**Performance Optimizations for Your Power Apps Code**

Your current Power Apps code has a few inefficiencies that can be optimized for better performance. Here are some key improvements:

**1. Reduce Multiple ParseJSON Calls**

You are repeatedly calling ParseJSON(Text(JITSitem.CaseJSON)) multiple times. Instead, parse the JSON once and store it in a variable.

**Optimized Approach:**

* Extract JITSitem.CaseJSON once into a variable.
* Reference this variable instead of repeatedly parsing.

**Implementation:**

Modify your ForAll loop:

PowerApps

CopyEdit

ForAll(colJITSv3raw As JITSitem,

With({parsedJSON: ParseJSON(Text(JITSitem.CaseJSON))},

Collect(colJITSv4, {

ID: JITSitem.ID,

JITScontrol: JITSitem.JITScontrol,

CaseTitle: JITSitem.CaseTitle,

Customers: ForAll(Table(parsedJSON.Customers) As JSONitem, {

DisplayName: Text(JSONitem.Value.DisplayName),

Email: Text(JSONitem.Value.Email),

Claims: Text(JSONitem.Value.Claims),

Department: Text(JSONitem.Value.Department),

JobTitle: Text(JSONitem.Value.JobTitle),

Picture: Text(JSONitem.Value.Picture)

}),

CustomerOrg: { Value: Text(parsedJSON.CustomerOrg) },

VIP: { Value: Text(parsedJSON.VIP) },

Priority: { Value: If(IsBlank(parsedJSON.Priority), "3 - Standard", Text(parsedJSON.Priority)) },

AsgnToOrg: Text(parsedJSON.AsgnToOrg),

AsgnToLead: {

DisplayName: Text(parsedJSON.AsgnToLead.DisplayName),

Email: Text(parsedJSON.AsgnToLead.Email),

Claims: Text(parsedJSON.AsgnToLead.Claims),

Department: Text(parsedJSON.AsgnToLead.Department),

JobTitle: Text(parsedJSON.AsgnToLead.JobTitle),

Picture: Text(parsedJSON.AsgnToLead.Picture)

},

AsgnToPersonnel: ForAll(Table(parsedJSON.AsgnToPersonnel) As JSONitem, {

DisplayName: Text(JSONitem.Value.DisplayName),

Email: Text(JSONitem.Value.Email),

Claims: Text(JSONitem.Value.Claims),

Department: Text(JSONitem.Value.Department),

JobTitle: Text(JSONitem.Value.JobTitle),

Picture: Text(JSONitem.Value.Picture)

}),

Status: Text(parsedJSON.Status),

CaseHold: { Value: If(IsBlankOrError(parsedJSON.CaseHold), "-", Text(parsedJSON.CaseHold)) },

CaseDetermination: { Value: Text(parsedJSON.CaseDetermination) },

CompletedDate: DateValue(Text(parsedJSON.CompletedDate)),

CompletedBy: {

DisplayName: Text(parsedJSON.CompletedBy.DisplayName),

Email: Text(parsedJSON.CompletedBy.Email),

Claims: Text(parsedJSON.CompletedBy.Claims),

Department: Text(parsedJSON.CompletedBy.Department),

JobTitle: Text(parsedJSON.CompletedBy.JobTitle),

Picture: Text(parsedJSON.CompletedBy.Picture)

},

CaseDescription: JITSitem.OriginNotes,

Created: JITSitem.Created,

CreatedBy: JITSitem.CreatedBy,

Modified: JITSitem.Modified,

ModifiedBy: JITSitem.ModifiedBy,

SurveyJSON: JITSitem.SurveyJSON,

RatingJSON: JITSitem.RatingJSON

})

)

);

**Why this is better:**

* Reduces redundant ParseJSON() calls, improving performance.
* With({parsedJSON: ...}) caches parsed data for each item.

**2. Eliminate Unnecessary ClearCollect Calls**

ClearCollect clears and re-creates the collection, but it's often unnecessary.

**Optimized Approach:**

Replace:

PowerApps

CopyEdit

ClearCollect(colJITSv4, Blank());

With:

PowerApps

CopyEdit

Clear(colJITSv4);

Since Collect already adds items, there's no need for an empty ClearCollect() before.

