \in A:\ A' = A' \cup \{a\},\ W'_a = a$

    5) answer ASM_2 (S',A',C',W')

- 'Proof' by restriction:
  - Created ASM_2 instance constructed as follows
    - $V \rightarrow S'$
    - $V \rightarrow C'$
    - $A \rightarrow A'$
    - $A \rightarrow W'$
  - True if placed |A| applications (items) onto |V| servers (bins) and for each server s

    $\sum (W_s) \leq C_s$ (placed weight is less than capacity)

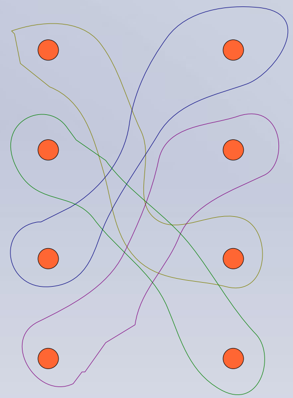
Restricted Set 3

- Removing the task allocation difficulty and some matching difficulty
  - Setting Server capacity to 1
  - Setting Application workload to 1
  - Setting OS license limit to 1
  - Set each Application to support exactly 1 OS
- Same as restriction 1 but with direct correlation between OS and Application
- Becomes Polynomial with these restrictions

- 1-to-1 relation between Application and OS.
- Now only required to match OS to compatible server.
- Same result if forced instead 1-to-1 related Server and OS.
- Server and App are modeled the same.
- Holds without loss of generality.

2-DM Defined

- Given:
  - Set X, Set Y
  - Set of pairs T, $where\,(x, y) \in T,\, x \in X,\, y \in Y$
- Question:
  - Does there exist $M \subseteq T$
  - $\forall x \in X,\ \exists\,(x, y) \in M$
  - $\forall y \in Y,\ \exists\,(x, y) \in M$
  - $\forall M_1, M_2 \in M,\ x_1 \neq x_2,\ y_1 \neq y_2$



- Transformation $ASM_3 \underset{P}{\Rightarrow} 2\mathrm{DM}$
    1) Accept ASM_3 (O,S,A,U,V,L,C,W)
    2) $O \rightarrow X'$
    3) $S \rightarrow Y'$
    4) $\forall (a, o) \in V \; if \; \exists (s, o) \in V$ add (s,a) to T'
    5) answer 2DM (X', Y', T')

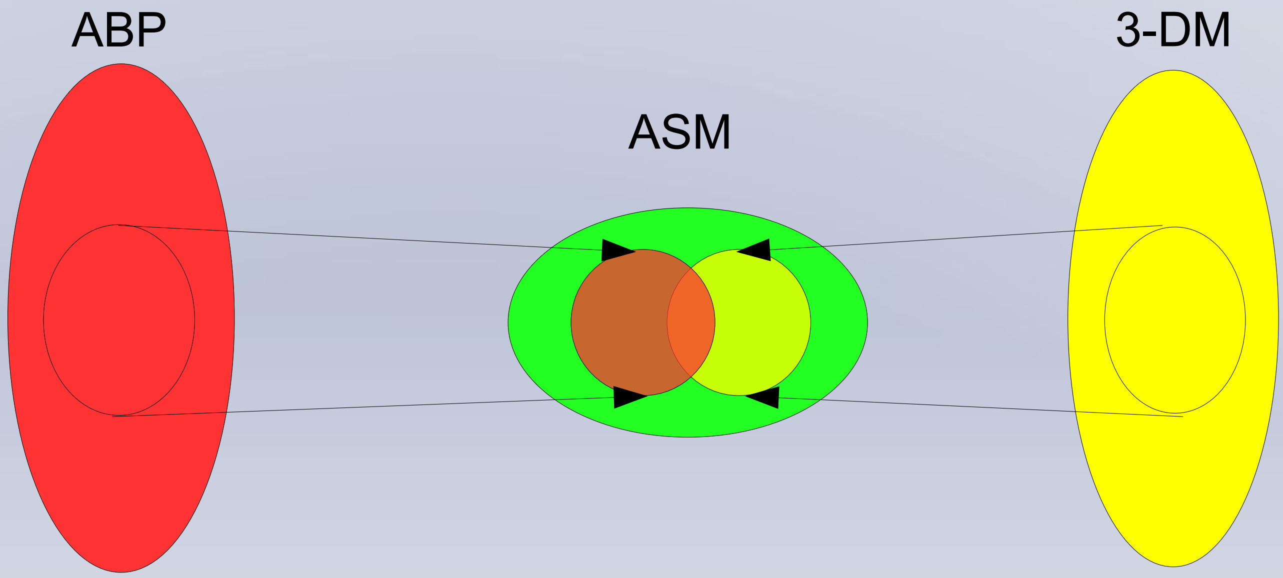
- If ASM_3 is yes
  - Exists $K \rightarrow M$;
  - Guarantied for M_i, x and y unique
  - Guarantied all x and y exactly once
  - Guarantied OS pairs, $UxV \rightarrow T$
  - 2-DM Yes if ASM_3 yes
- For constructed form 2-DM instance, if yes
  - Must use all x and y,
  - Must exist $U_i=(x,o)$ and $V_i=(y,o)$ if $T_i=(x,y)$

- 2-DM is yes iff ASM_3 is yes
- ASM_3 is no harder than 2-DM
- 2-DM is polynomial so ASM_3 is polynomial

Conclusions

- As a whole, Application-Server Matching problem is a NP-Complete
- The problem actually has two aspects that make it difficult
  - Matching OS on applications and server
  - Application Workload Allocating
- Only if both are made trivial the instance of ASM can be solved in polynomial time.

Hard Instance Reduction



References

- Garey, Michael R., and Johnson, David S. Computers and Intractability. New York: W. H. Freeman and Company, 1979