**3. Optimize the Final Patch Operation**

Instead of:

PowerApps

CopyEdit

ForAll(colJITSv4TEMP As ActionItem, Patch(colJITSv4, LookUp(colJITSv4, ID = ActionItem.ID), { Status: ... }));

Use:

PowerApps

CopyEdit

Patch(colJITSv4, colJITSv4TEMP, { Status: If(!IsBlank(CompletedBy.Email), "Closed - Resolved", If(IsBlank(AsgnToOrg), "Open - New", If(CaseHold.Value = "-", "Open - Researching", CaseHold.Value))) });

**Why this is better:**

* Patch(Collection, Collection, Updates) performs batch updates.
* Reduces ForAll() iterations, making it faster.

**Final Optimized Version**

PowerApps

CopyEdit

Set(varFlow\_JITS, ACP\_JECC\_GetSPOlist\_JITS.Run().listjson);

Clear(colJITSv4);

ClearCollect(colJITSv3raw,

ForAll(Table(ParseJSON(varFlow\_JITS).value) As JITSitem, {

ID: Value(Text(JITSitem.Value.ID)),

JITScontrol: Text(JITSitem.Value.JITS\_Control),

CaseTitle: Text(JITSitem.Value.Title),

CaseJSON: JITSitem.Value.CaseV6\_TaskComments,

OriginNotes: Text(JITSitem.Value.CaseV6\_Notes),

Created: DateTimeValue(Text(JITSitem.Value.Created)),

CreatedBy: { DisplayName: Text(JITSitem.Value.Author.DisplayName), Email: Text(JITSitem.Value.Author.Email) },

Modified: DateTimeValue(Text(JITSitem.Value.Modified)),

ModifiedBy: { DisplayName: Text(JITSitem.Value.Editor.DisplayName), Email: Text(JITSitem.Value.Editor.Email) },

SurveyJSON: Text(JITSitem.Value.CustomerReviewSurvey),

RatingJSON: Text(JITSitem.Value.CustomerReviewRatings)

})

);

ForAll(colJITSv3raw As JITSitem,

With({parsedJSON: ParseJSON(Text(JITSitem.CaseJSON))},

Collect(colJITSv4, {

ID: JITSitem.ID,

JITScontrol: JITSitem.JITScontrol,

CaseTitle: JITSitem.CaseTitle,

Customers: ForAll(Table(parsedJSON.Customers) As JSONitem, {

DisplayName: Text(JSONitem.Value.DisplayName),

Email: Text(JSONitem.Value.Email)

}),

CustomerOrg: { Value: Text(parsedJSON.CustomerOrg) },

VIP: { Value: Text(parsedJSON.VIP) },

Priority: { Value: If(IsBlank(parsedJSON.Priority), "3 - Standard", Text(parsedJSON.Priority)) },

AsgnToOrg: Text(parsedJSON.AsgnToOrg),

AsgnToLead: {

DisplayName: Text(parsedJSON.AsgnToLead.DisplayName),

Email: Text(parsedJSON.AsgnToLead.Email)

},

AsgnToPersonnel: ForAll(Table(parsedJSON.AsgnToPersonnel) As JSONitem, {

DisplayName: Text(JSONitem.Value.DisplayName),

Email: Text(JSONitem.Value.Email)

}),

Status: Text(parsedJSON.Status),

CaseHold: { Value: If(IsBlankOrError(parsedJSON.CaseHold), "-", Text(parsedJSON.CaseHold)) },

CaseDetermination: { Value: Text(parsedJSON.CaseDetermination) },

CompletedDate: DateValue(Text(parsedJSON.CompletedDate)),

CompletedBy: {

DisplayName: Text(parsedJSON.CompletedBy.DisplayName),

Email: Text(parsedJSON.CompletedBy.Email)

},

CaseDescription: JITSitem.OriginNotes,

Created: JITSitem.Created,

CreatedBy: JITSitem.CreatedBy,

Modified: JITSitem.Modified,

ModifiedBy: JITSitem.ModifiedBy,

SurveyJSON: JITSitem.SurveyJSON,

RatingJSON: JITSitem.RatingJSON

})

)

);

Patch(colJITSv4, colJITSv4, { Status: If(!IsBlank(CompletedBy.Email), "Closed - Resolved", If(IsBlank(AsgnToOrg), "Open - New", If(CaseHold.Value = "-", "Open - Researching", CaseHold.Value))) });

**Expected Improvements**

* **Faster Execution:** Reducing redundant JSON parsing improves performance.
* **Less Memory Usage:** Fewer function calls reduce computational load.
* **More Efficient Data Updates:** Using Patch() with collections speeds up bulk updates.
* **Cleaner Code:** Easier to maintain and debug.

Would you like further refinements based on specific bottlenecks? 🚀